Graph Prompting for Graph Learning Models: Recent Advances and Future Directions

Presenters: Xingbo Fu, Zehong Wang, Jundong Li

University of Virginia, University of Notre Dame, University of Connecticut

KDD 2025 Tutorial

August 2025

Presenters

Xingbo Fu is a PhD candidate at the University of Virginia. His research interests focus mainly on graph learning, federated learning, and healthcare informatics. He has publications in major conferences and journals, such as KDD, ICLR, AAAI, ECAI, ICHI, SGIKDD Explorations, and TMLR. He brings industry experience through internships at Amazon and Netflix.

Zehong Wang is a PhD candidate at the University of Notre Dame. His research interests are broadly in machine learning and data mining, with a particular focus on foundation models. His work has appeared at top-tier conferences such as KDD, NeurIPS, ICML, IJCAI, NAACL, ACL, SDM, and WSDM. He has also applied his expertise in industry as an Applied Scientist Intern at Amazon.

Jundong Li is an Associate Professor at the University of Virginia. His research interests are generally in data mining and machine learning, with a focus on graph machine learning, trustworthy/safe machine learning, and LLMs. He has received several prestigious awards, including the SIGKDD Rising Star Award (2024), the SIGKDD Best Research Paper Award (2022), and the NSF CAREER Award (2022).

Tutorial Outline

Time	Section
1:00pm ~ 1:30pm	Section 1: Introduction
1:30pm ~ 2:00pm	Section 2: Graph Pre-training for Graph Prompting
2:00pm ~ 3:00pm	Section 3: Techniques in Graph Prompting
3:00pm ~ 3:30pm	Coffee Break
3:30pm ~ 3:45pm	Section 4: Summary and Future Directions
3:45pm ~ 4:00pm	Section 5: Q&A

Tutorial Website & Survey Paper

Tutorial website

Slides & more information about this tutorial

https://www.xingbofu.com/tutorials/kdd25-graph-promptng

Graph Prompting for Graph Learning Models: Recent Advances and Future Directions

Xingbo Fu, Zehong Wang, Zihan Chen, Jiazheng Li, Yaochen Zhu, Zhenyu Lei, Cong Shen, Yanfang Ye, Chuxu Zhang, Jundong Li

https://arxiv.org/abs/2506.08326

August 3–7, 2025

KDD2025

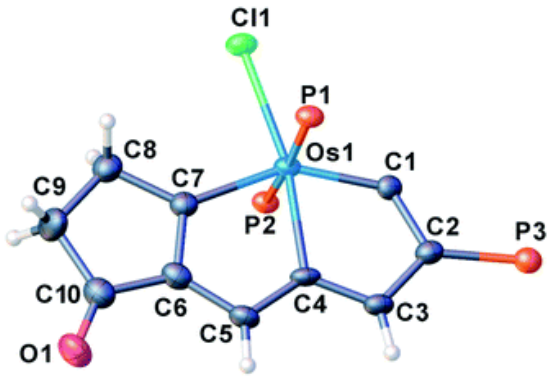
Section 1
Introduction

Jundong Li

Associate Professor

University of Virginia

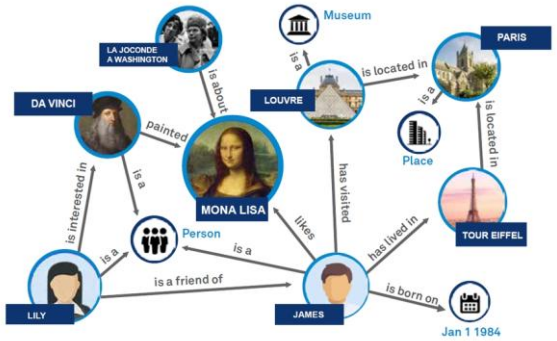
Graphs



Molecular graphs

Social network

Transportation network

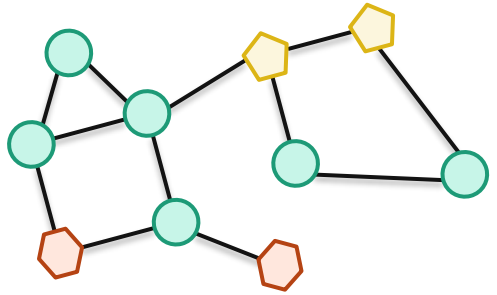


Knowledge graph

Modeling Real-World Data as Graphs

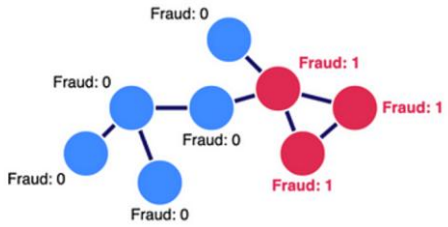
A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$

- $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$: the node set
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$: the edge set
- $X \in \mathbb{R}^{N \times N}$: node feature matrix
- $A \in \{0,1\}^{N \times N}$: adjacency matrix

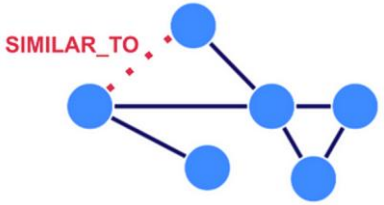


A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$

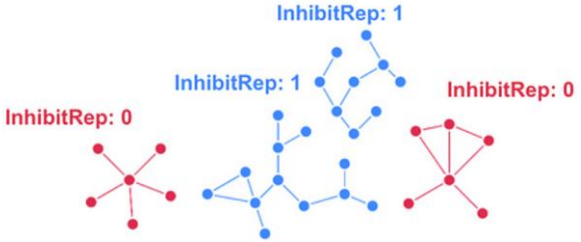
Tasks for graphs



Node-level tasks



Edge-level tasks

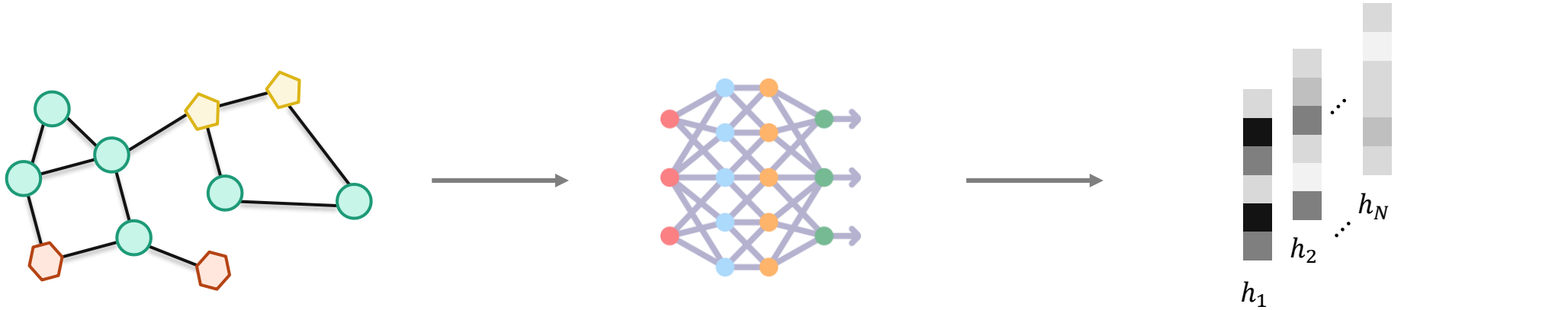


Graph-level tasks

Graph Representation Learning

Learn expressive graph representations by training graph learning models

- Graph embeddings can be used for downstream tasks

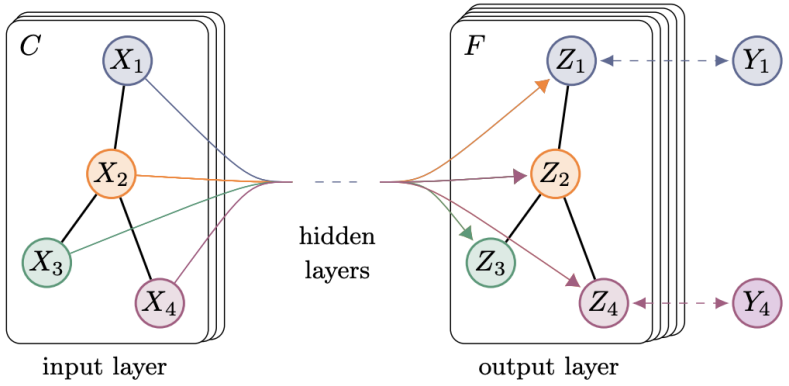


A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ | Graph learning model f | Low-dimensional representations

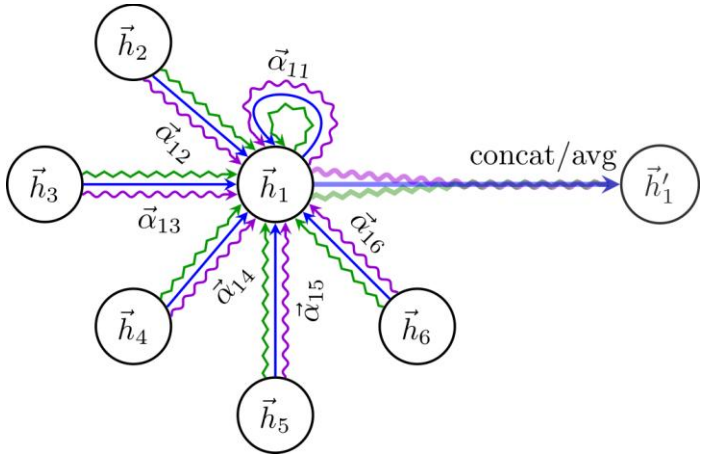
Graph Learning Models

Graph neural networks (GNN)

- Iteratively aggregate information in the neighborhood via the message-passing mechanism



Graph convolutional networks (GCN)[1]



Graph attention networks (GAT)[2]

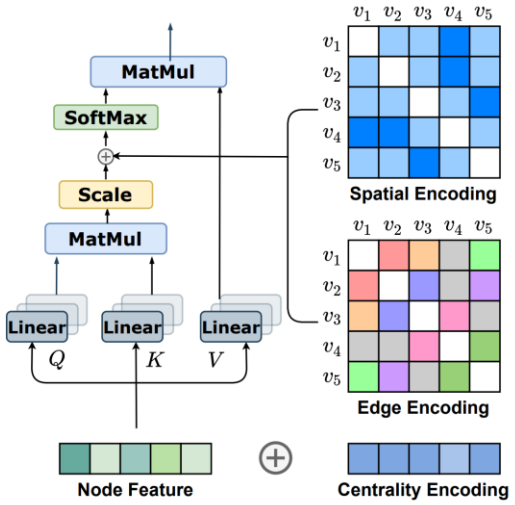
August 3–7, 2025
KDD2025

[1] Kipf, Thomas N., and Max Welling. "Semi-supervised Classification With Graph Convolutional Networks." ICLR 2017.
[2] Veličković, Petar, et al. "Graph Attention Networks." ICLR 2018.

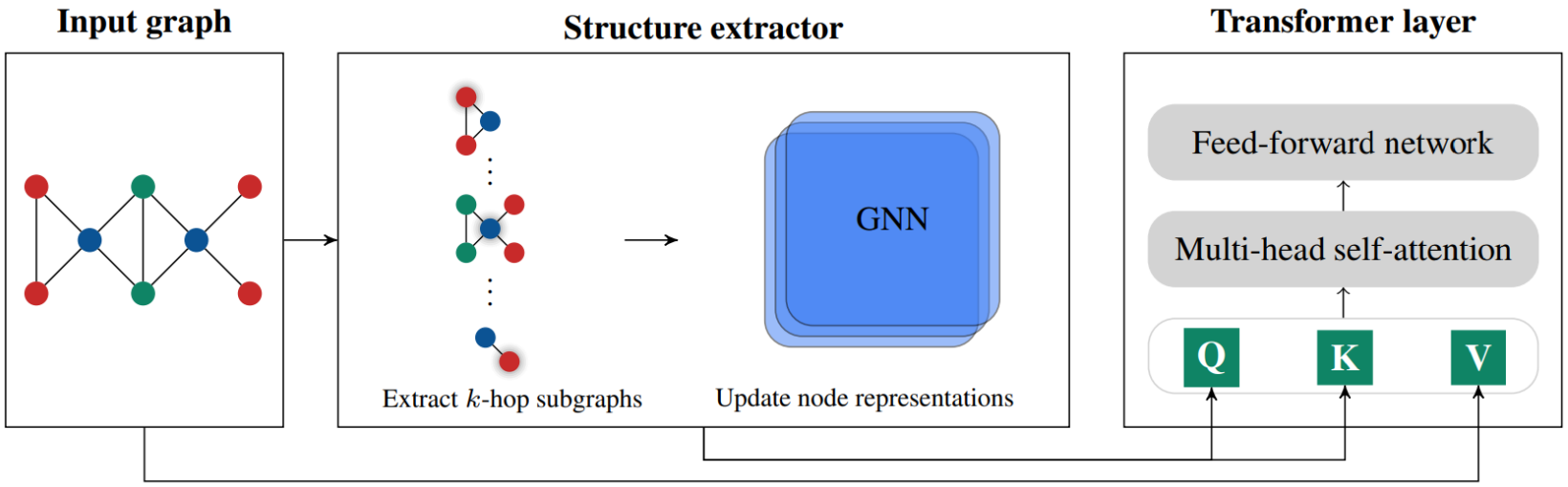
Graph Learning Models

Graph Transformers (GT)

- Apply Transformer architectures to graph data



Graphormer[1]



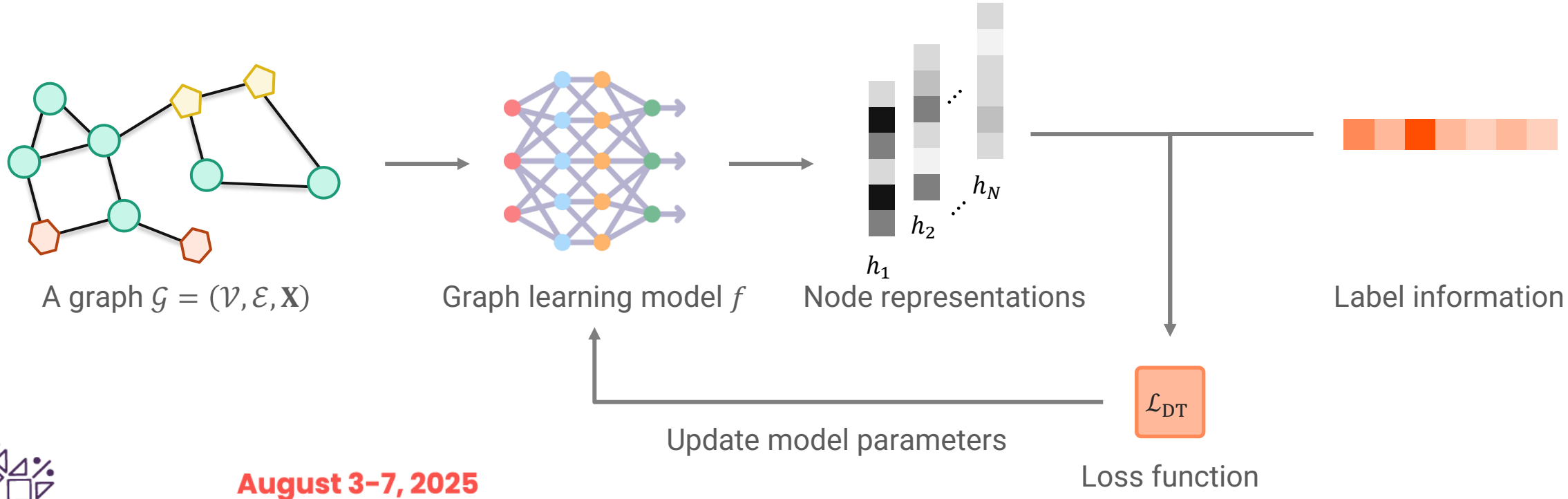
Structure-aware Transformer (SAT)[2]

[1] Ying, Chengxuan, et al. "Do Transformers Really Perform Badly for Graph Representation?." NeurIPS 2021.
[2] Chen, Dexiong, et al. "Structure-Aware Transformer for Graph Representation Learning." ICML 2022.

Traditional Training Paradigm for Graph Learning Models

End-to-end training

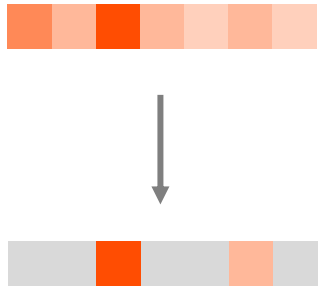
- Optimize graph learning models based on abundant label information



Limitations of Traditional Training Paradigm

Rely heavily on label information

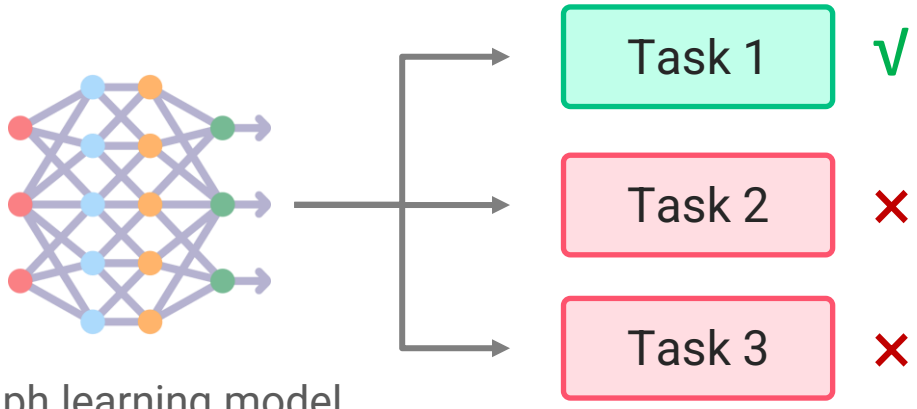
- Sufficient labeled graph data may be inaccessible in practice



Insufficient label information

Poor generalization

- Graph learning models cannot be well generalized to other downstream tasks



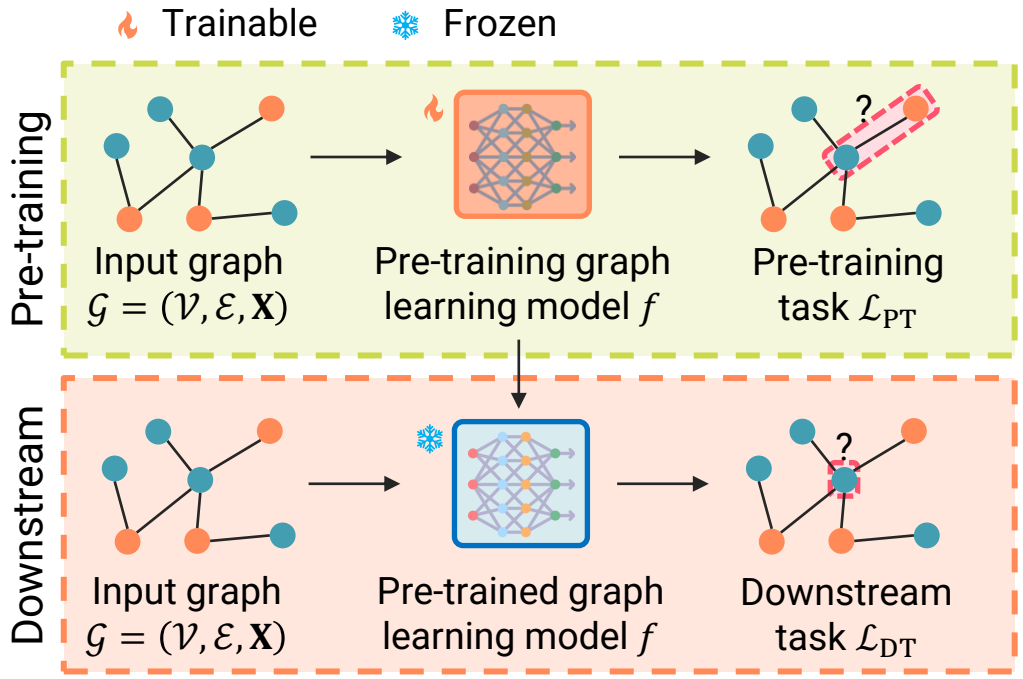
Graph learning model

Graph Pre-training via Self-supervised Learning

Learn generalizable graph embeddings without label information

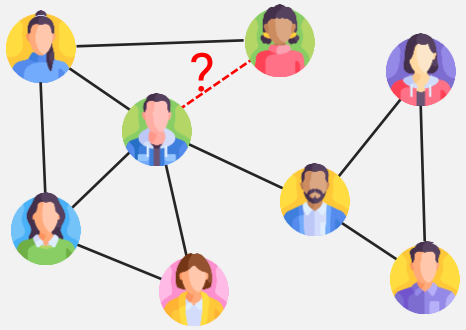
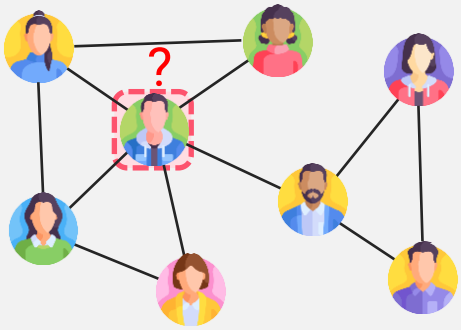
- Pre-training stage
- Graph learning models are learned by solving hand-crafted auxiliary tasks $\mathcal{L}_{\text{PT}}$
- Supervision signals are acquired from graph data itself

- Downstream tasks
- Pre-trained graph learning models directly generate graph embeddings used for downstream tasks $\mathcal{L}_{\text{DT}}$



Objective Gap between Pre-training and Downstream Tasks

Example: edge prediction → node classification

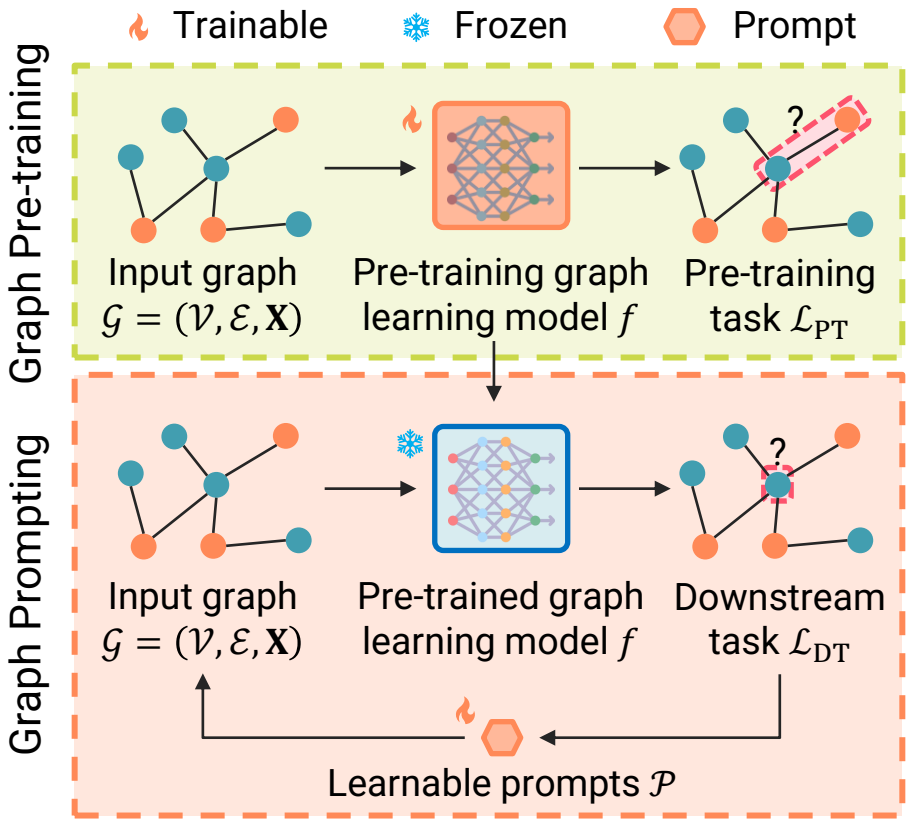
Phase	Pre-training stage	Downstream stage
Illustration		
Task	Edge prediction	Node classification
Involved embeddings	A pair of nodes	A single node
Prediction	Edge probabilities	Node class probabilities

The "Pre-training, Prompting" Scheme

Two-stage adaptation of pre-trained graph learning models

- Stage 1: graph pre-training via self-supervised learning
- A graph learning model f is pre-trained on a pre-training task $\mathcal{L}_{\text{PT}}$

- Stage 2: graph prompting for adaptation
- Adapt the pre-trained graph learning model f for the downstream task $\mathcal{L}_{\text{DT}}$ by learning extra prompts $\mathcal{P}$
- the pre-trained graph learning model f keeps frozen

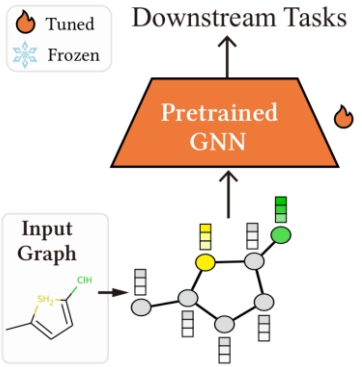


The "pre-training, prompting" scheme for graph learning models

Comparison between Graph Fine-tuning & Graph Prompting

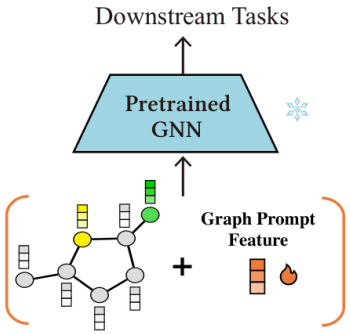
Graph fine-tuning

- Adapt pre-trained graph learning models by further tuning model parameters
- Tunable parameters: the whole graph learning model
- Unchanged part: graph data



Graph prompting

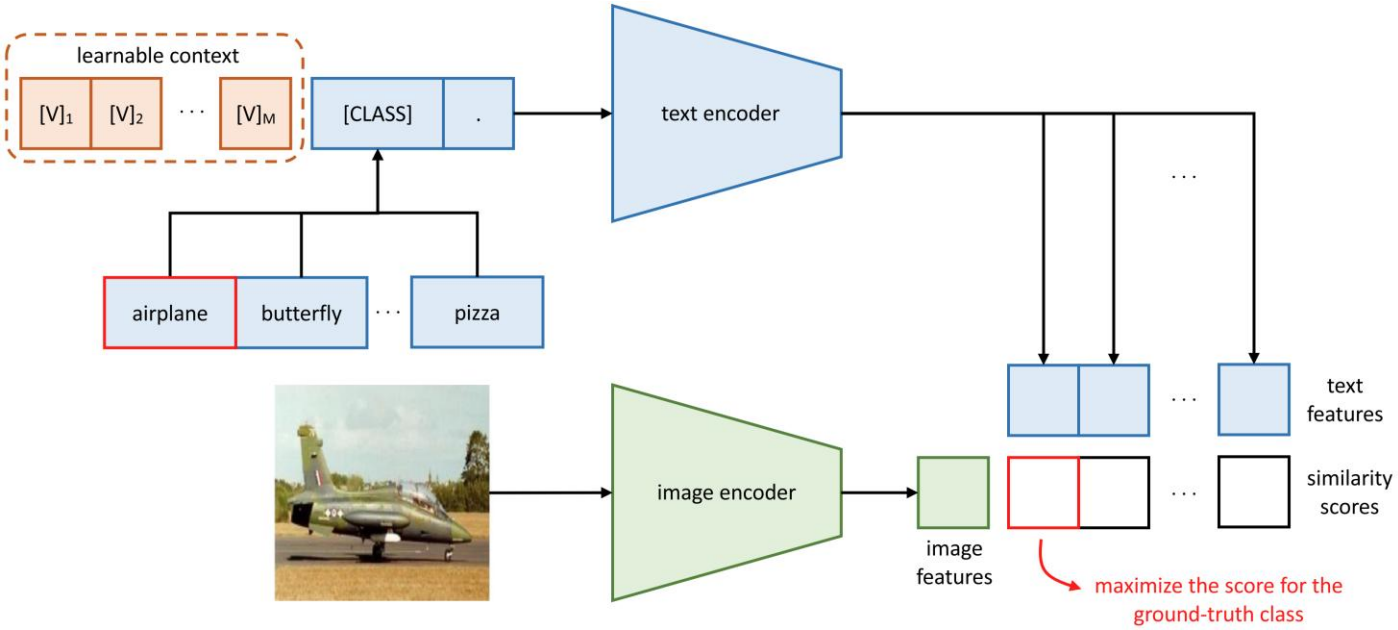
- Adapt pre-trained graph learning models by learning extra prompts
- Tunable parameters: additional prompts
- Unchanged part: the pre-trained graph learning model



Prompting in NLP and CV

Learning to modify the input data with extra trainable prompts

- Prompting in NLP: learn extra trainable tokens



Context optimization (CoOP)[1]

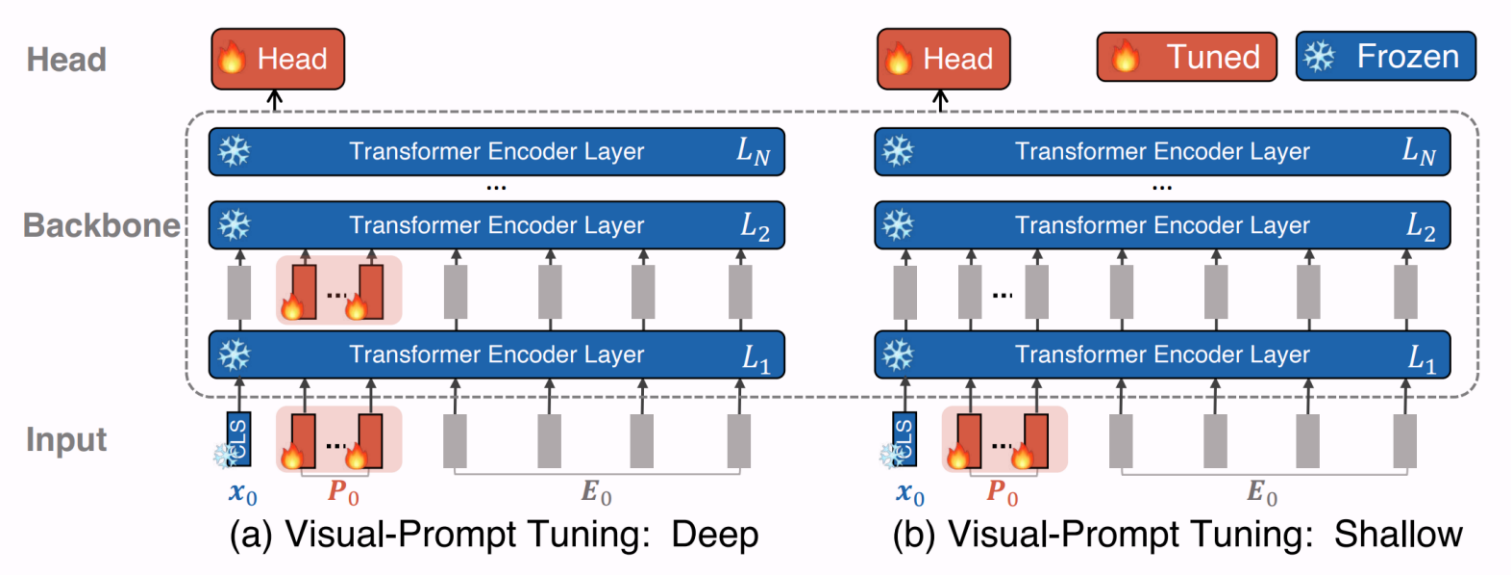
August 3–7, 2025
KDD2025

[1] Zhou, Kaiyang, et al. "Learning to Prompt for Vision-Language Models." IJCV 2022.

Prompting in NLP and CV

Learning to modify the input data with extra trainable prompts

- Prompting in CV: learn extra trainable patches



Visual prompt tuning (VPT)[1]

August 3–7, 2025
KDD2025

[1] Jia, Menglin, et al. "Visual Prompt Tuning." ECCV 2022.

Challenges in Prompting for Graph Learning Models

Non-Euclidean graph data

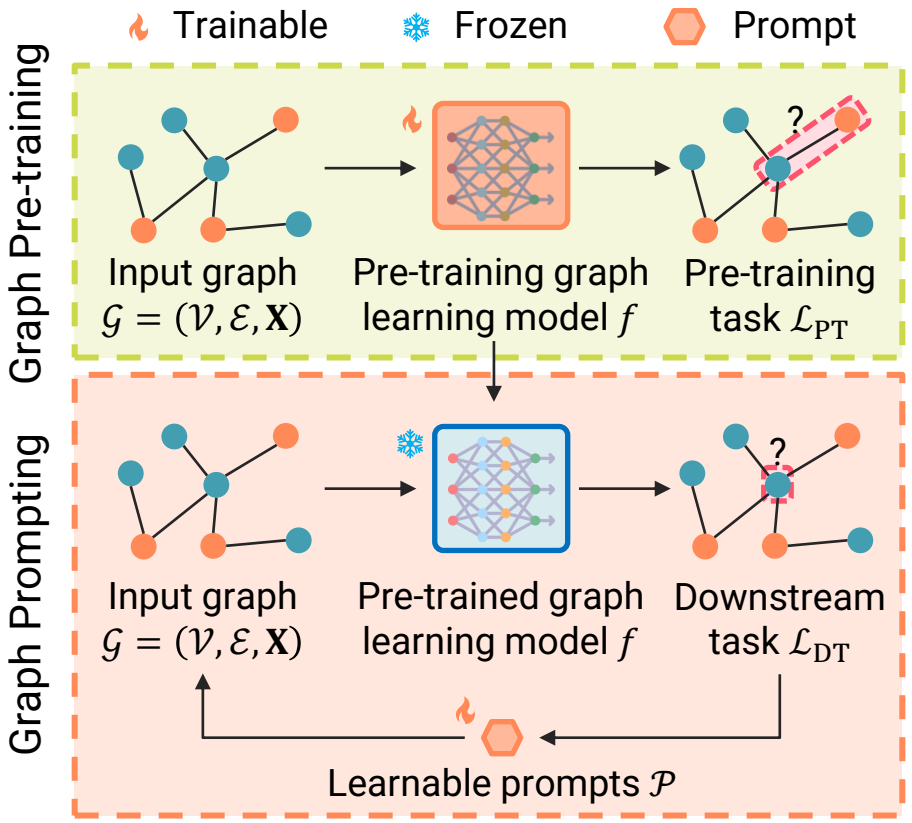
- Important structural information in graph data

Pre-training compatibility

- Pre-training strategies: generative/contrastive/…
- Should be compatible with various pre-training strategies

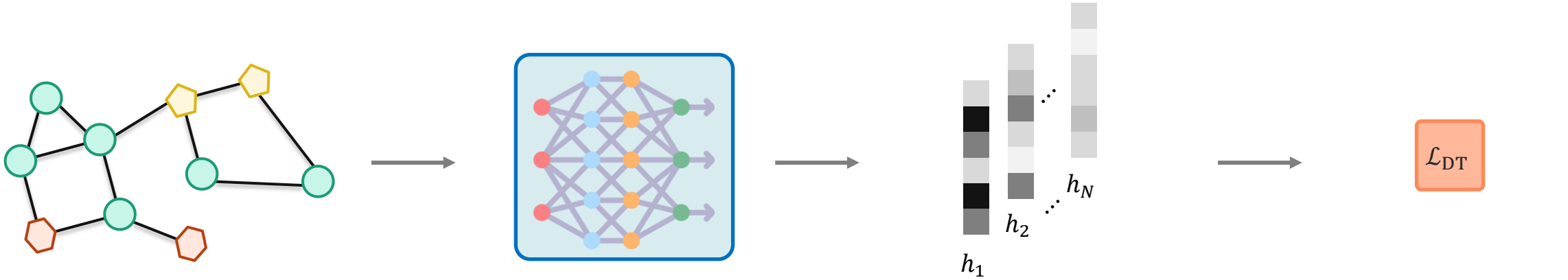
Downstream task universality

- Downstream tasks: node-level/graph-level/…
- Should be universal for different downstream tasks



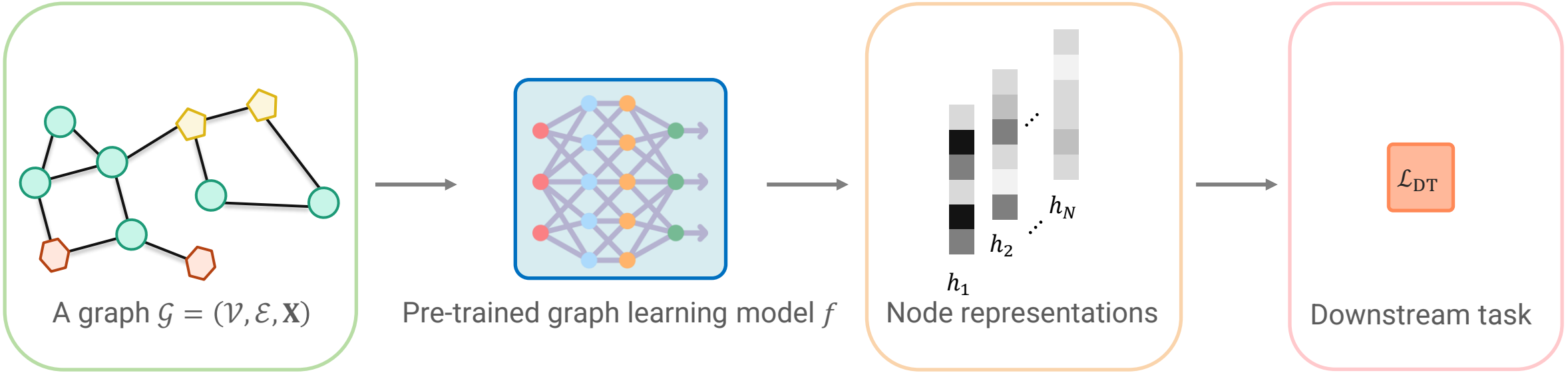
The “pre-training, prompting” scheme for graph learning models

Where Can We Design Prompts for Graph Prompting?

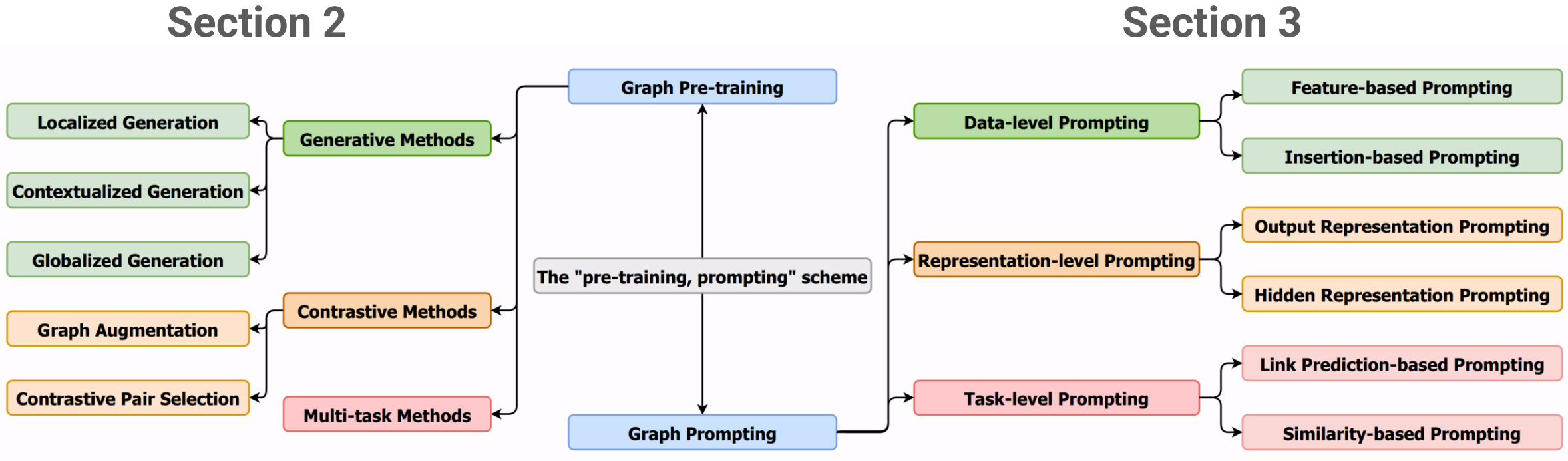


A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ | Pre-trained graph learning model f | Node representations | Downstream task

Where Can We Design Prompts for Graph Prompting?



Key Techniques in the "Pre-training, Prompting" Scheme



Section 2
Graph Pre-training for Graph Prompting

Zehong Wang

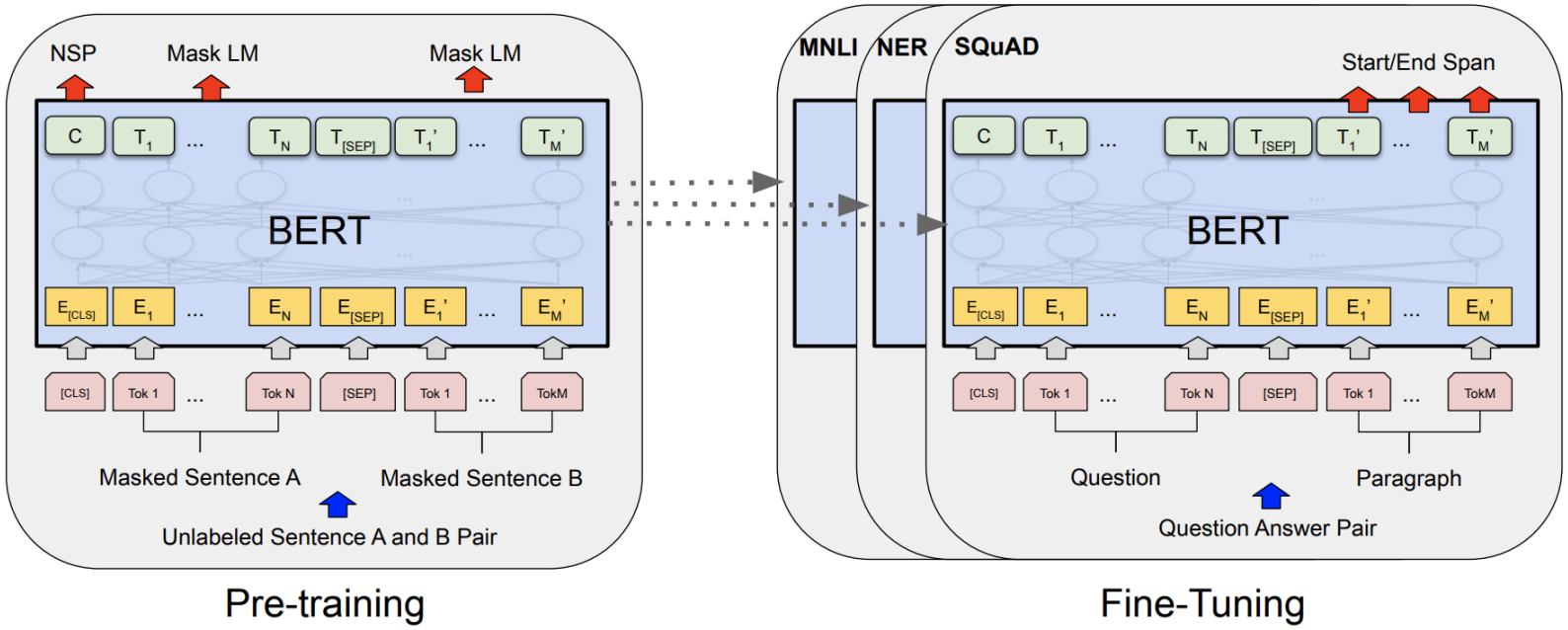
PhD Candidate

University of Notre Dame

From Supervised Learning to Self-supervised Learning

Generative Pre-training[1]

- Generate the masked parts
- Masked Token Prediction
- Next Sentence Prediction



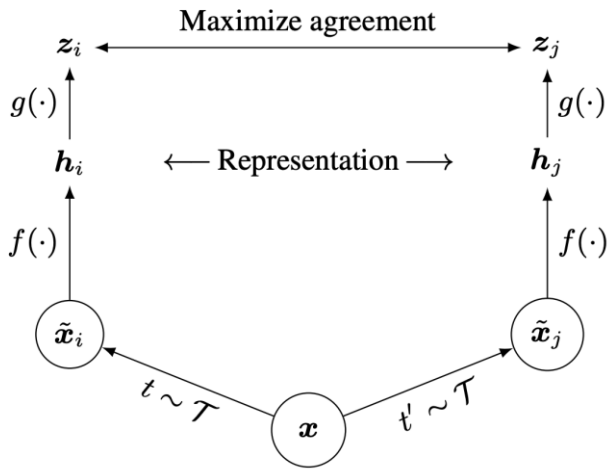
August 3–7, 2025 KDD2025

[1] Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL 2019.

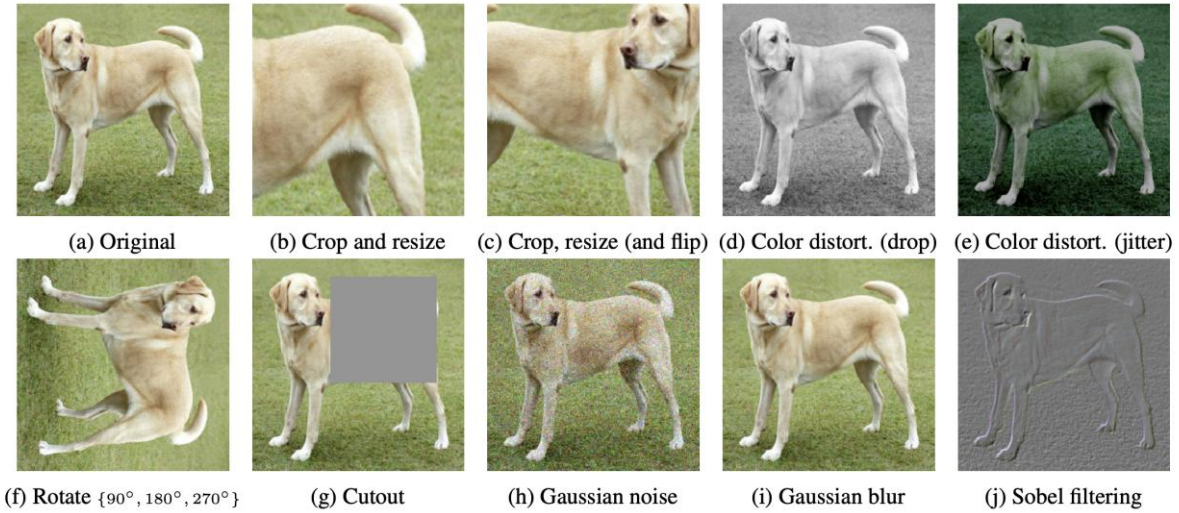
From Supervised Learning to Self-supervised Learning

Contrastive Pre-training[1]

- Pull positives together and push negatives apart for discriminative representations



Contrastive learning[1]



Data augmentation for harder tasks

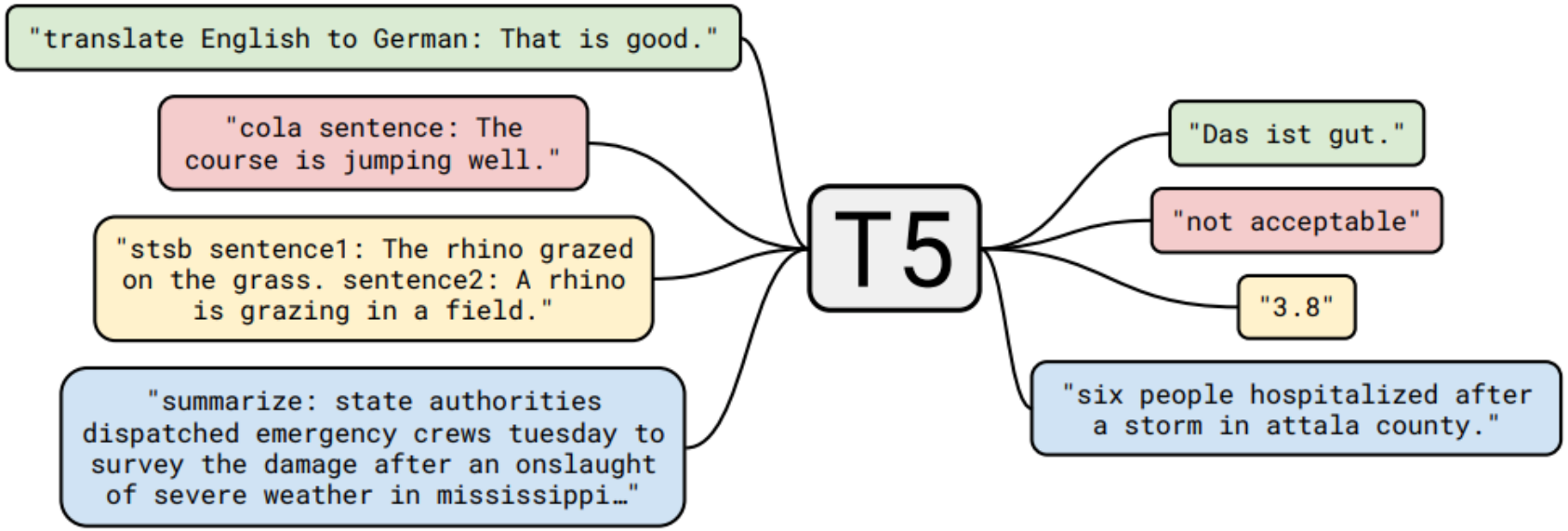
August 3–7, 2025
KDD2025

[1] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

From Supervised Learning to Self-supervised Learning

Multi-Task Pre-training[1]

- A single model to optimize multiple tasks, improving model capability and generalization



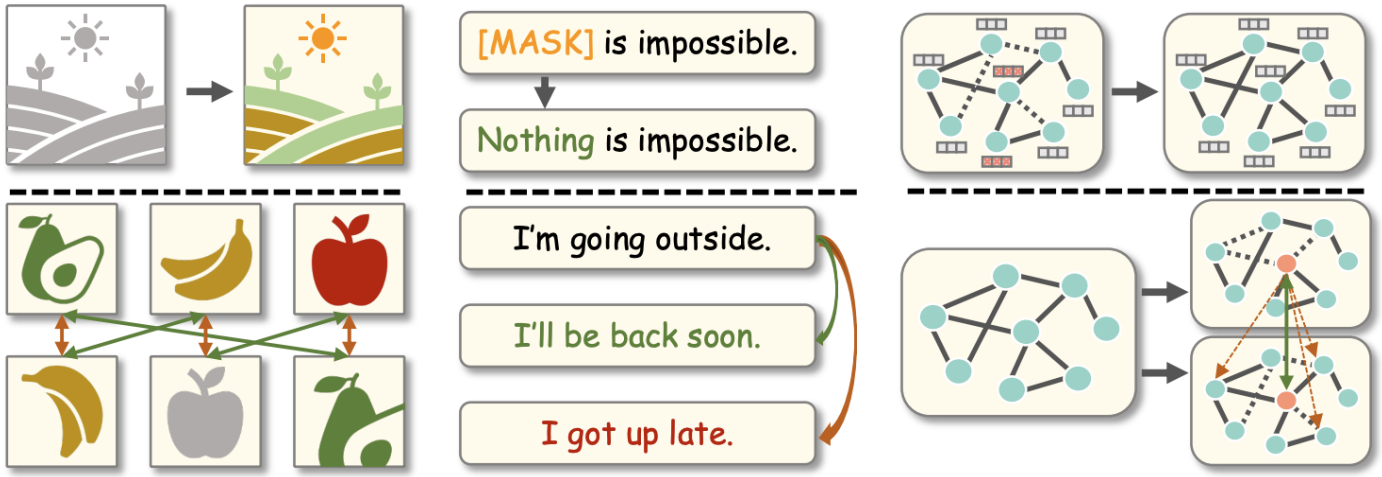
[1] Raffel, Colin, et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." JMLR 2020.

Pre-Training on Graphs

We need to carefully design pre-training tasks on graphs!

How to design pre-training tasks on graphs?

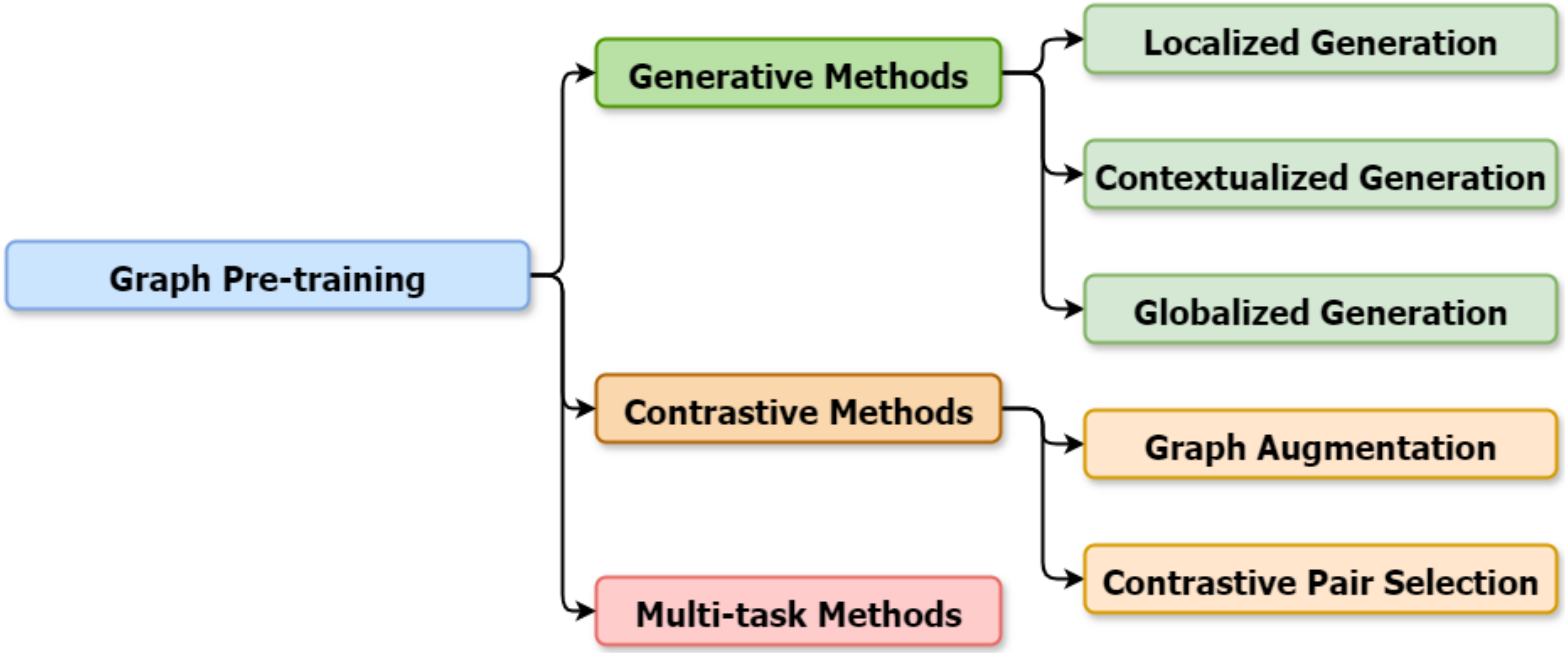
- Can we directly apply pre-training tasks on CV and NLP to graphs?
- Image/Text is in the Euclidean space
- Graph is in the non-Euclidean space



Toy examples of different pre-training tasks in CV, NLP, and graph learning[1]

[1] Liu, Yixin, et al. "Graph Self-Supervised Learning: A Survey." TKDE 2022.

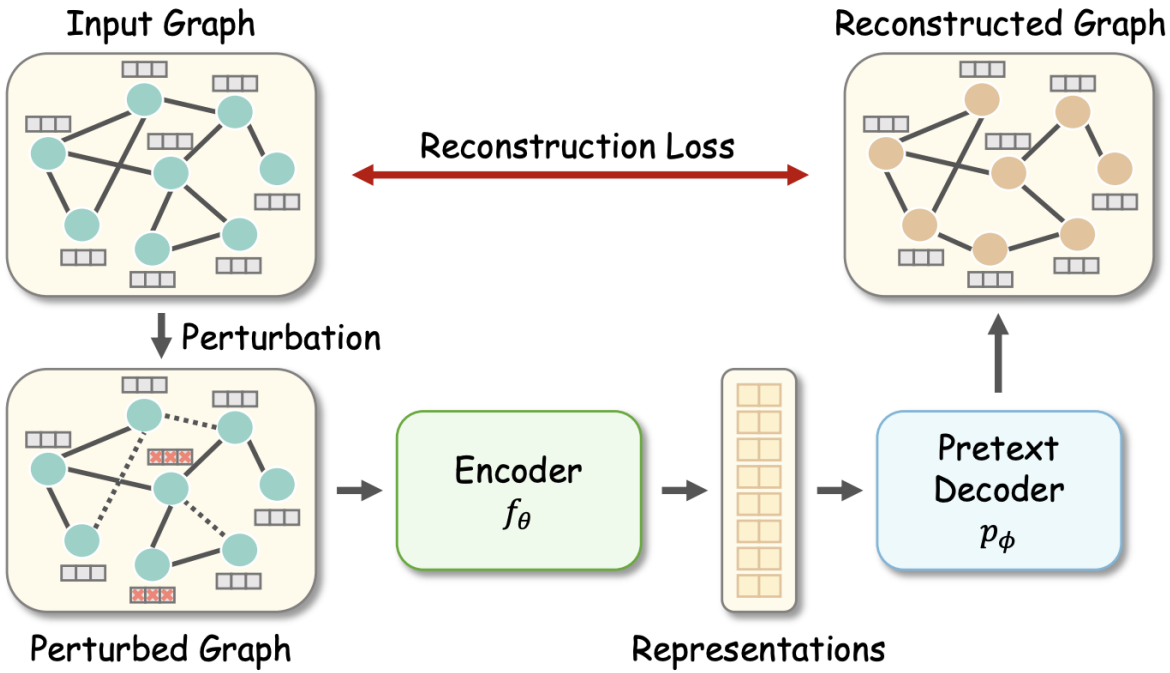
Taxonomy of Graph Pre-Training Techniques



Generative Methods

Intuition: Corrupt the graph and reconstruct the corrupted parts

- Three types of generative methods
- Localized generation
- Contextualized generation
- Globalized generation

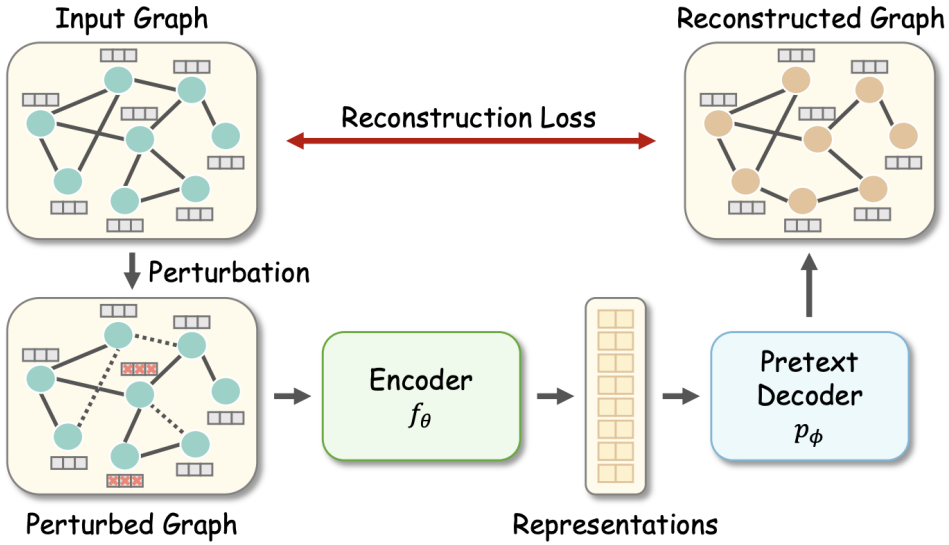


An illustration of generative methods for graph pre-training[1]

[1] Liu, Yixin, et al. "Graph Self-Supervised Learning: A Survey." TKDE 2022.

Generative Methods

Intuition: Corrupt the graph and reconstruct the corrupted parts



$\mathcal{G}$: the original graph

$\tilde{\mathcal{G}}$: the corrupted graph

θ: model parameters

Optimize the loss function: $\mathcal{L}_{\mathrm{PT}} = \min_{\theta} \mathcal{L}\left(f(\tilde{\mathcal{G}};\theta), \rho(\mathcal{G})\right)$

Reconstruction

Encode corrupted graph

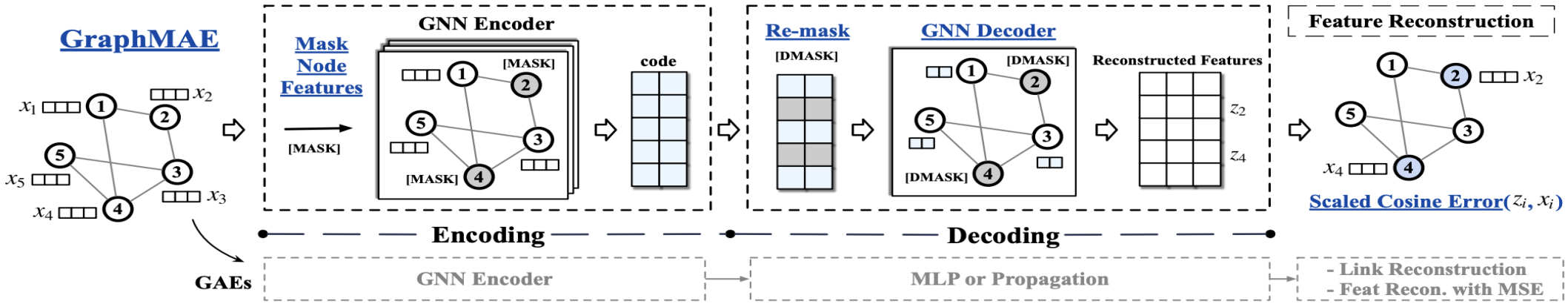
Define reconstruction target

Generative Methods

Localized generation: reconstruct node features

- GraphMAE[1]: Self-Supervised Masked Graph Autoencoders
- Corruption is to mask the node features $\tilde{\mathcal{G}} = mask_{node}(\mathcal{G})$
- Encoding $f(\tilde{\mathcal{G}}; \theta)$ is to encode the corrupted graphs **twice**
- Reconstruction is to reconstruct the masked node features

Could be extended to centrality level or clustering coefficient



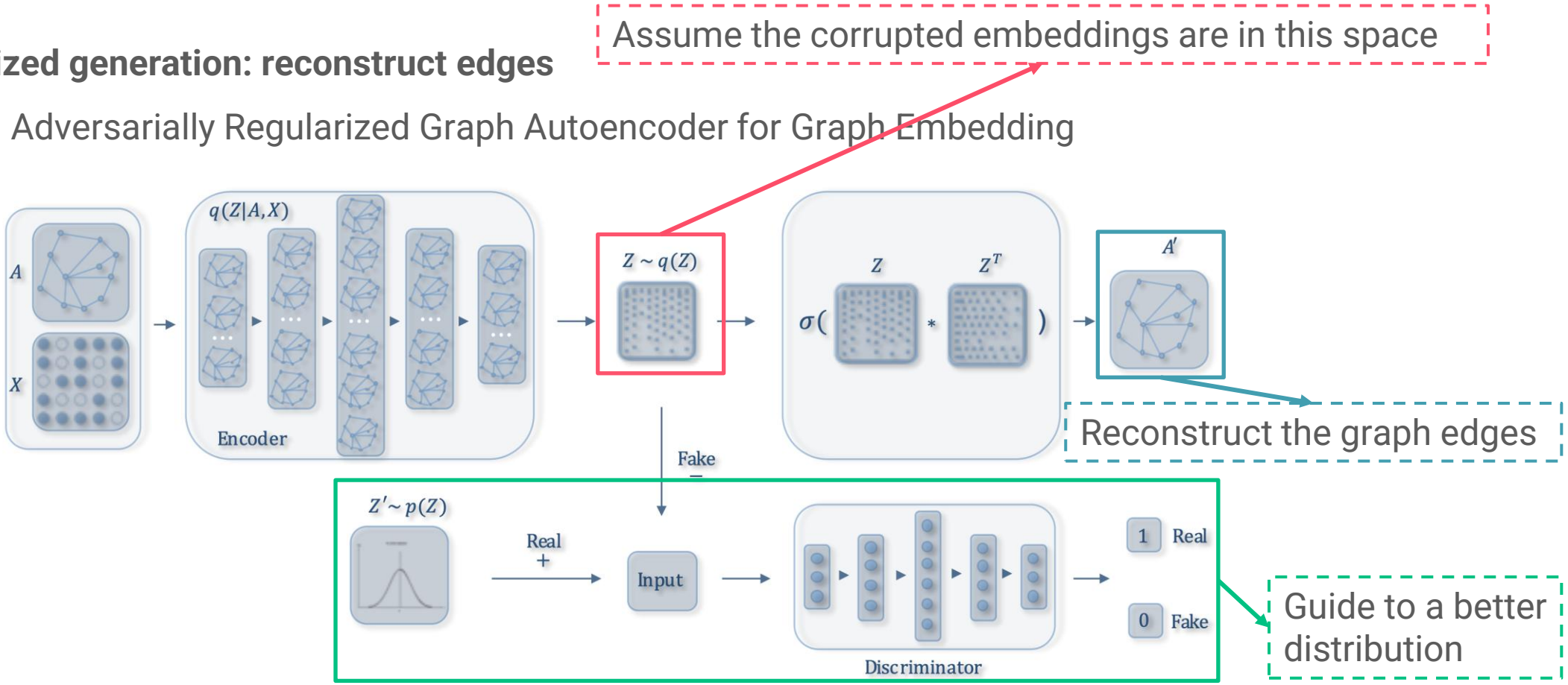
August 3–7, 2025
KDD2025

[1] Hou, Zhenyu, et al. "GraphMAE: Self-Supervised Masked Graph Autoencoders." KDD 2022.

Generative Methods

Contextualized generation: reconstruct edges

- ARVGA[1]: Adversarially Regularized Graph Autoencoder for Graph Embedding

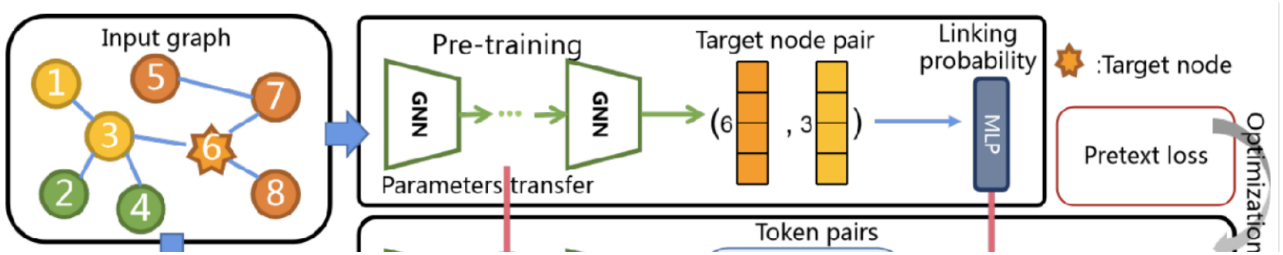


[1] Pan, Shirui, et al. "Adversarially Regularized Graph Autoencoder for Graph Embedding." IJCAI 2018.

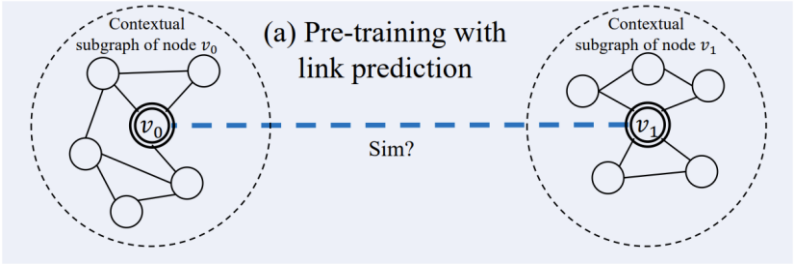
Generative Methods

Contextualized generation: reconstruct edges

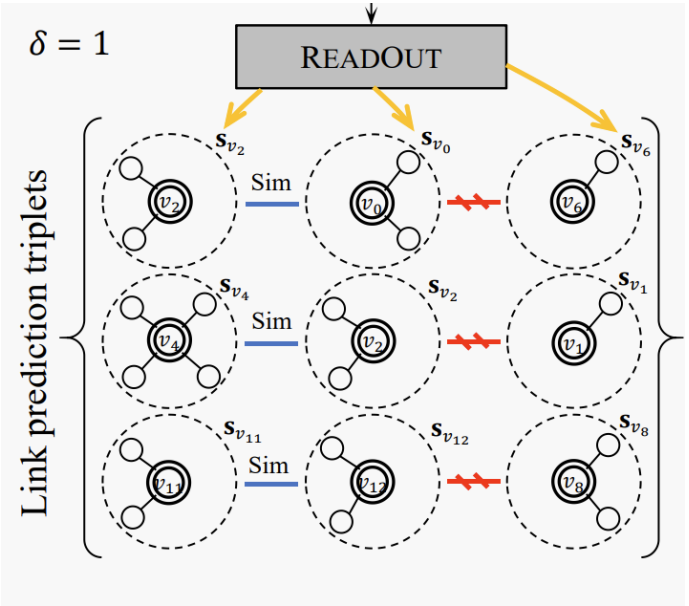
- Many graph prompting methods adapt contextualized generation for pre-training



GPPT[1]



GraphPrompt[2]



GraphPrompt+[3]

[1] Sun, Mingchen, et al. "GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks." KDD 2022.

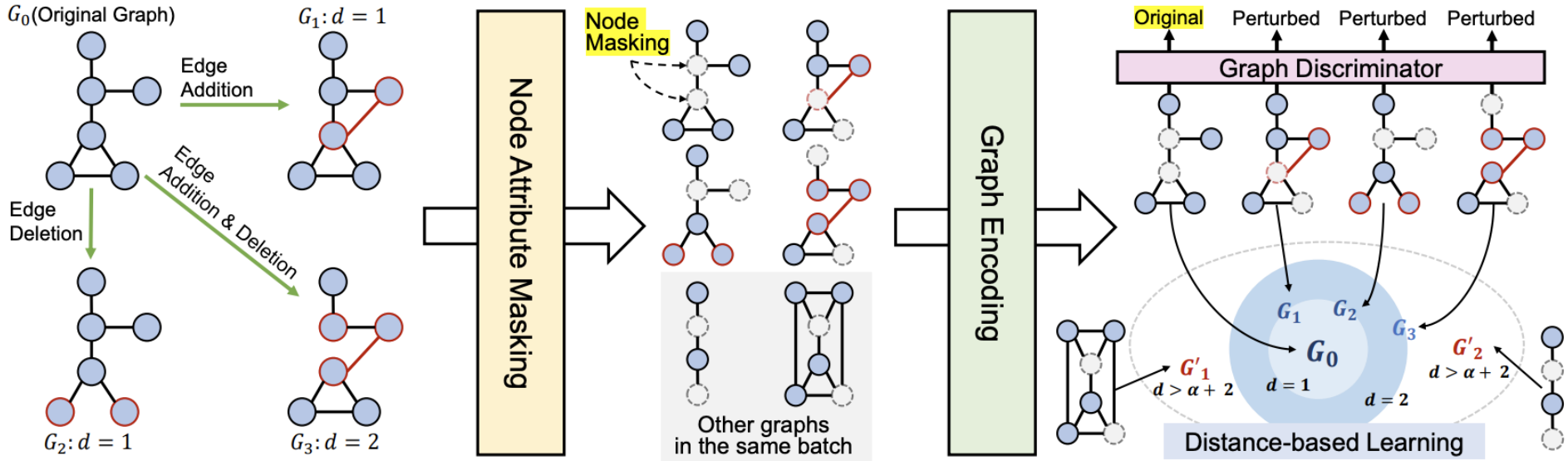
[2] Liu, Zemin, et al. "Graphprompt: Unifying pre-training and downstream tasks for graph neural networks." WWW 2023.

[3] Yu, Xingtong, et al. "Generalized Graph Prompt: Toward a Unification of Pre-Training and Downstream Tasks on Graphs." TKDE 2024.

Generative Methods

Globalized generation: reconstruct graph-level property

- Corruption: distance-based edge masking and random node feature masking
- Reconstruction: reconstruct the edit distance between graphs



The workflow of D-SLA[1]

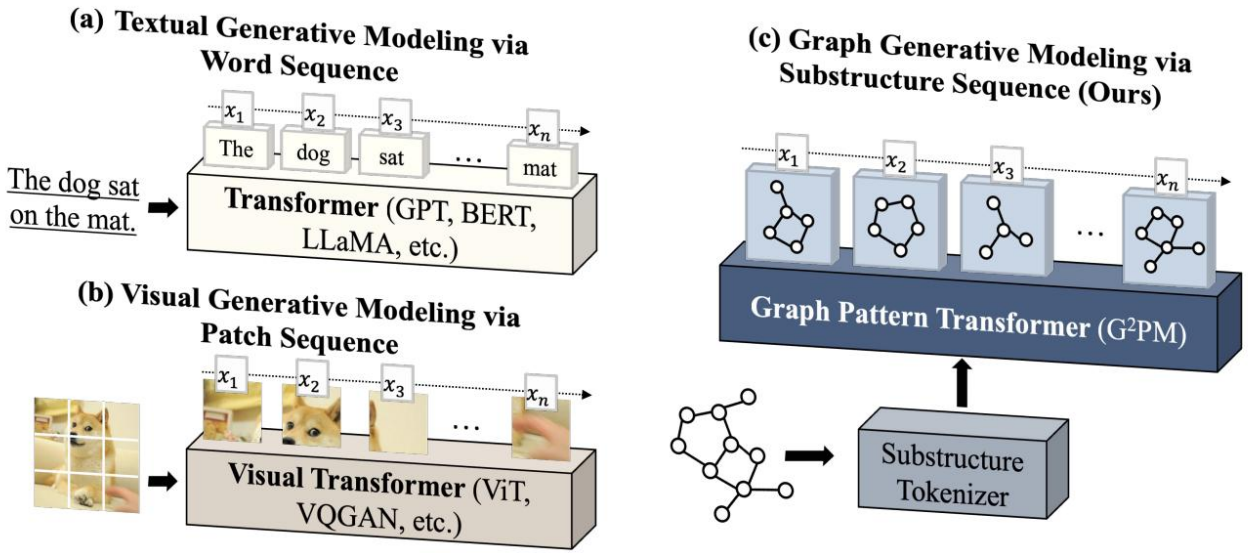
August 3–7, 2025
KDD2025

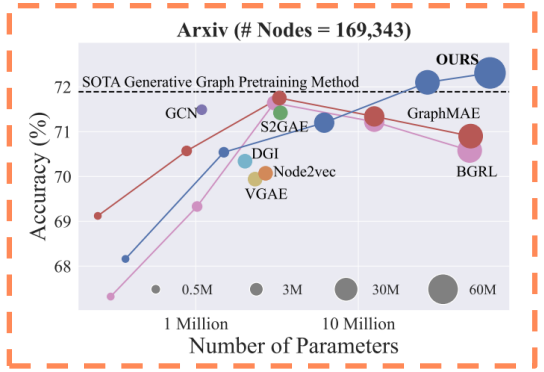
[1] Kim, Dongki, Jinheon Baek, and Sung Ju Hwang. "Graph Self-supervised Learning with Accurate Discrepancy Learning." NeurIPS 2022.

Generative Methods

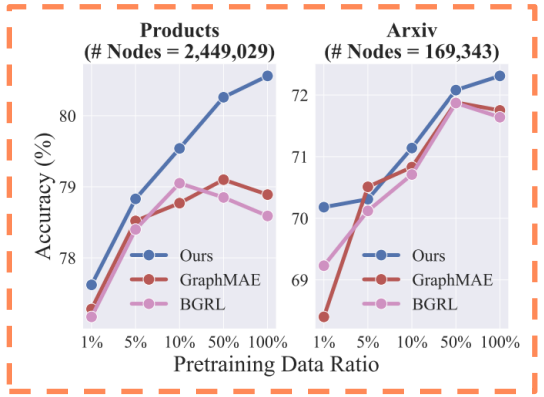
Others

- Scalable Graph Generative Modeling via Substructure Sequences (G2PM)
- Follow the general transformer pretraining on other domains, like CV and NLP





Model Scaling



Data Scaling

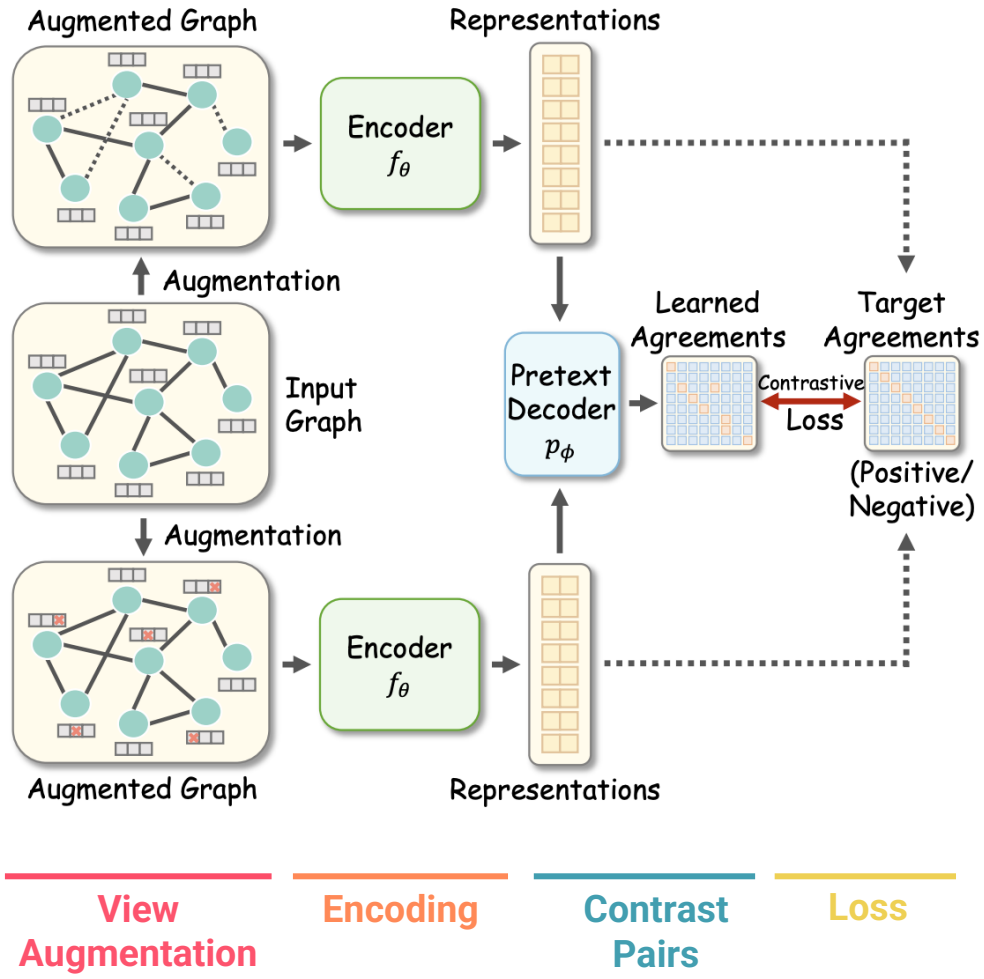
August 3–7, 2025
KDD2025

[1] Wang, Zehong, et al. "Scalable Graph Generative Modeling via Substructure Sequences." arXiv preprint arXiv:2505.16130 (2025).

Contrastive Methods

Main steps

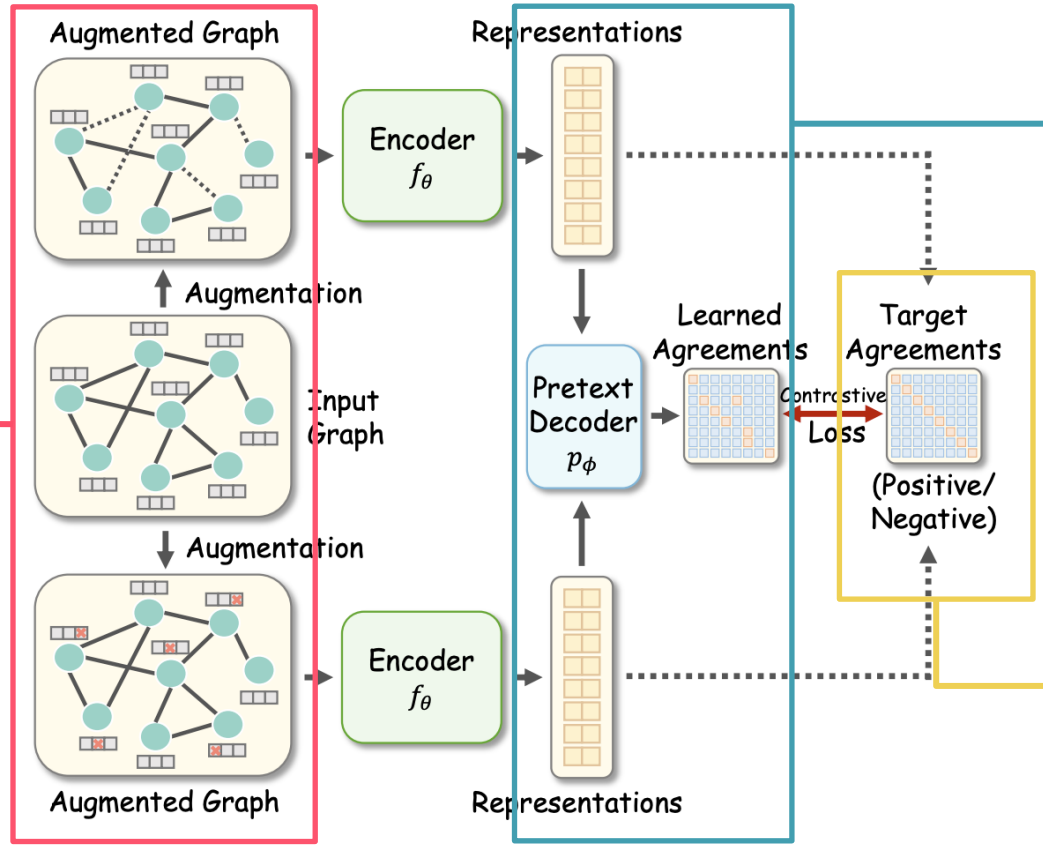
1. View augmentation
- Corrupt multiple graphs $\tilde{\mathcal{G}}_k = \text{corrupt}_k\ (\mathcal{G})$.
2. Encoding
- Encode each view $\mathbf{H}_\mathrm{k} = f(\tilde{\mathcal{G}}_k)$
3. Contrast pairs
- Pick the contrastive pairs and levels
4. Loss
- Maximize the similarity between positive pairs while minimizing the similarity between negative pairs



Contrastive Methods

Augmentation:

1. Feature, e.g., node feature masking, node feature shuffling, etc.
2. Structure, e.g., edge perturbation, edge diffusion
3. Substructure, e.g., ego-graph sampling, random walk sampling, etc.



Contrast Levels:

1. Local-Local, e.g., node-to-node contrast, node-to-subgraph contrast.
2. Local-Global, e.g., node-to-graph contrast.
3. Global-Global, e.g., graph-to-graph contrast.

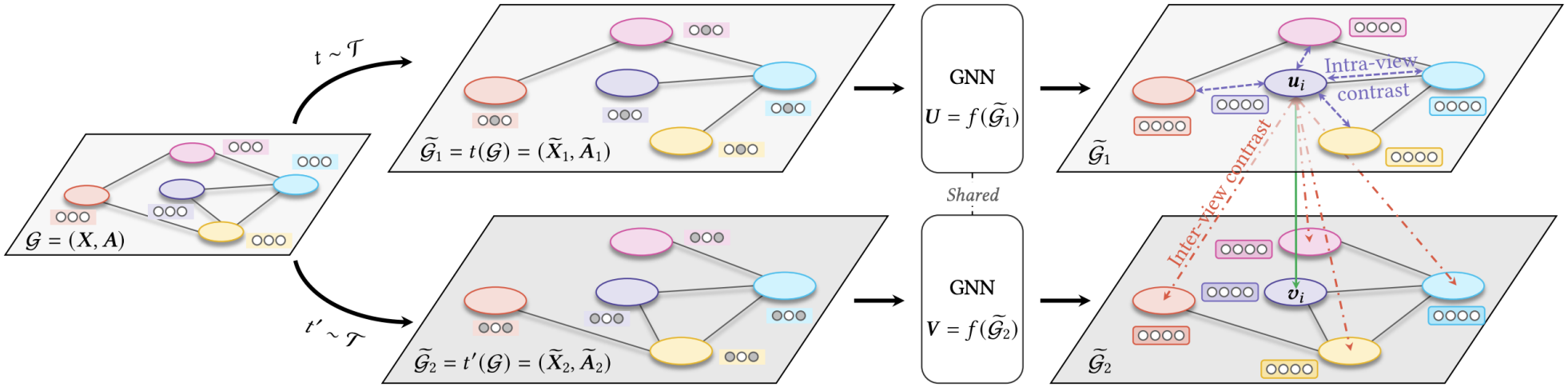
Loss:

1. InfoNCE
2. JS divergence
3. ...

Contrastive Methods

Local-to-local

- Augment two views of a single graph, treating the same node in these graphs as positives and the remaining as negatives



The workflow of local-to-local contrastive methods[1]

[1] Zhu, Yanqiao, et al. "Graph Contrastive Learning with Adaptive Augmentation." WWW 2021.

Contrastive Methods

Local-to-local

- Augment two views of a single graph, treating the same node in these graphs as positives and the remaining as negatives
- GRACE[1]: **random** node feature masking + **random** edge masking
- GCA[2]: **centrality-based** node feature masking + **centrality-based** edge masking

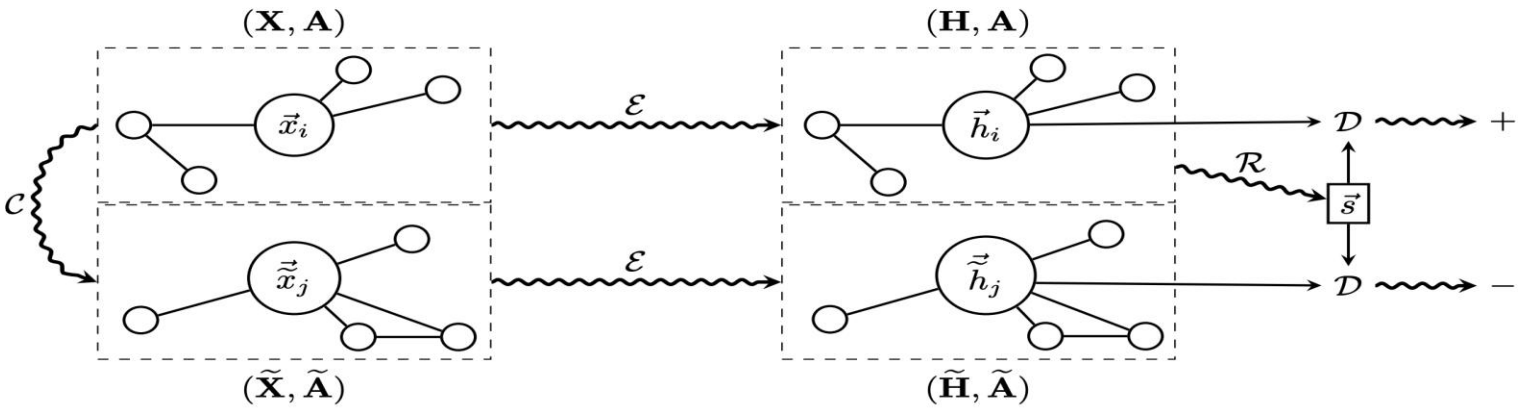
$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{positive pair}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negative pairs}}},$$

[1] Xia, Jun, et al. "SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation." WWW 2022.
[2] Zhu, Yanqiao, et al. "Graph Contrastive Learning with Adaptive Augmentation." WWW 2021.

Contrastive Methods

Local-to-global

- Contrast pairs: node embedding vs graph embedding
- Deep Graph Infomax[1]



Augmentation:
node feature shuffling + edge perturbation

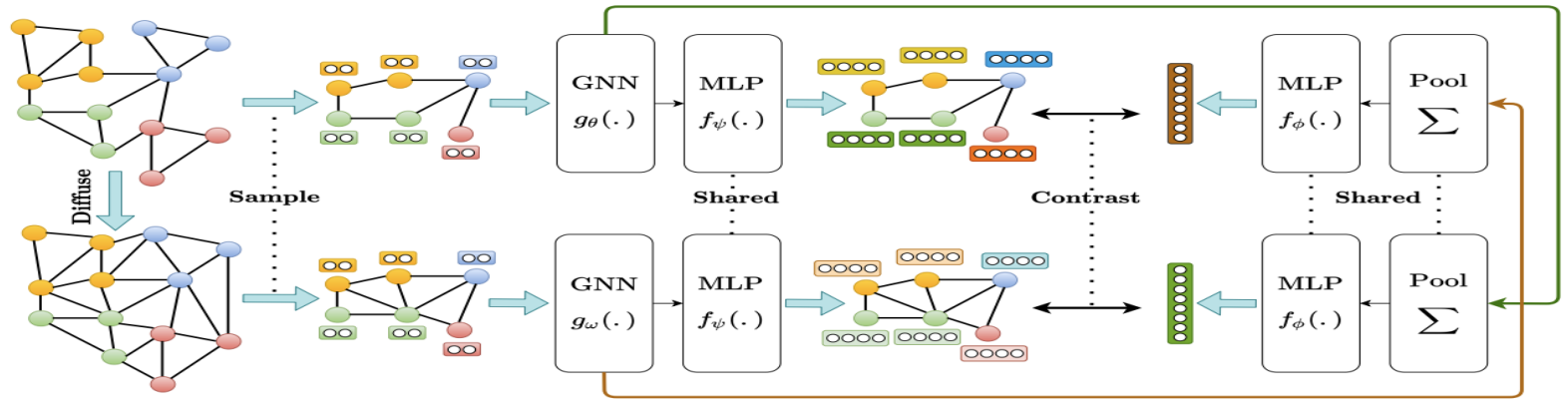
Loss: $$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\mathcal{D}\left(\vec{h}_i,\vec{s}\right)\right]+\sum_{j=1}^{M}\mathbb{E}_{(\tilde{\mathbf{X}},\tilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\tilde{h}}_j,\vec{s}\right)\right)\right]\right)$$

[1] Veličković, Petar, et al. "Deep Graph Infomax." ICLR 2019.

Contrastive Methods

Local-to-global

- Contrast pairs: node embedding vs graph embedding
- Multi-view contrastive learning[1]



View Augmentation | **Encoding** | **Contrast Pairs**

Augmentation:
Graph diffusion + subgraph sampling

Loss:

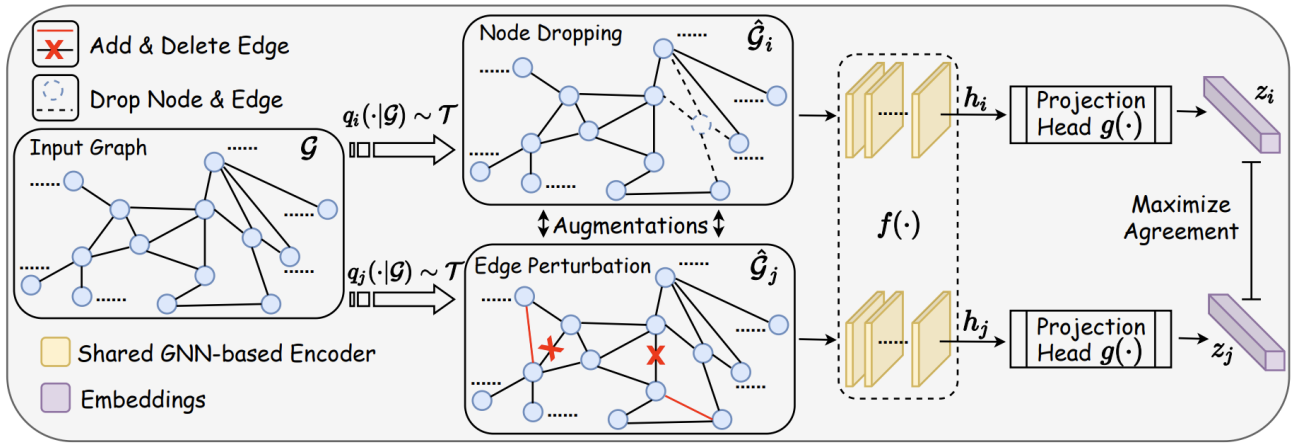
$$\max_{\theta,\omega,\phi,\psi} \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left[\frac{1}{|g|} \sum_{i=1}^{|g|} \left[\mathrm{MI}\left(\vec{h}_i^{\alpha}, \vec{h}_g^{\beta}\right) + \mathrm{MI}\left(\vec{h}_i^{\beta}, \vec{h}_g^{\alpha}\right) \right] \right]$$

[1] Hassani, Kaveh, and Amir Hosein Khasahmadi. "Contrastive Multi-View Representation Learning on Graphs." ICML 2020..

Contrastive Methods

Global-to-global

- Contrast pairs: graph embedding vs graph embedding
- GraphCL[1]



Augmentation:
Random node dropping + edge perturbation

Loss:

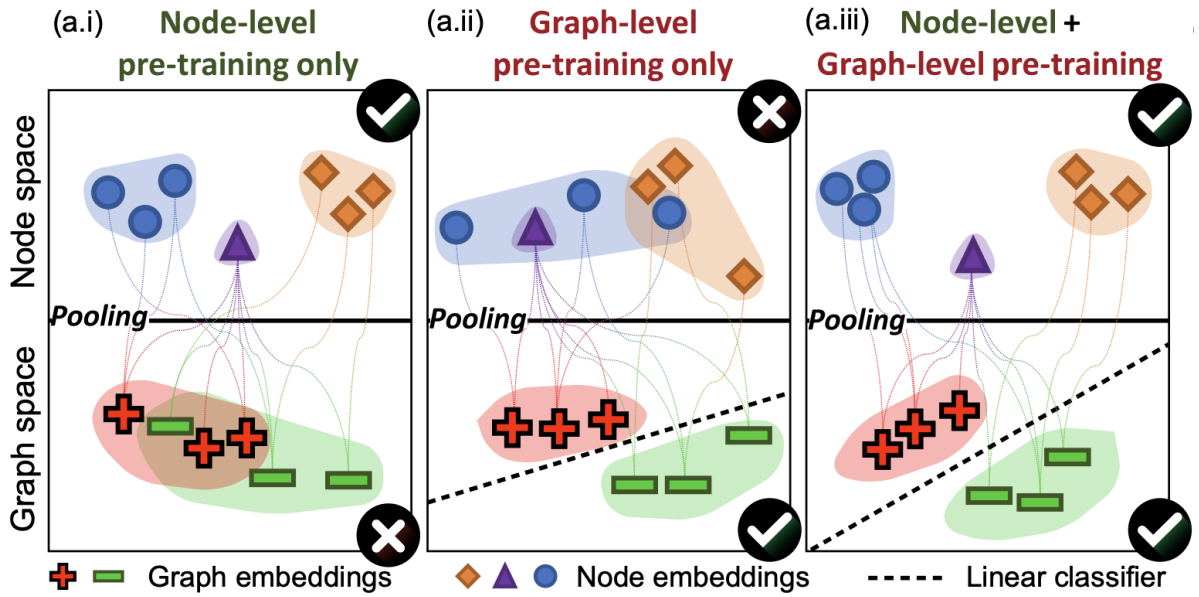
$$\ell_n = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n,j})/\tau)}{\sum_{n'=1,n'\neq n}^{N} \exp(\text{sim}(\boldsymbol{z}_{n,i}, \boldsymbol{z}_{n',j})/\tau)}$$

[1] You, Yuning, et al. "Graph Contrastive Learning with Augmentations." NeurIPS 2020.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Intuition: inject diverse capability into a single model



Multi-task methods[1]

August 3–7, 2025
KDD2025

[1] Hu, Weihua., et al. "Strategies For Pre-training Graph Neural Networks." ICLR 2020.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Intuition: inject diverse capability into a single model
- Node-level & graph-level
- Attribute prediction & structure prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

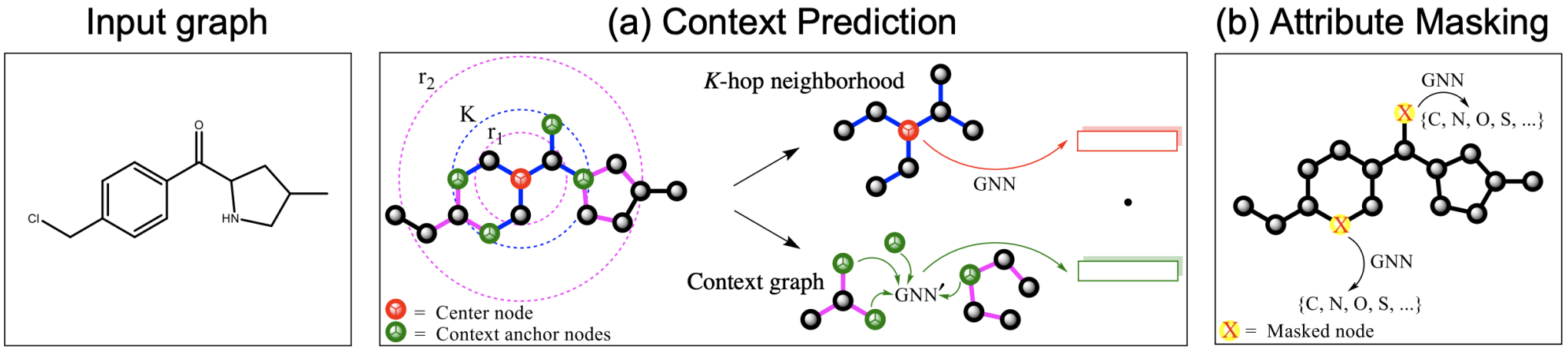
Different pre-training tasks[1]

[1] Hu, Weihua., et al. "Strategies For Pre-training Graph Neural Networks." ICLR 2020.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Intuition: inject diverse capability into a single model

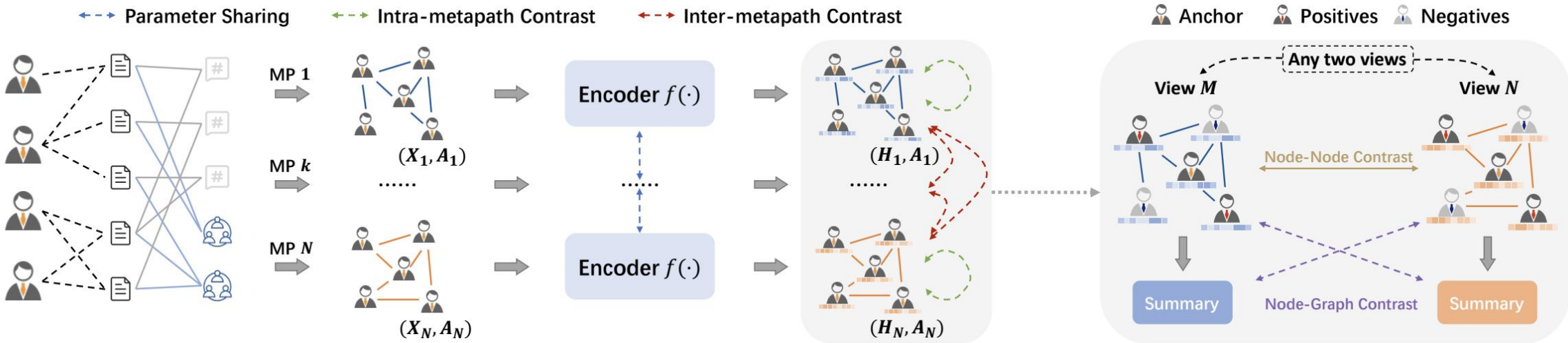


[1] Hu, Weihua., et al. "Strategies For Pre-training Graph Neural Networks." ICLR 2020.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Heterogeneous graphs: Multi-view contrastive learning[1]
- Use meta-paths on heterogeneous graphs to generate multiple views



View Augmentation | Encoding | Contrast Pairs (local-to-local &local-to-global)

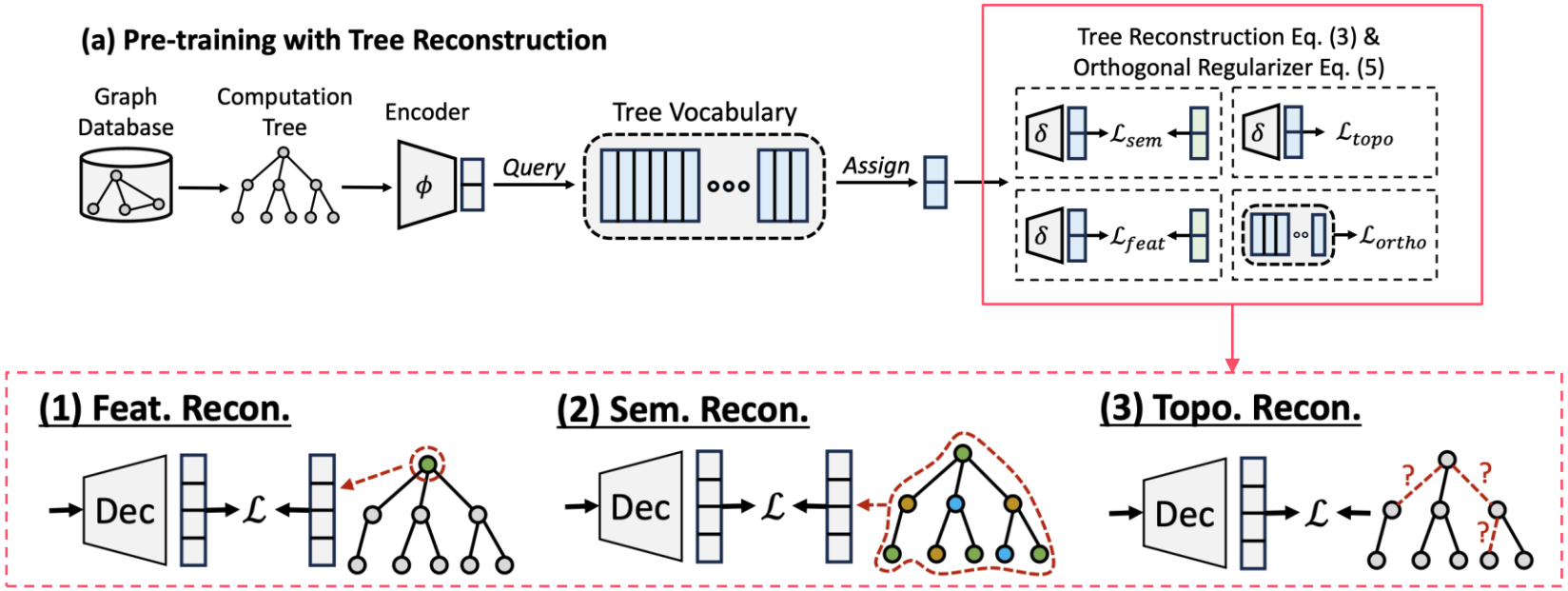
August 3–7, 2025 KDD2025

[1] Wang, Zehong, et al. "Heterogeneous Graph Contrastive Multi-view Learning." SDM 2023.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- GFT: Graph Foundation Model with Transferable Tree Vocabulary[1]



	Node	Link	Graph	Avg.
Different Pre-training Tasks				
n/a	72.52	47.30	73.83	66.54
w. $\mathcal{L}_{sem}$	76.25	90.39	74.99	79.47
w. $\mathcal{L}_{feat}$	75.85	90.42	74.42	79.13
w. $\mathcal{L}_{topo}$	75.50	90.28	74.57	78.96
GFT	**76.78**	**90.82**	**75.29**	**79.92**

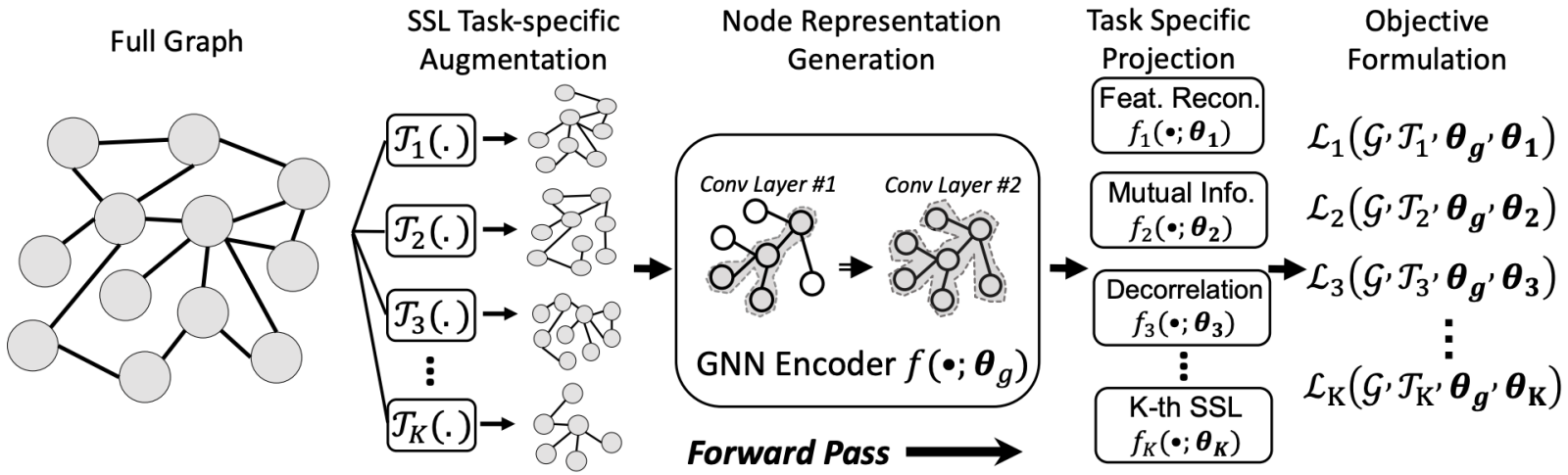
Jointly optimizing three-level tasks facilitates performance

[1] Wang, Zehong, et al. "GFT: graph foundation model with transferable tree vocabulary." NeurIPS 2024.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization[1]



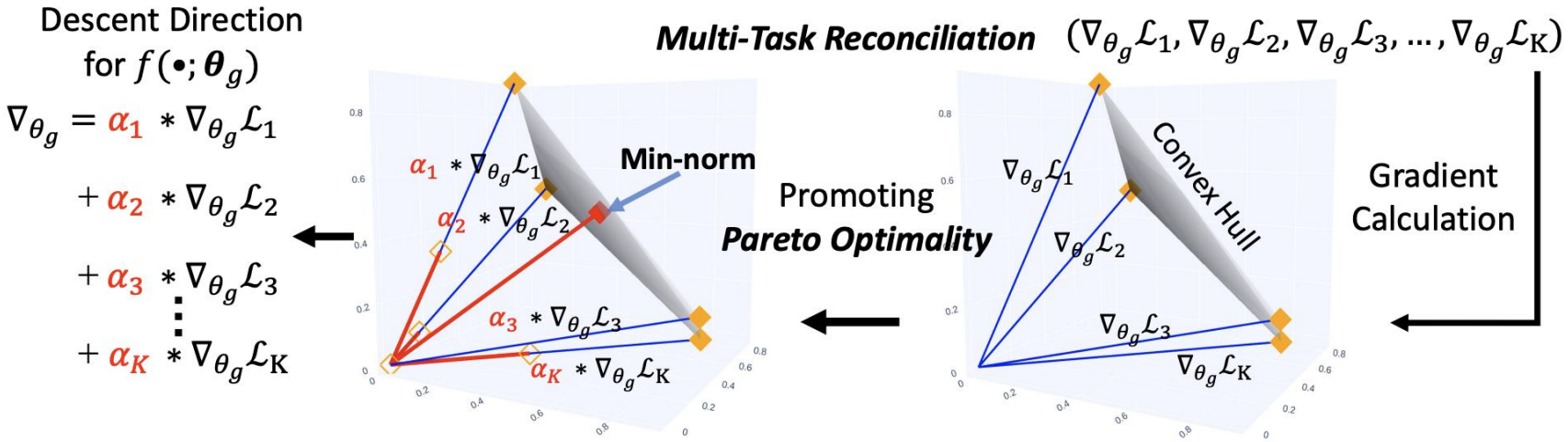
Multi-task learning on graphs

[1] Ju, Mingxuan, et al. "Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization." ICLR. 2023.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization[1]



Multi-task reconciliation via Parato optimization

[1] Ju, Mingxuan, et al. "Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization." ICLR. 2023.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization[1]

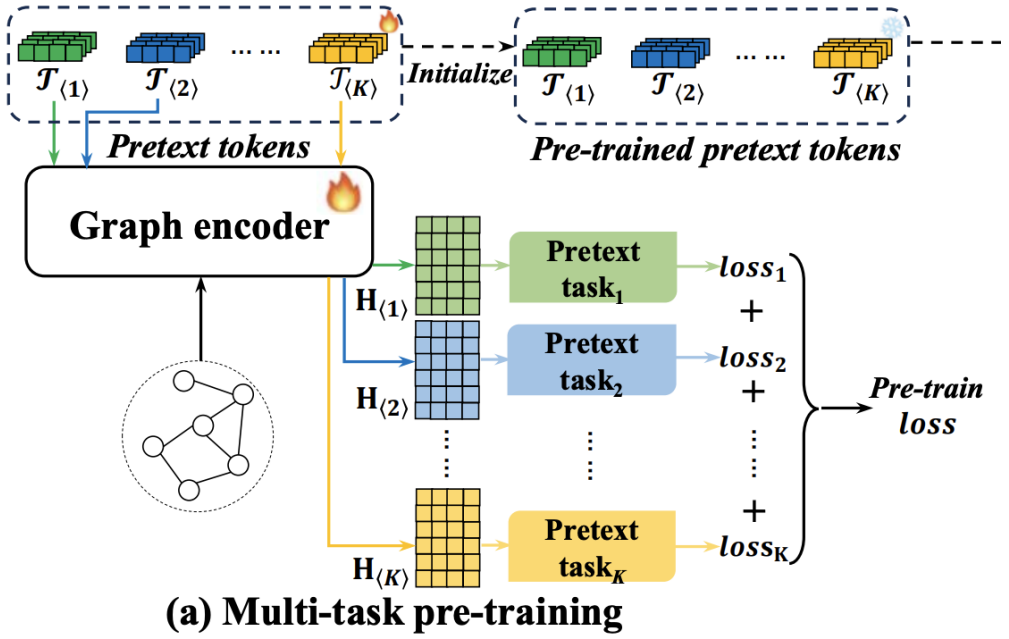
Method	WIKI.CS	PUBMED	AM.PHOTO	AM.COMP.	CO.CS	CO.PHY.	CHAM.	SQUIRREL	ACTOR	RANK
AVERAGE PERFORMANCE										
FeatRec	74.06	69.48	84.76	77.76	86.61	75.20	64.66	52.43	31.63	4.6
TopoRec	70.44	66.32	85.18	79.41	85.23	77.45	61.20	52.89	38.56	5.0
RepDecor	69.54	67.42	83.74	78.49	84.49	78.31	58.98	52.19	36.28	5.8
MI-NG	71.89	67.35	85.33	79.95	83.01	75.00	63.72	49.56	30.60	5.7
MI-NSG	75.59	69.25	82.99	80.85	86.02	80.30	64.69	53.89	38.22	3.6
PARETOGNN	**76.03**	**72.48**	**86.58**	**82.57**	**87.80**	**83.35**	**65.21**	**55.31**	**40.76**	**1.0**
w/o Pareto	74.64	69.82	85.82	82.09	86.54	82.32	64.37	54.90	40.12	2.4

[1] Ju, Mingxuan, et al. "Multi-task Self-supervised Graph Neural Networks Enable Stronger Task Generalization." ICLR. 2023.

Multi-task Methods

Jointly optimize multiple pre-training tasks

- MultiGPrompt[1]: multi-task pre-training for better task adaptation in prompting



(a) Multi-task pre-training

[1] Yu, Xingtong, et al. "MultiGPrompt for Multi-Task Pre-Training and Prompting on Graphs." WWW 2024.

Section 3
Techniques in Graph Prompting

Xingbo Fu

PhD Candidate

University of Virginia

Problem Formulation of Graph Prompting

A graph learning model f is trained through a pre-training task $\mathcal{L}_{\mathrm{PT}}$ via self-supervised learning. During the prompting stage, graph prompting trains learnable prompts $\mathcal{P}$ to adapt the pre-trained graph learning model f to a specific downstream task $\mathcal{L}_{\mathrm{DT}}$.

Problem Formulation of Graph Prompting

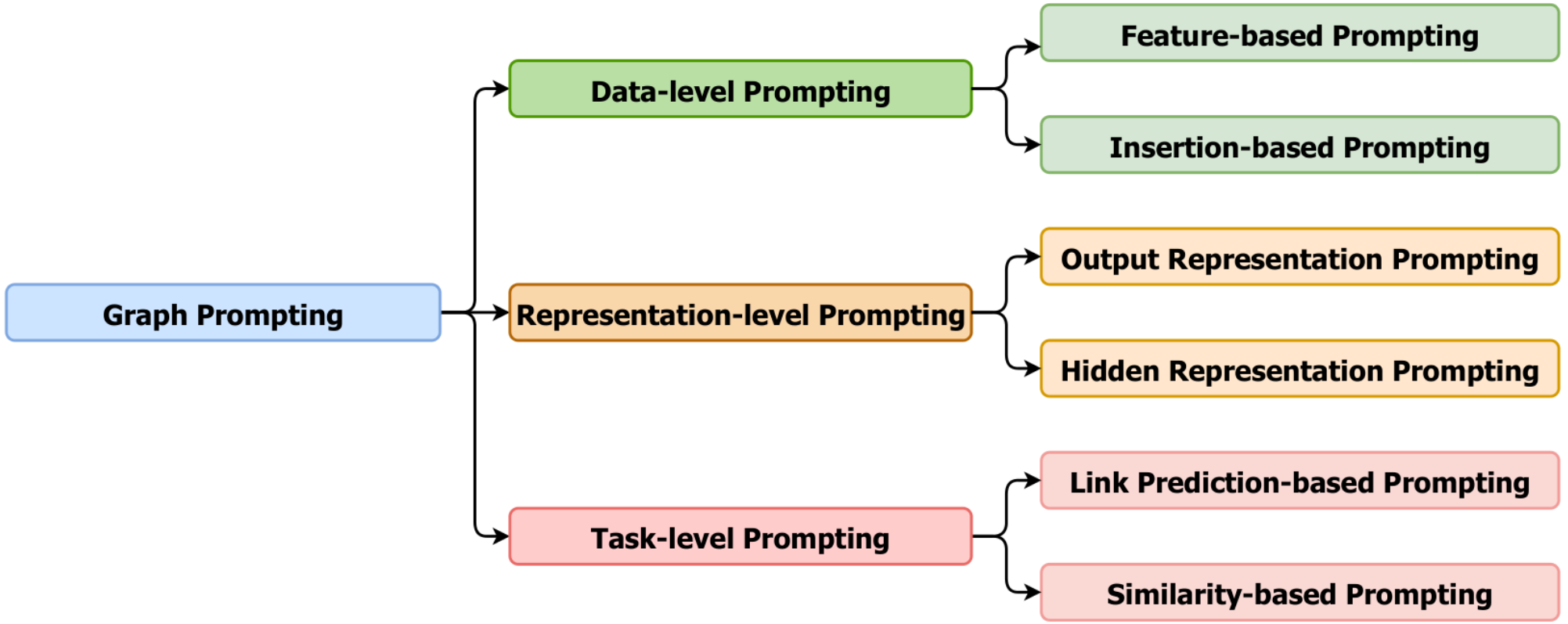
Pre-training compatibility

A graph learning model f is trained through a pre-training task $\mathcal{L}_{\mathrm{PT}}$ via self-supervised learning. During the prompting stage, graph prompting trains learnable prompts $\mathcal{P}$ to adapt the pre-trained graph learning model f to a specific downstream task $\mathcal{L}_{\mathrm{DT}}$.

Freeze pre-trained graph learning models

Downstream task universality

Taxonomy of Graph Prompting Techniques



Data-level Prompting

Learn trainable prompts at the data level

- Intuition: modify the input graph data to fit specific downstream tasks
- Formulation

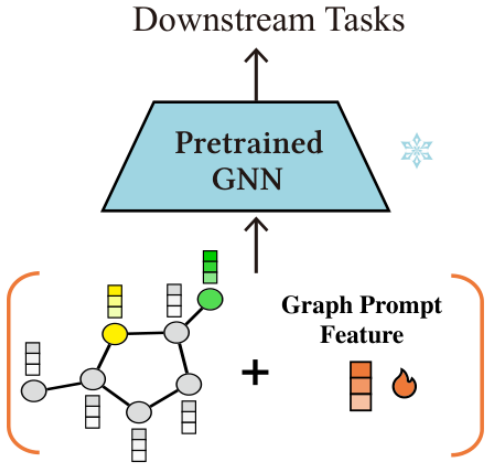
> *Given the input graph* $\mathcal{G} = (\mathbf{A}, \mathbf{X})$*, data-level prompting* $\mathcal{T}^D$ *transforms it into a prompted graph* $\tilde{\mathcal{G}} = (\tilde{\mathbf{A}}, \tilde{\mathbf{X}}) = \mathcal{T}^D((\mathbf{A}, \mathbf{X}), \mathcal{P})$ *with learnable prompts* $\mathcal{P}$ *for adaptation.*

- Two categories
- Feature-based prompting
- Insertion-based prompting

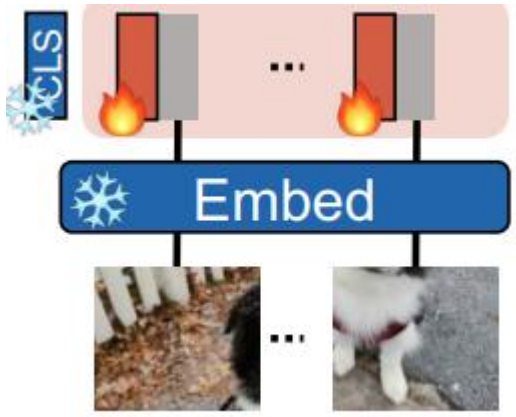
Data-level Prompting

Feature-based prompting

- Intuition: solely modify the feature matrix by learning prompt features
- Inspired by prompting techniques in NLP and CV
- The graph topology remains unchanged → $\tilde{\mathcal{G}} = \left(\mathbf{A}, \tilde{\mathbf{X}}\right) = \mathcal{T}^D\left((\mathbf{A}, \mathbf{X}), \mathcal{P}\right)$



Graph prompt feature (GPF)[1]



Visual prompt tuning (VPT)[2]

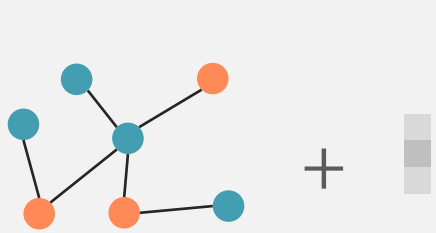
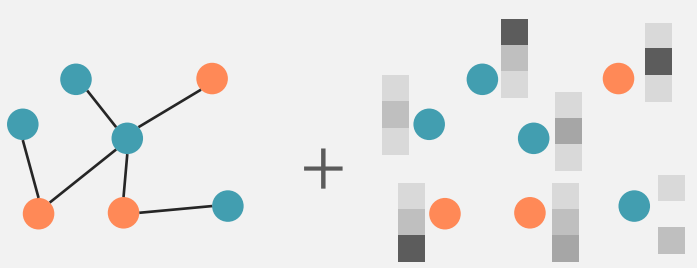
August 3–7, 2025
KDD2025

[1] Fang, Taoran, et al. "Universal Prompt Tuning for Graph Neural Networks." NeurIPS. 2023.
[2] Jia, Menglin, et al. "Visual Prompt Tuning." ECCV 2022.

Data-level Prompting

Feature-based prompting

- Goal: learn $\mathbf{p}_i$ for $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{p}_i$
- Shared prompt features v.s. customized prompt features

Shared prompt features	Customized prompt features
	
Fewer parameters	Increasing parameters in large-scale graphs
Easy to train	Prone to overfitting
Limited capability	More powerful

Data-level Prompting

Feature-based prompting

- Customized prompt features with basis vectors[1]

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{p}_i = \mathbf{x}_i + \sum_{m=1}^{M} \alpha_{i,m} \cdot \mathbf{b}_m$$

- GPF-plus[1]: $\alpha_{i,m}$ as the softmax values of projected node feature $\mathbf{x}_i$

$$\alpha_{i,m} = \frac{\exp\left(a_m^{\mathrm{T}} \mathbf{x}_i\right)}{\sum_{k=1}^{M} \exp\left(a_k^{\mathrm{T}} \mathbf{x}_i\right)}$$

- SUPT[2]: $\boldsymbol{\alpha}$ as the resulting scores derived from simple GNNs

$$\boldsymbol{\alpha} = \widehat{\mathbf{A}}^m \left(\mathbf{X} \oplus \sum_{m=1}^{M} \mathbf{b}_m \right) \mathbf{W}$$

$$\widehat{\mathbf{A}} = (\mathbf{D} + \mathbf{I})^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}) (\mathbf{D} + \mathbf{I})^{-\frac{1}{2}}$$

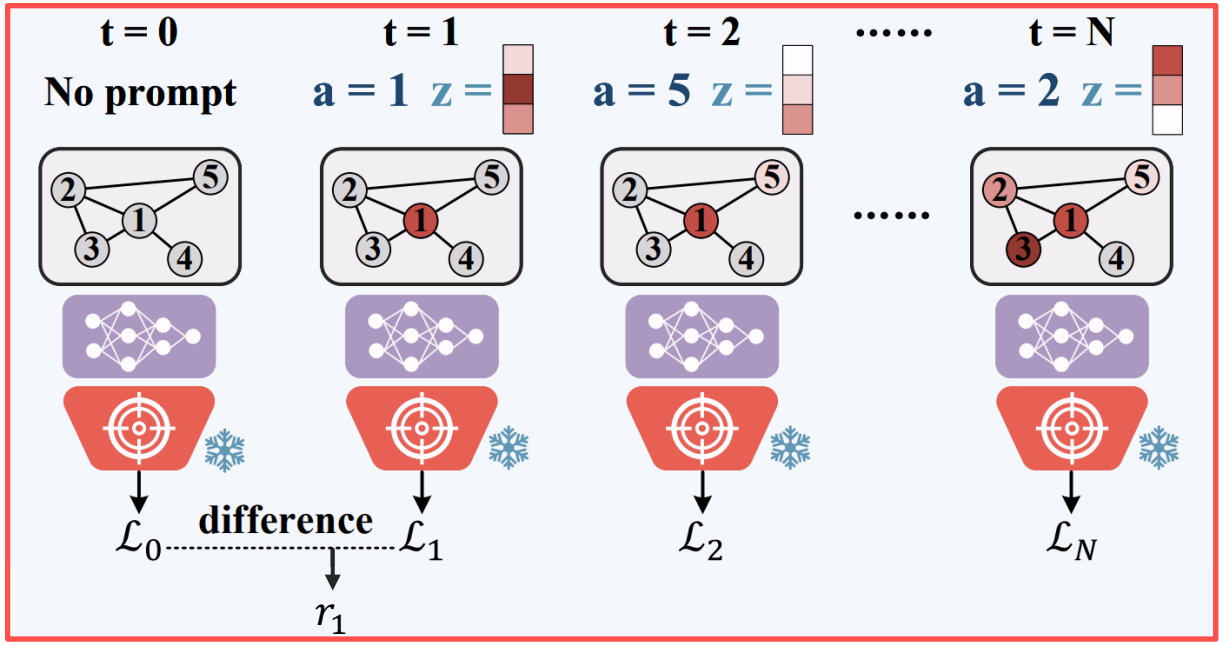
August 3–7, 2025
KDD2025

[1] Fang, Taoran, et al. "Universal Prompt Tuning for Graph Neural Networks." NeurIPS. 2023.
[2] Lee, Junhyun, et al. "Subgraph-level Universal Prompt Tuning." arXiv preprint arXiv:2402.10380 (2024).

Data-level Prompting

Feature-based prompting

- Customized prompt features by reinforcement learning with hybrid action space[1]
- Hybrid actions: discrete node index to prompt and its corresponding continuous prompt features
- States: node representations
- Reward function: instant loss decrease
- Policy network architecture: H-PPO with two parallel actor networks and a single critic network



August 3–7, 2025
KDD2025

[1] Zhu, Jiapeng, et al. "RELIEF: Reinforcement Learning Empowered Graph Feature Prompt Tuning." KDD 2025.

Data-level Prompting

Feature-based prompting

- Performance evaluation[1]
- GPF+: prompts based on features may not obtain sufficient information
- SUPT: prompts based on both features and structures can improve performance
- RELIEF: selective methods are better than learning on all the nodes

	Tuning Strategy	BBBP	Tox21	ToxCast	SIDER
Infomax	FT	65.26 ±1.05	71.54 ±0.73	57.98 ±0.42	53.45 ±0.49
	GPF	66.30 ±0.94	71.26 ±0.30	58.33 ±0.27	53.65 ±0.34
	GPF-plus	66.12 ±1.27	71.54 ±0.63	58.43 ±0.27	53.76 ±1.15
	$SUPT_{soft}$	66.45 ±0.85	**71.82** ±0.16	58.71 ±0.44	53.72 ±0.45
	$SUPT_{hard}$	66.14 ±0.85	71.55 ±0.37	58.65 ±0.33	53.82 ±0.49
	RELIEF	**67.93** ±0.73	71.58 ±0.13	**58.78** ±0.17	**53.95** ±0.40
AttrMasking	FT	66.48 ±0.44	72.32 ±0.19	57.35 ±0.42	54.62 ±0.58
	GPF	66.67 ±0.60	72.31 ±0.30	58.01 ±0.27	55.65 ±1.92
	GPF-plus	66.29 ±0.36	72.75 ±0.38	57.91 ±0.38	55.05 ±1.20
	$SUPT_{soft}$	66.28 ±0.81	**72.76** ±0.41	58.28 ±0.34	54.70 ±1.03
	$SUPT_{hard}$	66.93 ±0.91	72.75 ±0.41	58.18 ±0.44	55.20 ±1.26
	RELIEF	**67.18** ±0.55	72.40 ±0.32	**58.41** ±0.14	**56.54** ±0.75
ContextPred	FT	62.82 ±0.83	70.11 ±0.38	57.68 ±0.69	56.68 ±0.78
	GPF	61.65 ±0.76	70.42 ±0.29	58.51 ±0.38	56.55 ±0.46
	GPF-plus	61.25 ±0.73	70.12 ±0.42	57.64 ±0.70	56.86 ±0.52
	$SUPT_{soft}$	61.85 ±1.56	70.37 ±0.14	57.82 ±0.37	56.57 ±0.50
	$SUPT_{hard}$	62.29 ±1.41	70.40 ±0.15	58.06 ±0.39	56.34 ±0.72
	RELIEF	**62.99** ±0.67	**70.51** ±0.14	**58.58** ±0.04	**56.89** ±0.25
GCL	FT	62.13 ±1.66	61.35 ±0.88	53.96 ±0.80	52.63 ±0.71
	GPF	61.58 ±1.81	59.92 ±1.29	54.44 ±0.31	51.21 ±0.56
	GPF-plus	62.19 ±1.45	60.13 ±0.52	54.43 ±0.43	50.90 ±0.78
	$SUPT_{soft}$	63.96 ±0.85	60.13 ±0.52	54.57 ±0.76	51.44 ±1.24
	$SUPT_{hard}$	64.15 ±0.96	60.56 ±0.20	54.72 ±0.71	51.60 ±0.85
	All in One	62.90 ±3.77	**61.49** ±0.96	54.72 ±1.03	52.73 ±0.84
	RELIEF	**64.80** ±0.69	61.45 ±0.35	**55.03** ±0.39	**52.82** ±0.59

August 3–7, 2025
KDD2025

[1] Zhu, Jiapeng, et al. "RELIEF: Reinforcement Learning Empowered Graph Feature Prompt Tuning." KDD 2025.

Data-level Prompting

Feature-based prompting

- Quantifying Prompts Impact[1]
- Prompt Coverage Ratio (PCR): the proportion of nodes prompted at least once during prompting
- Average Prompt Magnitude (APM): the absolute values of all entries in prompts

Tuning Strategy	PCR	APM (10^{-2})	**OV.** (10^{-2})	ROC-AUC
GPF	1.00	7.15	7.15	62.08
GPF-plus	1.00	6.69	6.69	62.38
$\text{SUPT}_{\text{soft}}$	1.00	6.46	6.46	62.39
$\text{SUPT}_{\text{hard}}$	0.65	6.10	3.97	62.68
RELIEF	0.61	6.03	3.68	63.72

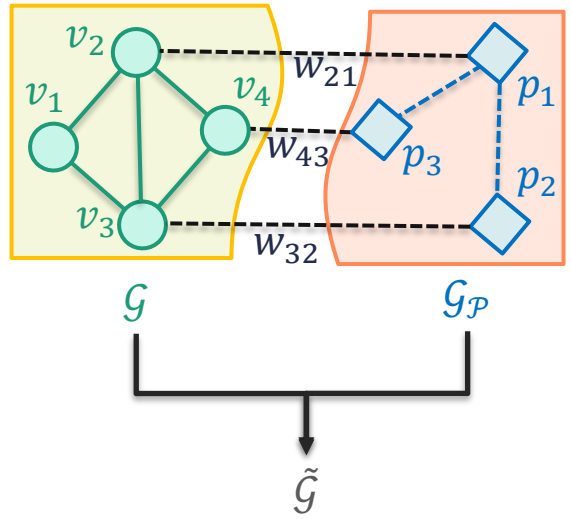
No need for graph prompting on every node!

[1] Zhu, Jiapeng, et al. "RELIEF: Reinforcement Learning Empowered Graph Feature Prompt Tuning." KDD 2025.

Data-level Prompting

Insertion-based prompting

- Intuition: insert additional prompt nodes as learnable prompts into the original graph
- $\tilde{\mathcal{G}}$ includes the prompt nodes and the original graph nodes
- $\mathcal{P} = \{p_1, p_2, \cdots, p_M\}$: M learnable prompts as the feature vectors of the prompt nodes



- Key challenges
- The connection among the prompt nodes
- The connection among the prompt nodes and the original graph nodes

Data-level Prompting

The connection among the prompt nodes

- Solution 1: free learnable parameters a_{ij} indicating how possible p_i and p_j should be connected[1]
- Solution 2: the dot product of each prompt node pair
- Connect p_i and p_j if $\sigma(\mathbf{p}_i \cdot \mathbf{p}_j^{\mathrm{T}}) > \delta$[1]
- Solution 3: treat the prompt nodes as independent[1,2]

The connection among the prompt nodes and the original graph nodes

- Solution 1: the dot product between a prompt node and an original graph node
- $w_{ij} = \sigma(\mathbf{x}_i \cdot \mathbf{p}_j^{\mathrm{T}}) > \delta$ if $\sigma(\mathbf{x}_i \cdot \mathbf{p}_j^{\mathrm{T}}) > \delta$[1]
- Solution 2: $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \sum_{m=1}^{M} \mathbf{p}_m$[1]
- Solution 3: $N \times M$ free learnable parameters[2]

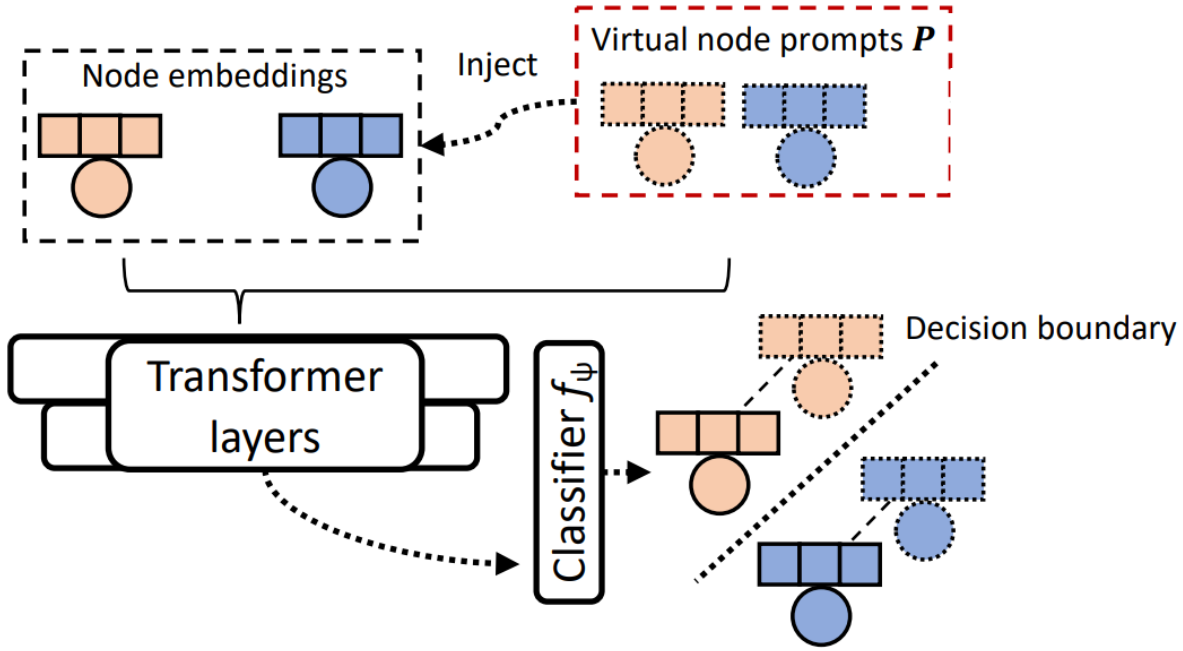
August 3–7, 2025
KDD2025

[1] Sun, Xiangguo, et al. "All in One: Multi-task Prompting for Graph Neural Networks." KDD 2023.
[2] Ge, Qingqing, et al. "PSP: Pre-training and Structure Prompt Tuning for Graph Neural Networks." ECML PKDD. 2024.

Data-level Prompting

Insertion-based prompting for Graph Transformers → ***No worries about edge connections!***

- VNT[1]: directly inject prompt nodes in the input of Graph Transformers



[1] Tan, Zhen, et al. "Virtual Node Tuning for Few-shot Node Classification." KDD 2023.

Representation-level Prompting

Learn trainable prompts at the representation level

- Intuition: apply learnable prompts to node representations
- Formulation

> *Given the (hidden) representation matrix* $\mathbf{H}^{(l)}$ *at the* $l-$*th layer, representation-level prompting* $\mathcal{T}^R$ *transforms it into a prompted representation matrix* $\widetilde{\mathbf{H}}^{(l)} = \mathcal{T}^R\left(\mathbf{H}^{(l)}, \mathcal{P}\right)$ *with learnable prompts* $\mathcal{P}$ *for adaptation.*

- Two categories
- Output representation prompting
- Hidden representation prompting

Representation-level Prompting

Output representation prompting

- Intuition: directly modify the output representations after the final layer
- Only the output representations are modified: $\tilde{\mathbf{h}}^{(L)} = \mathcal{T}^R\left(\mathbf{h}^{(L)}, \mathcal{P}\right)$

- Straightforward design: GraphPrompt[1]
- Element-wise multiplication: $\tilde{\mathbf{h}}_v^{(L)} = \mathbf{p}_v \odot \mathbf{h}_v^{(L)}$
- $\mathbf{p}_v$ is shared by all the nodes
- Limitation: shared $\mathbf{p}_v$ is insufficient

[1] Liu, Zemin, et al. "Graphprompt: Unifying pre-training and downstream tasks for graph neural networks." WWW 2023.

Representation-level Prompting

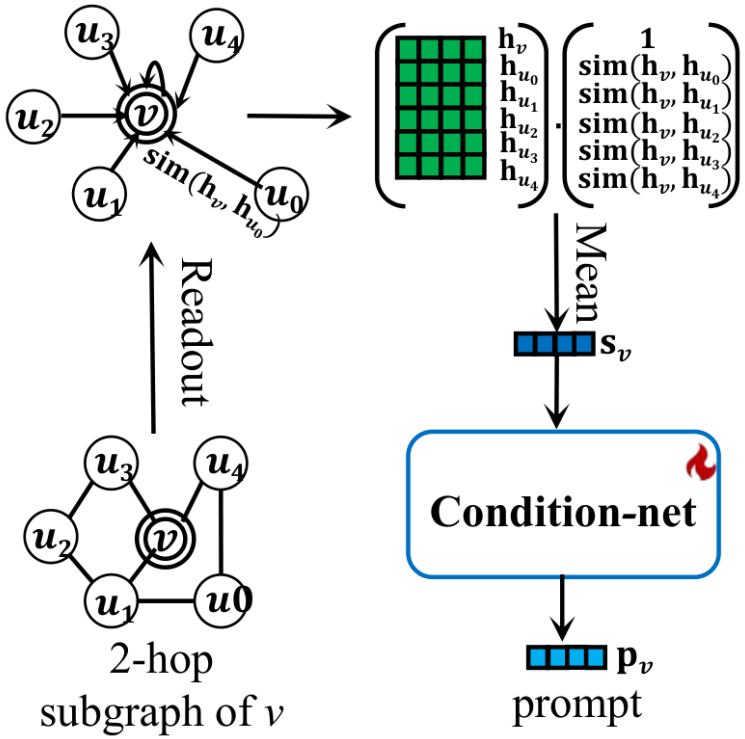
Output representation prompting

- Non-homophilic patterns of a node can be characterized by considering a multi-hop neighborhood around the node[1]

$$\mathbf{s}_v = \frac{1}{\mathcal{N}_2(v)} \sum_{u_j \in \mathcal{N}_2(v)} \mathbf{h}_v^{(L)} \cdot \mathrm{sim}\left(\mathbf{h}_v^{(L)}, \mathbf{h}_j^{(L)}\right)$$

- Obtain customized $\mathbf{p}_v$ through a condition-net[2]

$$\mathbf{p}_v = \varphi(\mathbf{s}_v) = \mathrm{CondNet}(\mathbf{s}_v)$$



ProNoG[1]

[1] Yu, Xingtong, et al. "Non-Homophilic Graph Pre-Training and Prompt Learning." KDD 2025.

[2] Zhou, Kaiyang, et al. "Conditional Prompt Learning for Vision-Language Models." CVPR 2022.

Representation-level Prompting

Output representation prompting

- ProNoG[1] performs well on non-homophilic graphs

Methods	Wisconsin	Squirrel	Chameleon	Cornell
GCN	21.39 ± 6.56	20.00 ± 0.29	25.11 ± 4.19	21.81 ± 4.71
GAT	28.01 ± 5.40	21.55 ± 2.30	24.82 ± 4.35	23.03 ± 13.19
H2GCN	23.60 ± 4.64	21.90 ± 2.15	25.89 ± 4.96	32.77 ± 14.88
FAGCN	35.03 ± 17.92	20.91 ± 1.79	22.71 ± 3.74	28.67 ± 17.64
DGI	28.04 ± 6.47	20.00 ± 1.86	19.33 ± 4.57	32.54 ± 15.66
GraphCL	29.85 ± 8.46	21.42 ± 2.22	27.16 ± 4.31	24.69 ± 14.06
DSSL	28.46 ± 10.31	20.94 ± 1.88	<u>27.92</u> ± 3.93	20.36 ± 5.38
GraphACL	<u>34.57</u> ± 10.46	<u>24.44</u> ± 3.94	26.72 ± 4.67	<u>33.17</u> ± 16.06
GPPT	27.39 ± 6.67	20.09 ± 0.91	24.53 ± 2.55	25.09 ± 2.92
GraphPrompt	31.48 ± 5.18	21.22 ± 1.80	25.36 ± 3.99	31.00 ± 13.88
GraphPrompt+	31.54 ± 4.54	21.24 ± 1.82	25.73 ± 4.50	31.65 ± 14.48
ProNoG	**44.72** ± 11.93	**24.59** ± 3.41	**30.67** ± 3.73	**37.90** ± 9.31

[1] Yu, Xingtong, et al. "Non-Homophilic Graph Pre-Training and Prompt Learning." KDD 2025.

Representation-level Prompting

Hidden representation prompting

- Intuition: modify node representations with layer-wise prompts
- More parameters provide more flexibility

- GraphPrompt+[1]: an advanced version of GraphPrompt
- Learnable prompts at each layer $\mathcal{P} = \{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \cdots, \mathbf{p}^{(\mathrm{L})}\}$
- Prompted representation fusion with learnable coefficients

$$\widetilde{\mathbf{H}} = \sum_{l=0}^{L} w^{(l)} \cdot \widetilde{\mathbf{H}}^{(l)}$$

[1] Yu, Xingtong, et al. "Generalized Graph Prompt: Toward a Unification of Pre-Training and Downstream Tasks on Graphs." TKDE 2024.

Representation-level Prompting

Hidden representation prompting

- EdgePrompt+[1]: learn edge prompts for each edge
- $\boldsymbol{e}_{ij}^{(l)}$: prompt vector on edge (v_i, v_j) at the l-th layer
- Without edge prompts

$$\mathbf{h}_i^{(l)} = \mathrm{COMB}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \mathrm{AGG}^{(l)}\left(\left\{\mathbf{h}_j^{(l-1)} : v_j \in \mathcal{N}(v_i)\right\}\right)\right)$$

- With edge prompts

$$\mathbf{h}_i^{(l)} = \mathrm{COMB}^{(l)}\left(\mathbf{h}_i^{(l-1)}, \mathrm{AGG}^{(l)}\left(\left\{\boldsymbol{h}_j^{(l-1)} : v_j \in \mathcal{N}(v_i)\right\}, \left\{\mathbf{e}_{ij}^{(l)} : v_j \in \mathcal{N}(v_i)\right\}\right)\right)$$

[1] Fu, Xingbo , et al. "Edge Prompt Tuning for Graph Neural Networks." ICLR 2025.

Representation-level Prompting

Hidden representation prompting

- EdgePrompt+[1]: learn edge prompts for each edge
- Compute edge prompts as the weighted average of the anchor prompts

$$\mathbf{e}_{ij}^{(l)} = \sum_{m=1}^{M} b_{ijm}^{(l)} \cdot \mathbf{p}_m^{(l)}$$

M anchor prompts at the l-th layer

$$\mathcal{P}^{(l)} = \left\{\mathbf{p}_1^{(l)}, \mathbf{p}_2^{(l)}, \cdots, \mathbf{p}_M^{(l)}\right\}$$

- $\mathbf{b}_{ij}^{(l)}$ is computed using a score function $\varphi^{(l)}$

$$\mathbf{b}_{ij}^{(l)} = \mathrm{Softmax}\left(\varphi^{(l)}\left(v_i, v_j\right)\right)$$

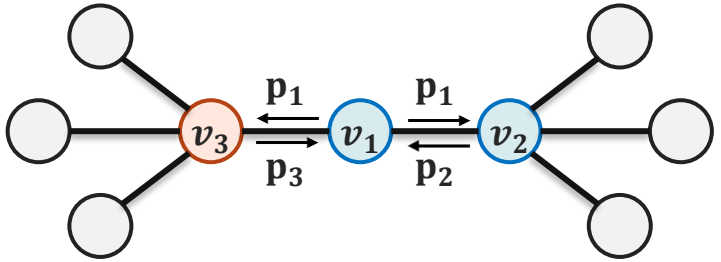
$$\varphi^{(l)}\left(v_i, v_j\right) = \mathrm{LeakyReLU}\left(\left[\mathbf{h}_i^{(l-1)} || \mathbf{h}_j^{(l-1)}\right] \cdot \mathbf{W}^{(l)}\right)$$

[1] Fu, Xingbo , et al. "Edge Prompt Tuning for Graph Neural Networks." ICLR 2025.

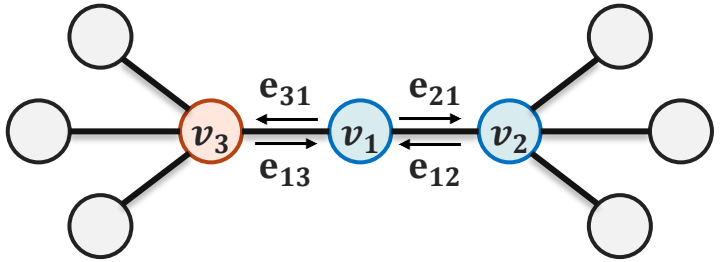
Representation-level Prompting

Hidden representation prompting

- Why EdgePrompt+ works?
- Prompting on nodes: add learnable prompts to node features or representations[1,2]
- Limitation: $\mathbf{p}_i$ will be uniformly aggregated by neighboring nodes



- Prompting on edges: add learnable prompts on edges
- $\mathbf{e}_{ij}$ provides customized prompts to different neighboring nodes
- Each node broadcasts distinct prompts to its neighbors



Representation-level Prompting

Hidden representation prompting

- EdgePrompt+ is more powerful than EdgePrompt

→ Customized edge prompts > shared edge prompts

- EdgePrompt+ is compatible with both generative and contrastive pre-training strategies
- GPF-plus is competitive among data-level prompting methods

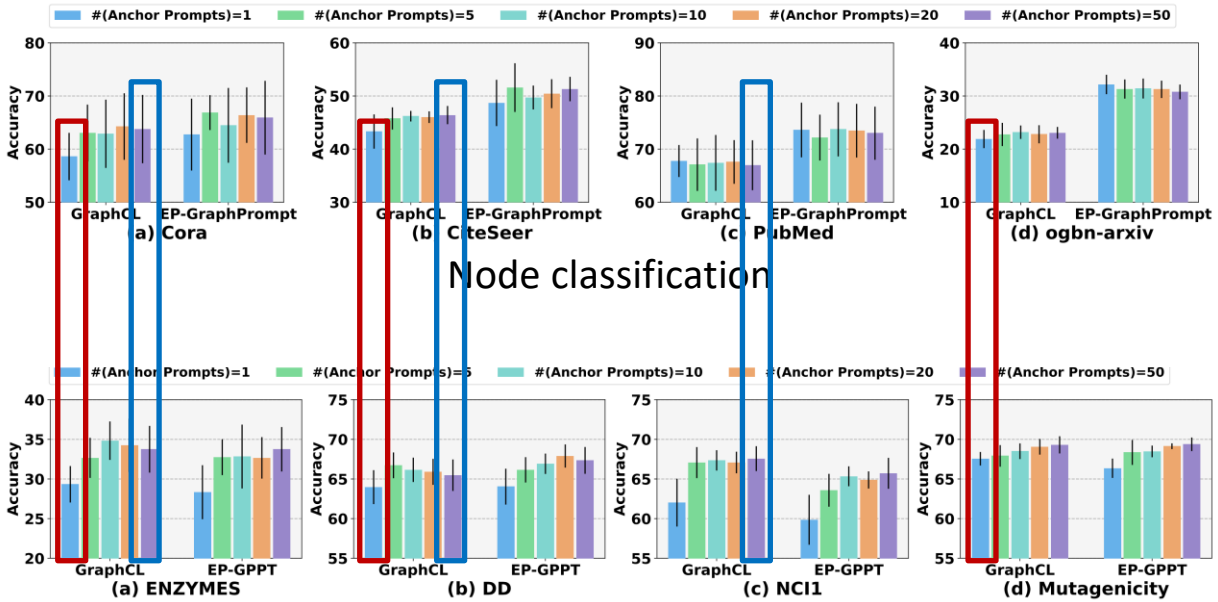
Pre-training Strategies	Tuning Methods	ENZYMES	DD	NCI1	NCI109	Mutagenicity
GraphCL	Classifier Only	$30.50_{\pm 1.16}$	$62.89_{\pm 2.19}$	$62.49_{\pm 1.95}$	$61.68_{\pm 0.93}$	$66.62_{\pm 1.87}$
	GraphPrompt	$27.83_{\pm 1.61}$	$64.33_{\pm 1.79}$	$63.19_{\pm 1.71}$	$62.18_{\pm 0.48}$	$\mathbf{67.62_{\pm 0.65}}$
	ALL-in-one	$25.92_{\pm 0.55}$	$66.54_{\pm 1.82}$	$57.52_{\pm 2.61}$	$62.74_{\pm 0.78}$	$63.43_{\pm 2.53}$
	GPF	$30.08_{\pm 1.25}$	$64.54_{\pm 2.22}$	$62.66_{\pm 1.83}$	$62.29_{\pm 0.90}$	$66.54_{\pm 1.85}$
	GPF-plus	$31.00_{\pm 1.50}$	$67.26_{\pm 2.29}$	$64.56_{\pm 1.10}$	$62.84_{\pm 0.22}$	$66.82_{\pm 1.63}$
	EdgePrompt	$29.50_{\pm 1.57}$	$64.16_{\pm 2.13}$	$63.05_{\pm 2.11}$	$62.59_{\pm 0.93}$	$66.87_{\pm 1.88}$
	EdgePrompt+	$\mathbf{34.00_{\pm 1.25}}$	$\mathbf{67.98_{\pm 2.05}}$	$\mathbf{66.30_{\pm 2.54}}$	$\mathbf{66.52_{\pm 0.91}}$	$67.47_{\pm 2.37}$
SimGRACE	Classifier Only	$27.07_{\pm 1.04}$	$61.77_{\pm 2.40}$	$61.27_{\pm 3.64}$	$62.12_{\pm 1.10}$	$67.36_{\pm 0.71}$
	GraphPrompt	$26.87_{\pm 1.47}$	$62.58_{\pm 1.84}$	$62.45_{\pm 1.52}$	$62.41_{\pm 0.69}$	$68.03_{\pm 0.78}$
	ALL-in-one	$25.73_{\pm 1.18}$	$65.16_{\pm 1.47}$	$58.52_{\pm 1.59}$	$62.01_{\pm 0.66}$	$64.43_{\pm 1.00}$
	GPF	$28.53_{\pm 1.76}$	$65.64_{\pm 0.70}$	$61.45_{\pm 3.13}$	$61.90_{\pm 1.26}$	$67.19_{\pm 0.74}$
	GPF-plus	$27.33_{\pm 2.01}$	$67.20_{\pm 1.56}$	$61.61_{\pm 2.89}$	$62.84_{\pm 0.23}$	$67.69_{\pm 0.64}$
	EdgePrompt	$29.33_{\pm 2.30}$	$63.97_{\pm 2.14}$	$62.02_{\pm 3.02}$	$62.02_{\pm 1.03}$	$67.55_{\pm 0.85}$
	EdgePrompt+	$\mathbf{32.67_{\pm 2.53}}$	$\mathbf{67.72_{\pm 1.62}}$	$\mathbf{67.07_{\pm 1.96}}$	$\mathbf{66.53_{\pm 1.30}}$	$\mathbf{68.31_{\pm 1.36}}$
EP-GPPT	Classifier Only	$29.08_{\pm 1.35}$	$62.12_{\pm 2.82}$	$56.85_{\pm 4.35}$	$62.27_{\pm 0.78}$	$66.30_{\pm 1.78}$
	GraphPrompt	$26.67_{\pm 1.60}$	$61.61_{\pm 1.91}$	$58.77_{\pm 0.97}$	$62.16_{\pm 0.89}$	$66.37_{\pm 1.17}$
	ALL-in-one	$24.92_{\pm 1.33}$	$63.61_{\pm 2.12}$	$59.14_{\pm 2.12}$	$59.70_{\pm 1.37}$	$64.86_{\pm 1.60}$
	GPF	$28.33_{\pm 1.73}$	$63.48_{\pm 2.08}$	$58.14_{\pm 4.16}$	$62.52_{\pm 1.39}$	$66.10_{\pm 0.96}$
	GPF-plus	$29.25_{\pm 1.30}$	$\mathbf{66.92_{\pm 2.34}}$	$62.93_{\pm 3.23}$	$64.13_{\pm 1.42}$	$67.57_{\pm 1.45}$
	EdgePrompt	$28.33_{\pm 3.41}$	$64.03_{\pm 2.26}$	$59.85_{\pm 3.15}$	$62.98_{\pm 1.44}$	$66.36_{\pm 1.22}$
	EdgePrompt+	$\mathbf{32.75_{\pm 2.26}}$	$66.16_{\pm 1.60}$	$\mathbf{63.58_{\pm 2.07}}$	$\mathbf{65.15_{\pm 1.60}}$	$\mathbf{68.35_{\pm 1.57}}$
EP-GraphPrompt	Classifier Only	$31.33_{\pm 3.22}$	$62.58_{\pm 2.40}$	$62.09_{\pm 2.31}$	$60.19_{\pm 1.71}$	$65.13_{\pm 0.81}$
	GraphPrompt	$30.20_{\pm 1.93}$	$64.72_{\pm 1.98}$	$62.57_{\pm 1.45}$	$62.32_{\pm 0.95}$	$65.85_{\pm 0.65}$
	ALL-in-one	$29.07_{\pm 1.16}$	$65.60_{\pm 2.38}$	$58.67_{\pm 2.42}$	$57.69_{\pm 1.08}$	$64.66_{\pm 0.76}$
	GPF	$30.93_{\pm 1.76}$	$66.21_{\pm 1.66}$	$61.80_{\pm 2.78}$	$62.27_{\pm 1.18}$	$65.61_{\pm 0.59}$
	GPF-plus	$30.67_{\pm 3.06}$	$\mathbf{67.50_{\pm 2.45}}$	$62.59_{\pm 2.09}$	$61.98_{\pm 1.60}$	$65.51_{\pm 1.10}$
	EdgePrompt	$30.80_{\pm 2.09}$	$65.87_{\pm 1.35}$	$61.75_{\pm 2.49}$	$62.33_{\pm 1.65}$	$65.77_{\pm 0.90}$
	EdgePrompt+	$\mathbf{33.27_{\pm 2.71}}$	$67.47_{\pm 2.14}$	$\mathbf{65.06_{\pm 1.84}}$	$\mathbf{64.64_{\pm 1.57}}$	$\mathbf{66.42_{\pm 1.31}}$

Graph classification

Representation-level Prompting

Hidden representation prompting

- One anchor prompt is insufficient in most cases
- Two many anchor prompts may not improve performance (hard to train)



Task-level Prompting

Particularly classification tasks

Learn trainable prompts at the task level

- Intuition: reformulate downstream tasks into alternative forms
- Formulation

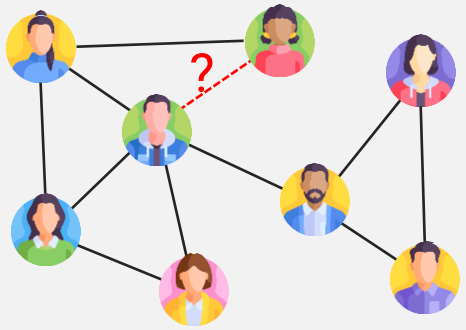
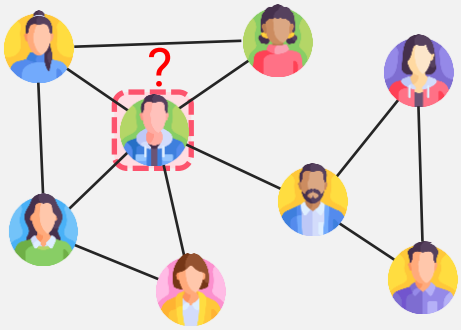
> *Given a downstream task $\mathcal{L}_{\mathrm{DT}}$ (e.g., node classification), task-level prompting $\mathcal{T}^T$ transforms it into a different task $\tilde{\mathcal{L}}_{\mathrm{DT}} = \mathcal{T}^T(\mathcal{L}_{\mathrm{DT}}, \mathcal{P})$ with learnable prompts $\mathcal{P}$ for adaptation.*

- Two categories
- Link prediction-based prompting
- Similarity-based prompting

Task-level Prompting

Link prediction-based prompting

- Recall: the gap between edge prediction for pre-training and node classification as the downstream task

Phase	Pre-training stage	Downstream stage
Illustration		
Task	Edge prediction	Node classification
Involved embeddings	A pair of nodes	A single node
Prediction	Edge probabilities	Node class probabilities

Can we convert node classification to edge prediction?

Task-level Prompting

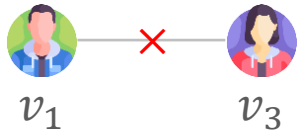
Link prediction-based prompting

- Intuition: convert node classification to edge prediction
- Pre-training stage: edge prediction with positive and negative pairs

Positive pair

$$g(v_1, v_2) = 1$$

Negative pair



$$g(v_1, v_2) = 0$$

Pre-training objective

$$\min_{\pi_{\text{GNN}}, \phi_{\text{Proj}}} \sum_{(v_i, v_j)} \mathcal{L}_{\text{CE}} \left(\phi_{\text{Proj}}(\mathbf{h}_i, \mathbf{h}_j), g(v_i, v_j) \right)$$

Task-level Prompting

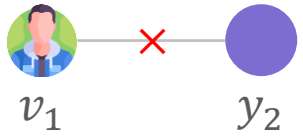
Link prediction-based prompting

- Intuition: convert node classification to edge prediction
- Downstream stage: construct positive and negative pairs using learnable task tokens

Positive pair

$$g(v_1, y_1) = 1$$

Negative pair



$$g(v_1, y_2) = 0$$

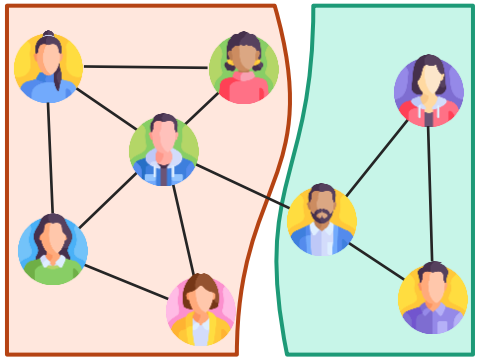
Downstream objective

$$\min_{\pi_{\text{GNN}}, \phi_{\text{Proj}}, \mathbf{E}} \sum_{(v_i, y_c)} \mathcal{L}_{\text{CE}} \left(\phi_{\text{Proj}}(\mathbf{h}_i, \mathbf{e}_c), g(v_i, y_c) \right)$$

Task-level Prompting

Link prediction-based prompting

- GPPT: one global task token $\mathbf{E}$ is insufficient for all nodes[1]
- Task tokens should vary with clusters
- For each cluster m, train an independent task token $\mathbf{E}^m$



$$\min_{\pi_{\text{GNN}},\phi_{\text{Proj}},\mathbf{E}^1,\cdots,\mathbf{E}^M} \sum_{(v_i,y_c)} \mathcal{L}_{\text{CE}}\left(\phi_{\text{Proj}}(\mathbf{h}_i, \mathbf{e}_c^m), g(v_i, y_c)\right)$$

- Use structure tokens to replace node embeddings: $\mathbf{e}_i = a_i \cdot \mathbf{h}_i + \sum_{v_j \in \mathcal{N}(v_i)} a_j \cdot \mathbf{h}_j$
- Final objective

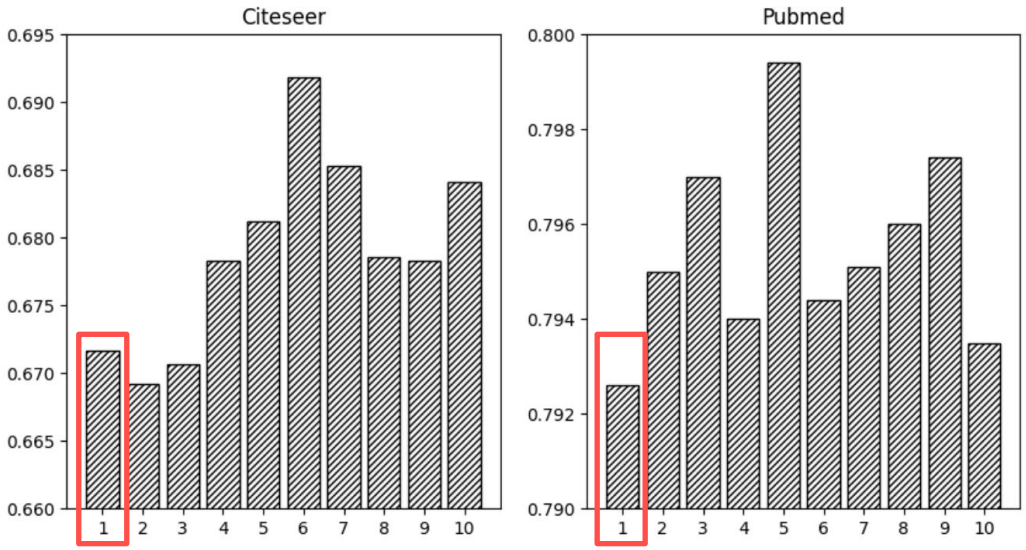
$$\min_{\pi_{\text{GNN}},\phi_{\text{Proj}},\mathbf{E}^1,\cdots,\mathbf{E}^M} \sum_{(v_i,y_c)} \mathcal{L}_{\text{CE}}\left(\phi_{\text{Proj}}(\mathbf{e}_i, \mathbf{e}_c^m), g(v_i, y_c)\right) + \lambda \sum_m \left\|\mathbf{E}^M (\mathbf{E}^M)^{\mathrm{T}} - \mathbf{I}\right\|_F^2$$

[1] Sun, Mingchen, et al. "GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks." KDD 2022.

Task-level Prompting

Link prediction-based prompting

- The impact of cluster numbers in GPPT[1]
- The performance of GPPT is significantly damaged with one cluster



Cluster number study

[1] Sun, Mingchen, et al. "GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks." KDD 2022.

Task-level Prompting

Similarity-based prompting

- Intuition: compare graph representations to class prototypes using contrastive loss

Class prototypes

$$\min_{\mathcal{P}} \sum_{(x,y)\in\mathcal{D}} -\log \frac{\exp\left(\text{sim}\left(\mathbf{h}_i, \mathbf{s}_y\right)/\tau\right)}{\sum_{c\in\mathcal{Y}} \exp\left(\text{sim}\left(\mathbf{h}_i, \mathbf{s}_c\right)/\tau\right)}$$

- How to get class prototypes?
- GraphPrompt, GraphPrompt+, ProNoG: use the average of instance representations belonging to class c
- HetGPT[1]: initialized using average representation and tuned while prompting

[1] Ma, Yihong, et al. "HetGPT: Harnessing the Power of Prompt Tuning in Pre-Trained Heterogeneous Graph Neural Networks." WWW 2024.

Comparison of Graph Prompting Techniques

Technique Categories	Prompting Stage	Prompting Strategies	Single Forward Pass	DT Universality
Data-level prompting	Graph data	Feature-based prompting	✗	✓
		Insertion-based prompting	✗	✓
Representation-level prompting	Node representations	Output representation prompting	✓	✓
		Hidden representation prompting	✗	✓
Task-level prompting	Downstream tasks	Link prediction-based prompting	✓	✗
		Similarity-based prompting	✓	✗

Section 4
Summary and Future Directions

Xingbo Fu

PhD Candidate

University of Virginia

Summary

Graph prompting

- Objective gap between pre-training and downstream tasks
- Example: link prediction → node classification

- Graph pre-training methods: the foundation step of graph prompting
- Generative methods, contrastive method, multi-task methods

- Mainstream techniques in graph prompting
- Data-level, representation-level, task-level techniques

Future Directions

Benchmarks and datasets

- Standard experimental settings for performance evaluation
- Cross-dataset & cross-domain evaluation
- One existing benchmark[1]

August 3–7, 2025 KDD2025

[1] Zi, Chenyi, et al. "ProG: A Graph Prompt Learning Benchmark." NeurIPS 2024.

Future Directions

Theoretical foundation

- Empirical improvements → theoretical guarantees

Theorem 3. *Given a GPF-like prompt vector p_ω, if a GCN model F_θ has non-linear function layers but the model's weight matrix is row full-rank, then there exists an optimal ω for any input graph G such that $P_\omega(G) \in B_G$.*

Theorem 4. *Given the All-in-One-like prompt graph SG_ω (a subgraph containing prompt tokens and token structures), if a GCN model F_θ does not have any non-linear transformations, or has non-linear layers but the model's weight matrix is row full-rank, then there exists an optimal ω for any input graph G such that $P_\omega(G) \in B_G$.*

The upper bound of graph prompting[1]

[1] Wang, Qunzhong, et al. "Does Graph Prompt Work? A Data Operation Perspective with Theoretical Analysis." ICML 2025.

Future Directions

Theoretical foundation

- Empirical improvements → theoretical guarantees

Theorem 1. *(Universal Capability of GPF) Given a pre-trained GNN model f, an input graph $\mathcal{G}: (\mathbf{A}, \mathbf{X})$, an arbitrary prompting function $\psi_t(\cdot)$, for any prompted graph $\hat{\mathcal{G}}: (\hat{\mathbf{A}} \in \mathbb{A}, \hat{\mathbf{X}} \in \mathbb{X})$ in the candidate space of the graph template $\mathcal{G}^* = \psi_t(\mathcal{G})$, there exists a GPF extra feature vector $\hat{p}$ that satisfies:*

$$f(\mathbf{A}, \mathbf{X} + \hat{p}) = f(\hat{\mathbf{A}}, \hat{\mathbf{X}}) \tag{11}$$

Universal capability of graph prompting[1]

[1] Fang, Taoran, et al. "Universal Prompt Tuning for Graph Neural Networks." NeurIPS. 2023.

Future Directions

Theoretical foundation

- Empirical improvements → theoretical guarantees

Theorem 1. *Given a random graph* $\mathcal{G} \sim CSBM(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, p, q)$ *and a pre-trained GCN model* f, *there always exist a set of* $M \geq 2$ *anchor prompts* $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_M\}$ *and the score vectors* $\boldsymbol{b}_{i,j}$ *for each edge* (v_i, v_j) *that improve the expected distance after GCN operation between classes* c_1 *and* c_2 *to* T *times without using edge prompts, where* $T \in (1, 1 + \frac{p}{|p-q|}]$.

Separability improvements by graph prompting[1]

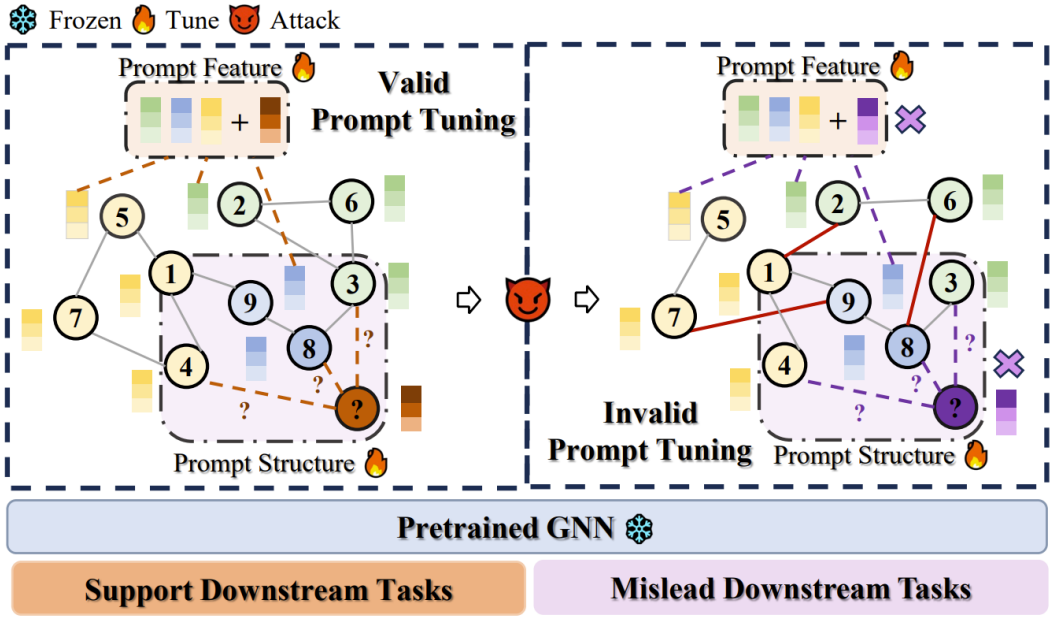
August 3–7, 2025
KDD2025

[1] Fu, Xingbo, Yinhan He, and Jundong Li. "Edge Prompt Tuning for Graph Neural Networks." ICLR 2025.

Future Directions

Attacks & defense in graph prompting

- A new attack and defense perspective by graph prompting



Adversarial attacks in graph prompting[1]

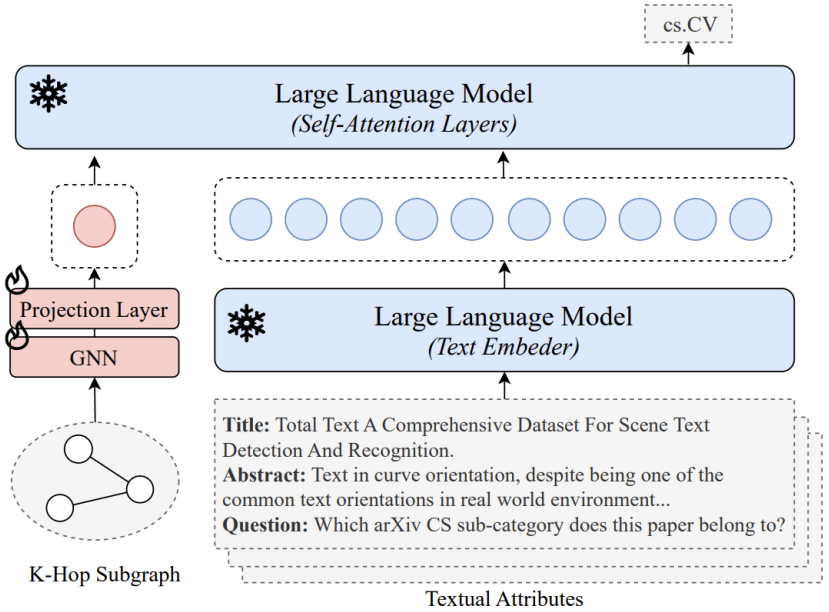
August 3–7, 2025
KDD2025

[1] Song, Shuhan, et al. "GPromptShield: Elevating Resilience in Graph Prompt Tuning Against Adversarial Attacks." ICLR. 2025.

Future Directions

LLM Incorporation for graph prompting

- Graph prompting with semantic information in graph data
- Text-attributed graphs (TAGs)



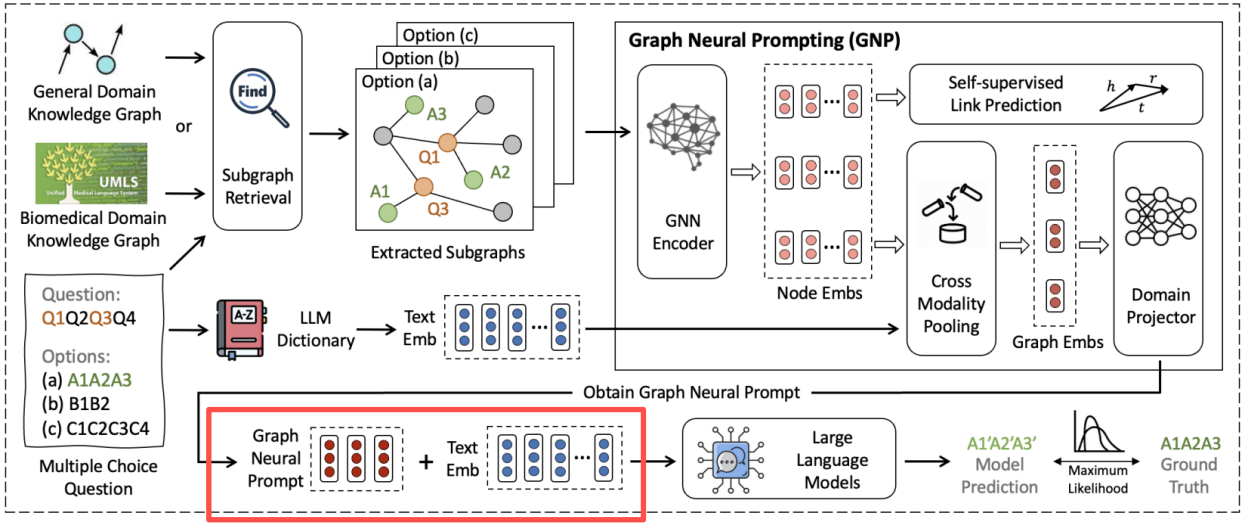
GraphPrompter[1]

[1] Liu, Zheyuan, et al. "Can we Soft Prompt LLMs for Graph Learning Tasks?." WWW 2024.

Future Directions

LLM Incorporation for graph prompting

- Graph prompting with semantic information in graph data
- Knowledge graphs (KGs)



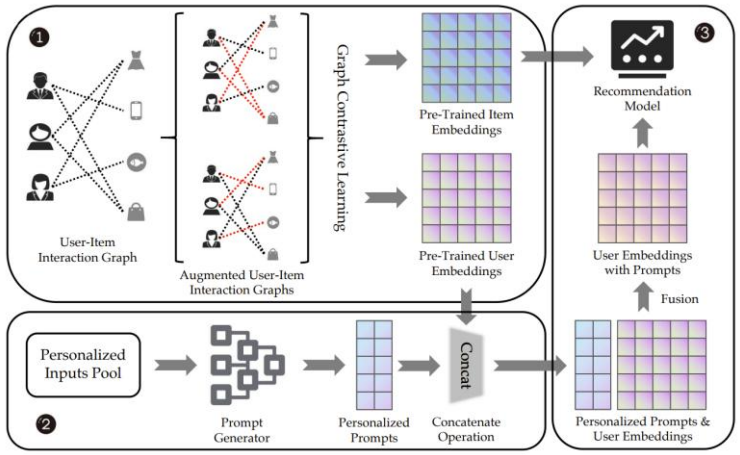
Graph Neural Prompting (GNP)[1]

August 3–7, 2025
KDD2025

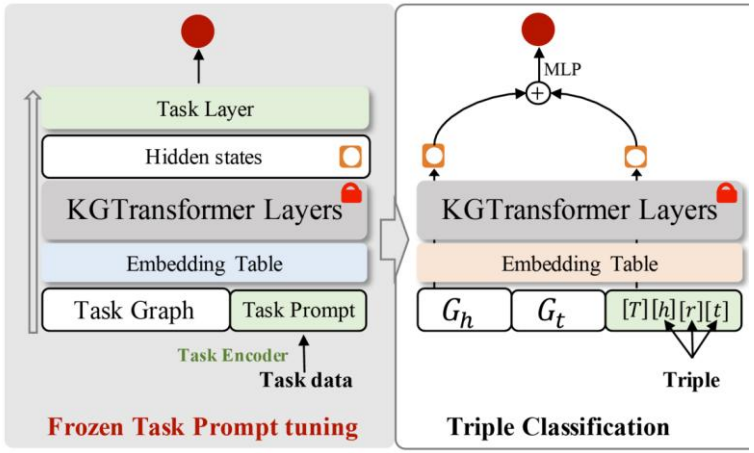
[1] Tian, Yijun, et al. "Graph Neural Prompting with Large Language Models." AAAI 2024.

Future Directions

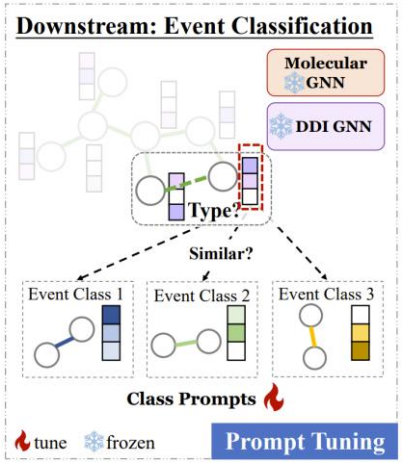
Applications in various domains



Recommendation systems[1]



Knowledge engineering[2]



Biology and medicine[3]

[1] Yang, Haoran, et al. "An Empirical Study Towards Prompt-Tuning for Graph Contrastive Pre-Training in Recommendations." NeurIPS 2023.

[2] Zhang, Wen, et al. "Structure Pretraining and Prompt Tuning for Knowledge Graph Transfer." WWW 2023.

[3] Wang, Yingying, et al. "DDIPrompt: Drug-Drug Interaction Event Prediction based on Graph Prompt Learning." CIKM. 2024.

THANK YOU!