Towards Graph Foundation Models

WWW 2024 Tutorial

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

Welcome to Big AI era!

- **Driving Forces:**
  - Technology advances
  - Availability of big data for training
  - Availability of powerful GPU

- **Performance improves with size.**
  - “The race to scale” begins…

- **The new thing (2021--)**
  - HUGE neural networks
  - VAST amounts of training data
  - MASSIVE compute power for training

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

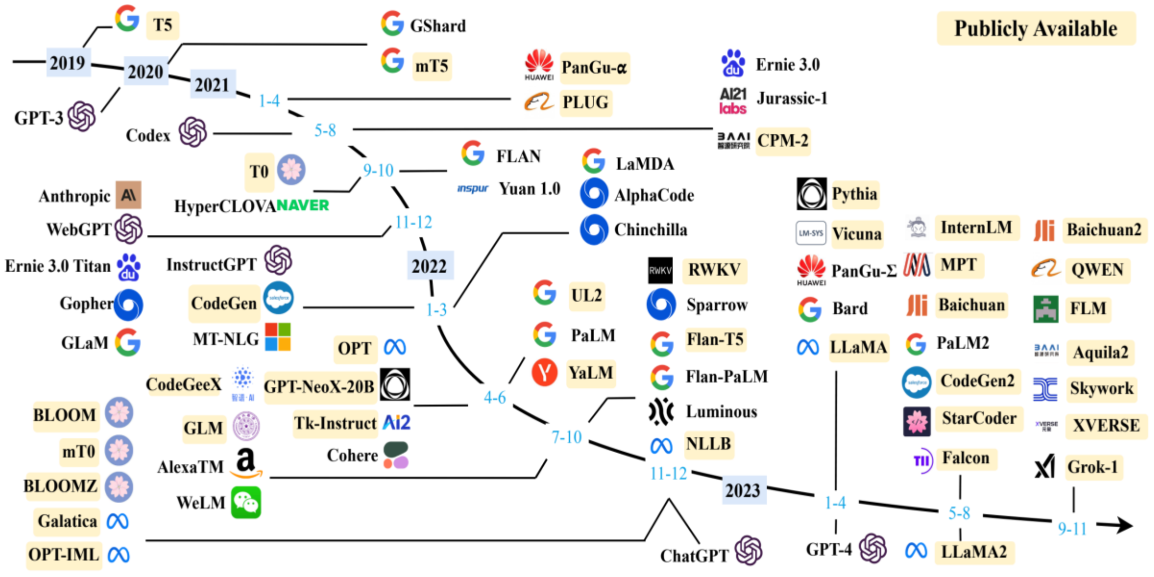
AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and

Foundation Models

A foundation model is a model that is trained on broad data and can be adapted to a wide range of downstream tasks.

➢ Big Idea

- Pretrain model, then fine-tune
- Revolutionize many research domains
  - Language
  - Vedio…



➢ Representative Examples

- Large Language Models (LLMs)
  - E.g., ELMo with millions of parameters to GPT-4 with trillions of parameters.
- Vedio Models: SORA

Graph Foundation Models

A graph foundation model (GFM) is a model pre-trained on extensive graph data, adapted for diverse downstream graph tasks.

➢ Motivation

- Existing LLMs struggle to model graph data
  - Euclidean data v.s. non-Euclidean data
- Existing LLMs struggle to handle graph tasks
  - node/edge/graph-level tasks

➢ Scope of this tutorial

- Concept of graph foundation model
- Recent progress
  - GNN-based methods
  - LLM-based methods
  - GNN+LLM-based methods
- Future directions

Towards Graph Foundation Models: A Survey and Beyond

JIAWEI LIU, CHENG YANG*, Beijing University of Posts and Telecommunications, China
ZHIYUAN LU, JUNZE CHEN, YIBO LI, Beijing University of Posts and Telecommunications, China
MENGMEI ZHANG, TING BAI, Beijing University of Posts and Telecommunications, China
YUAN FANG, Singapore Management University, Singapore
LICHAO SUN, Lehigh University, USA
PHILIP S. YU, University of Illinois Chicago, USA
CHUAN SHI†, Beijing University of Posts and Telecommunications, China

Foundation models have emerged as critical components in a variety of artificial intelligence applications, and showcase significant success in natural language processing and several other domains. Meanwhile, the field of graph machine learning is witnessing a paradigm transition from shallow methods to more sophisticated deep learning approaches. The capabilities of foundation models to generalize and adapt motivate graph machine learning researchers to discuss the potential of developing a new graph learning paradigm. This paradigm envisions models that are pre-trained on extensive graph data and can be adapted for various graph tasks. Despite this burgeoning interest, there is a noticeable lack of clear definitions and systematic analyses pertaining to this new domain. To this end, this article introduces the concept of Graph Foundation Models (GFMs), and offers an exhaustive explanation of their key characteristics and underlying technologies. We proceed to classify the existing work related to GFMs into three distinct categories, based on their dependence on graph neural networks and large language models. In addition to providing a thorough review of the current state of GFMs, this article also outlooks potential avenues for future research in this rapidly evolving domain.

Outline

Philip S. Yu University of Illinois Chicago

09:00-09:05 Introduction (5mins)

Chuan Shi Beijing University of Posts and Telecommunications

09:05-09:40 Overview (35mins)

Cheng Yang Beijing University of Posts and Telecommunications

09:40-10:30 GNN-based Methods (50mins)

10:30-11:00 Break (30mins)

Yuan Fang Singapore Management University

11:00-12:00 LLM/GNN+LLM-based Methods (50mins)

Host: Chuan Shi Beijing University of Posts and Telecommunications

12:00-12:30 Panel (30mins)

北京邮电大学
Beijing University of Posts and Telecommunications

Towards Graph Foundation Models
Part I: Overview

Prof. Chuan Shi

shichuan@bupt.edu.cn

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Outline

√ Graph Foundation Models

- Progress in Related Work

- Challenges and Future Direction

Foundation Models

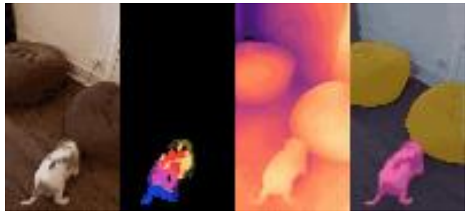
A foundation model is any model that is trained on broad data and can be adapted to a wide range of downstream tasks.[1]

Language

OpenAI × GPT4

Language foundation models show initial signs of universal AI capabilities.

Vision



Meta × DINOv2

Vision foundation models exhibit strong image understanding capabilities.

Speech

Google × USM

Speech foundation models show the capability to recognize hundreds of languages.

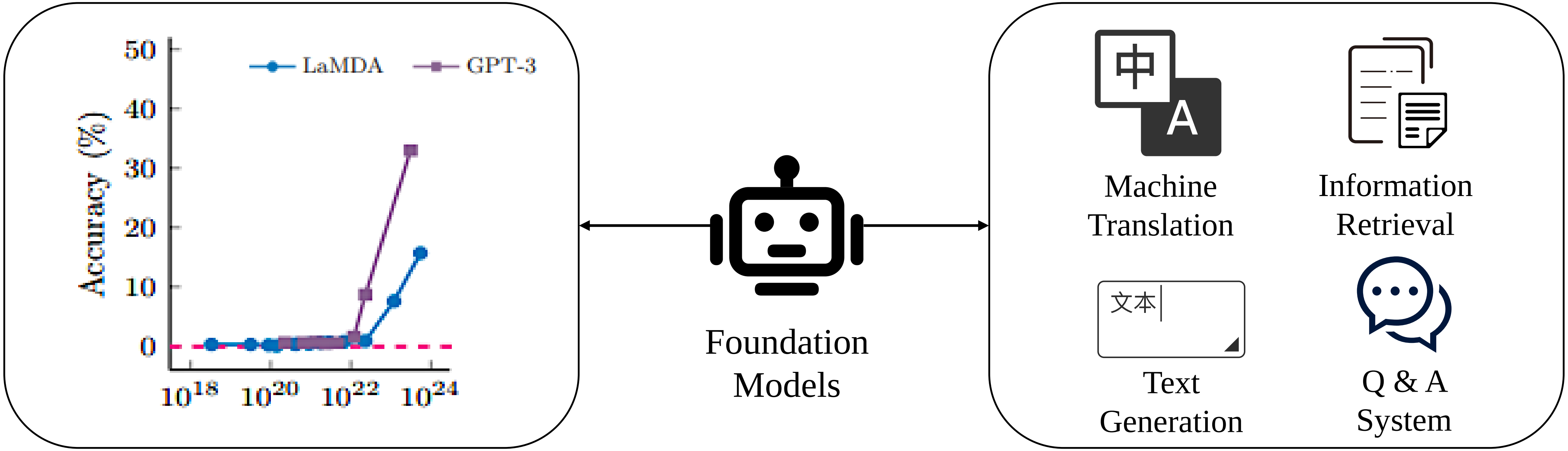
Foundation models have become a reality in domains like language, vision, and speech.

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brun-skill, et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

Characteristics of Foundation Models

Two Characteristics of Foundation Models:

- Emergence: As a foundation model scales up, it spontaneously manifests novel capabilities.
- Homogenization: The model’s versatility enables its deployment across diverse applications.



Emergence

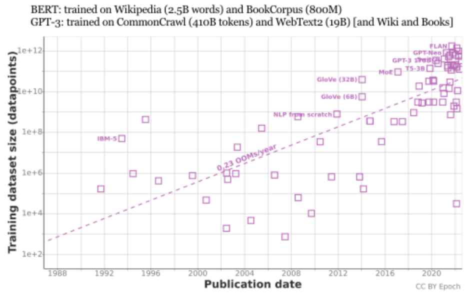
Homogenization

Wei J, Tay Y, Bommasani R, et al. Emergent abilities of large language models[J]. arXiv preprint arXiv:2206.07682, 2022.

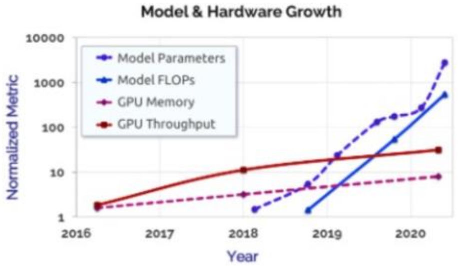
Factors Driving Foundation Model Success

Data

- The increasing number of data-collecting devices results in a massive growth in data volume.



Data Growth



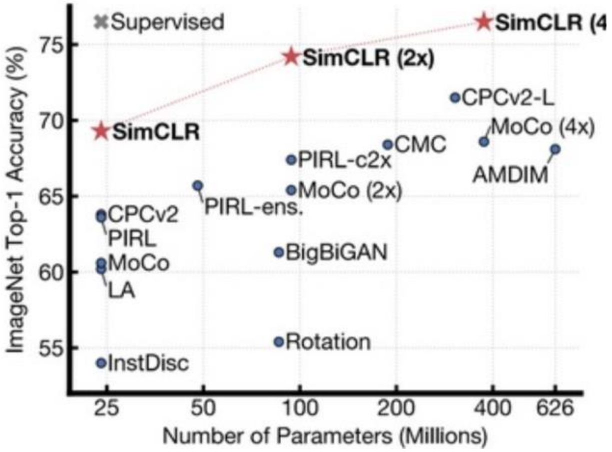
GPU Development

Hardware

- the rapid advancement of GPU hardware

Self-supervised Learning (SSL)

- exploiting raw unlabeled data



SSL

Transformer Architectures

- attention mechanism

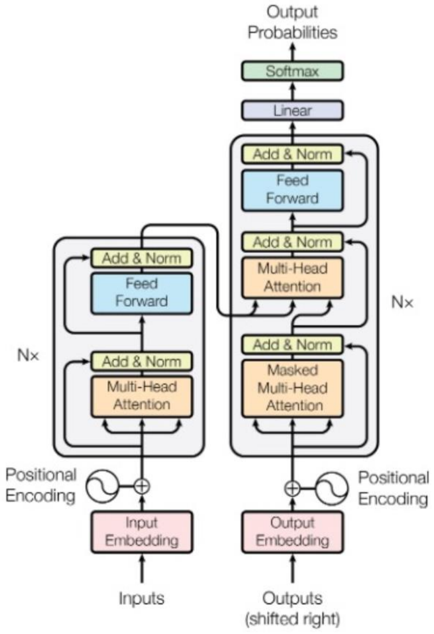


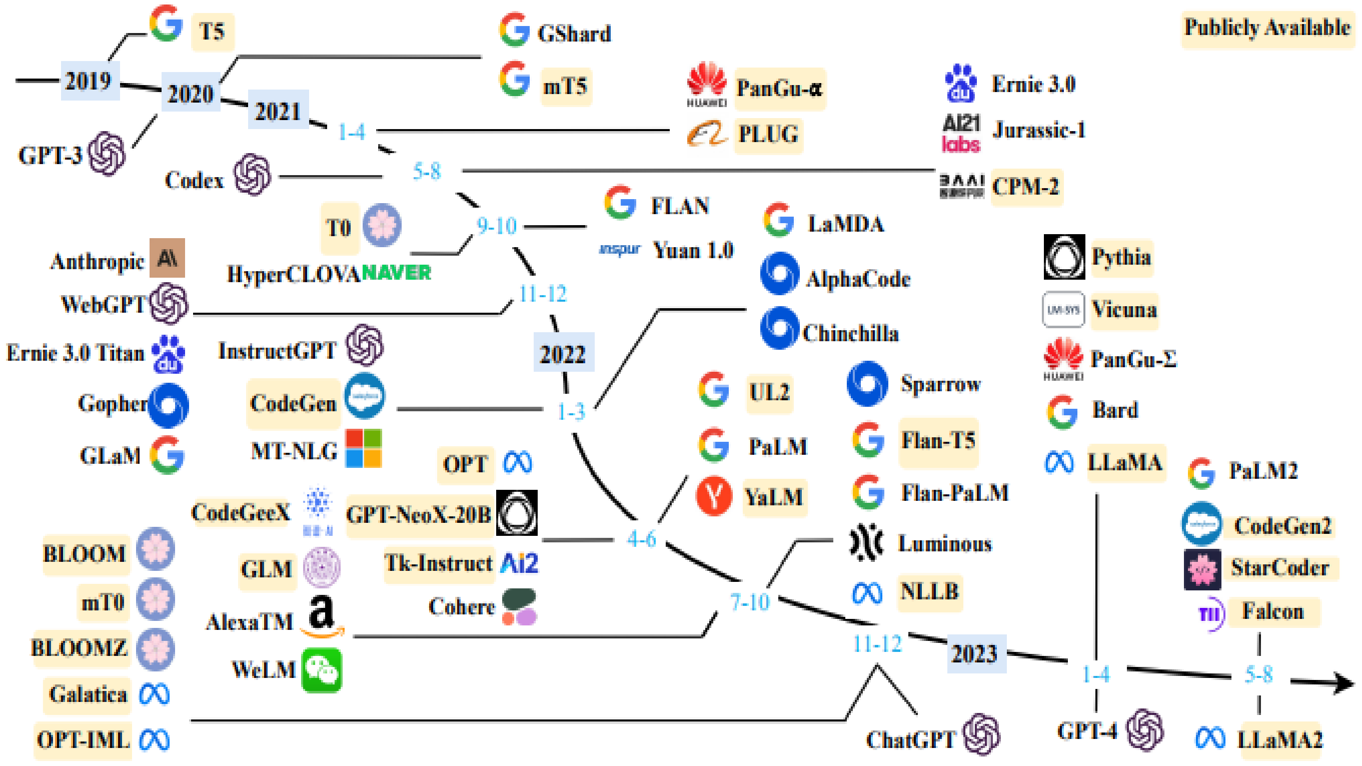
Figure 1: The Transformer - model architecture.

Transformer

Language Foundation Models

Large Language Models (LLMs) refer to pre-trained language models with massive parameters and are typical representatives of foundation models.

- LLMs have progressed from models like ELMo with millions of parameters to GPT-4 with trillions of parameters.
- LLMs showcase key AI abilities like comprehension, generation, logic, and memory, hinting at the path towards artificial general intelligence (AGI).

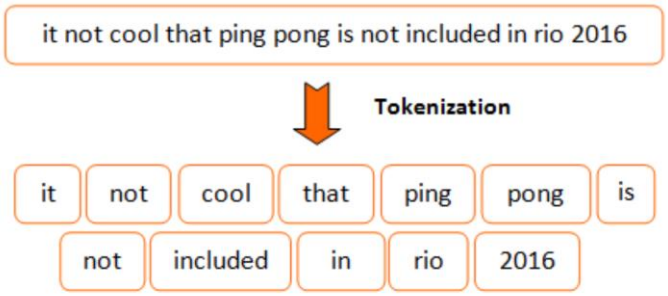


Zhao W X, Zhou K, Li J, et al. A survey of large language models[J]. arXiv preprint arXiv:2303.18223, 2023.

Large Language Models

Data

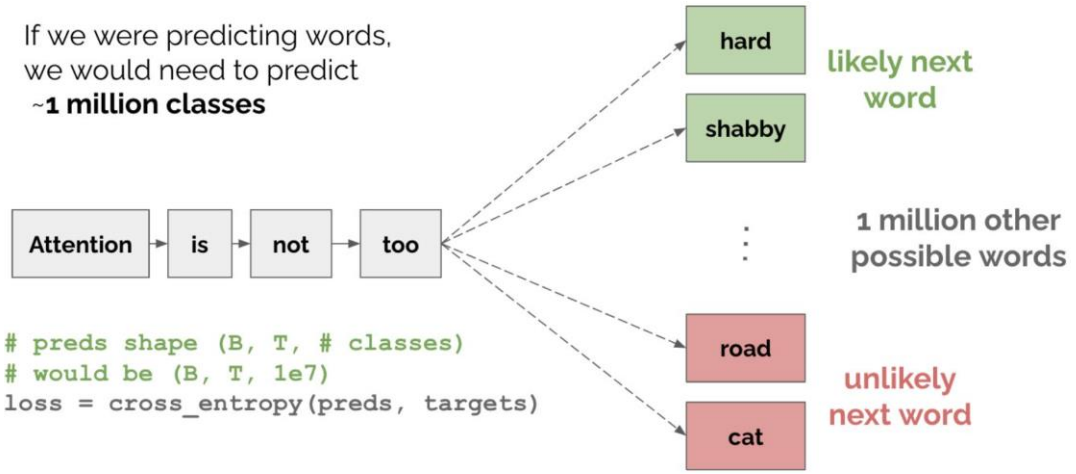
- ➢ Language data: text or spoken content in a human language
  - sequential data
  - Euclidean data



Language Data

Backbone Architectures

- ➢ Mostly based on Transformer
  - e.g., BERT[1], GPT-3[2]
- ➢ Pre-trained with pretext tasks:
  - next word prediction (NWP)
  - masked language modeling (MLM)…



Next Word Prediction (NWP)

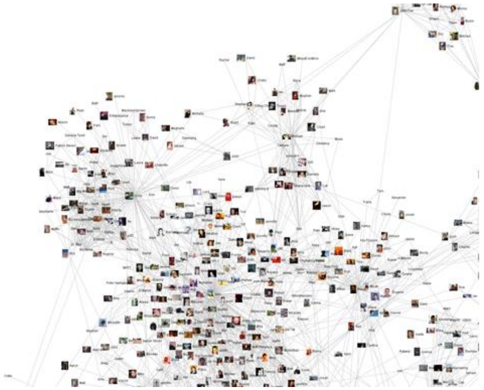
Downstream Tasks

- ➢ Hundreds of downstream tasks
  - e.g., machine translation, sentiment analysis…

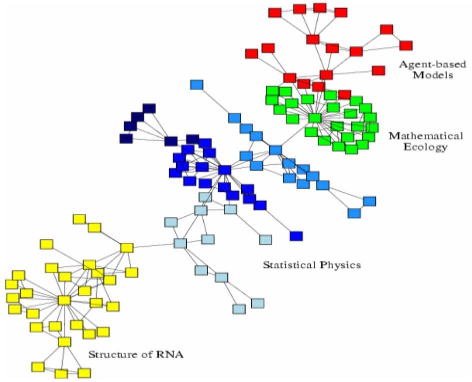
[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
[2] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[C]. NeurIPS 2020, 33: 1877-1901.

Graphs

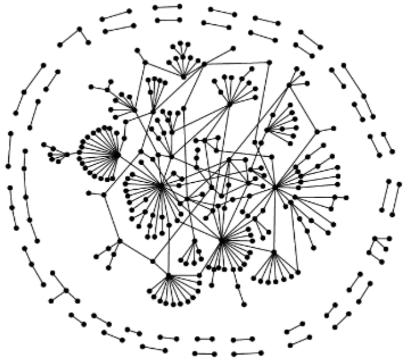
Graphs are a general language for describing and modeling complex systems.



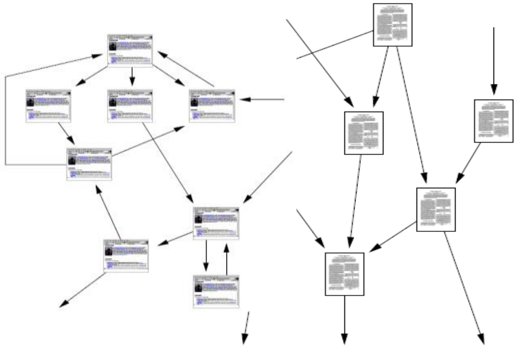
Social networks



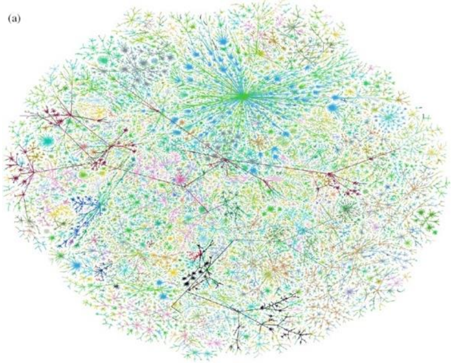
Economic networks



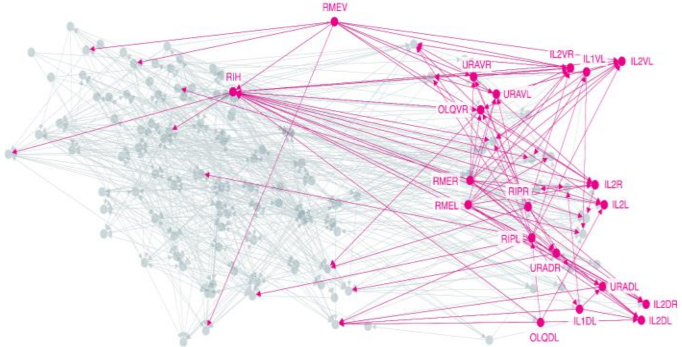
Biomedical networks



Information networks



Internet

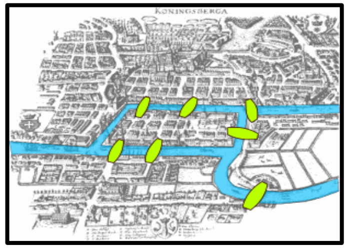


Networks of neurons

Graph Machine Learning

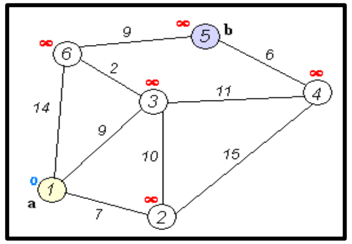
- Graph G is an ordered pair (V, E), where V is the node set and E is the edge set.
- Graph machine learning refers to the application of machine learning to graph data, commonly known as graph learning or graph models.

Seven Bridges of Königsberg



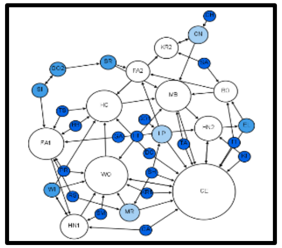
Graph theory
- Euler

Shortest Path Problem



Graph algorithms
- Dijkstra

Long Tail Distribution

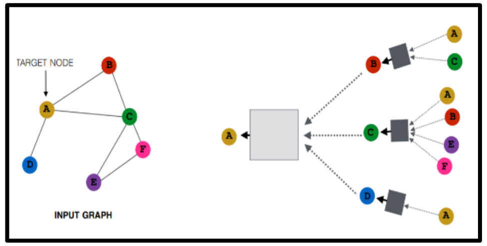


Network Science
- Barabasi

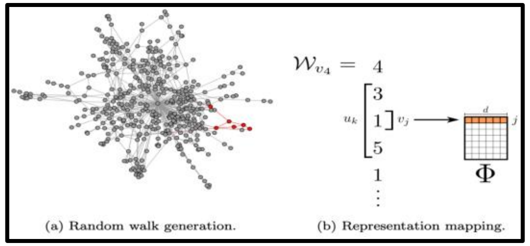
1736 1956 2002

2017 2014 2013

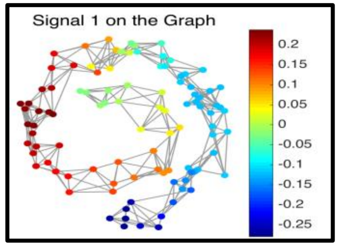
Graph neural networks
- GCN



Graph embedding
- DeepWalk

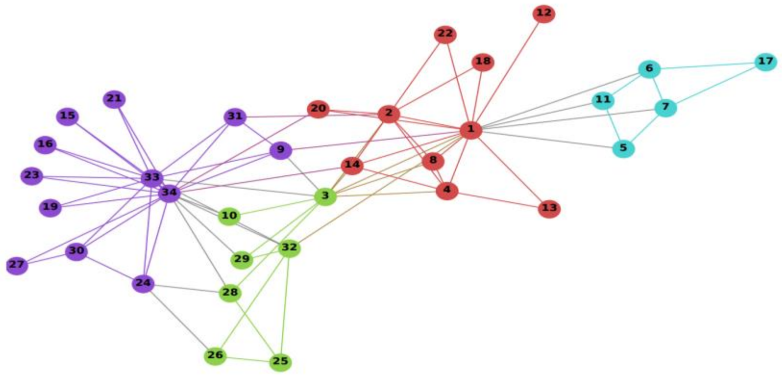


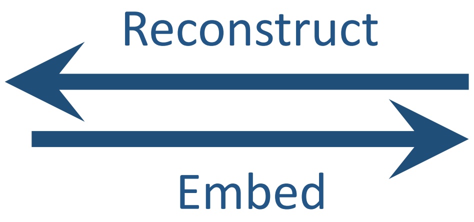
Graph signal processing
- Shuman

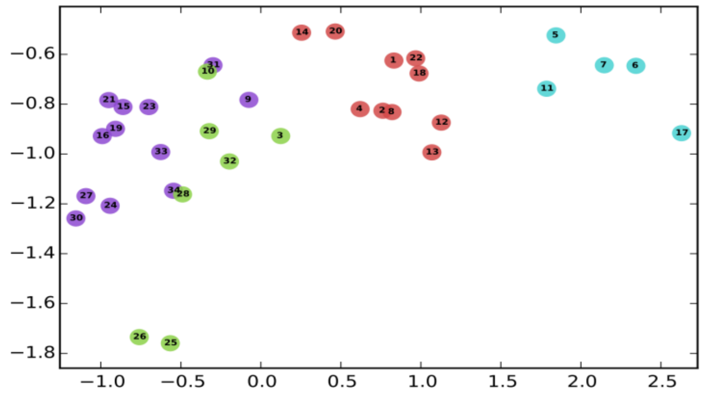


Graph Representation Learning

Graph Representation Learning (GRL): embed each node of a graph into a low-dimensional vector space



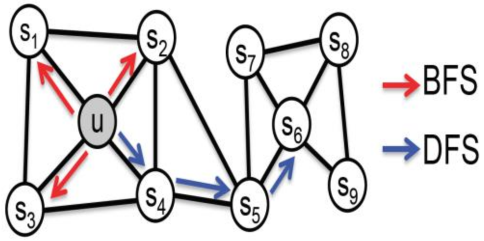




Shallow model

➢ Random walk based

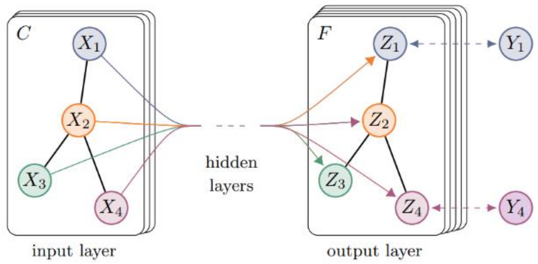
- e.g., DeepWalk, node2vec



Deep model

➢ GNN based

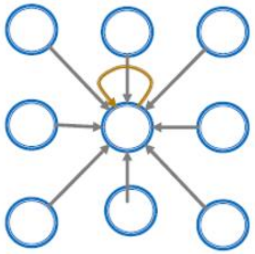
- e.g., GCN, GraphSage, GAT



Data in GNN

Data

- **Graph data**
  - non-Euclidean data
- **Various domains**
  - social networks
  - molecules
  - E-commerce…
- **Various types**
  - homogenous graph
  - heterogenous graph
  - hypergraph…


Graph

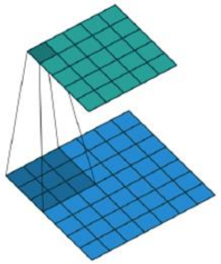
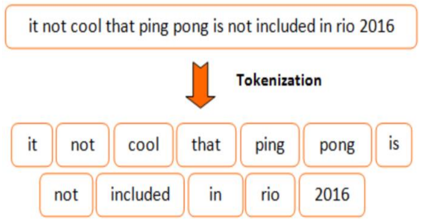
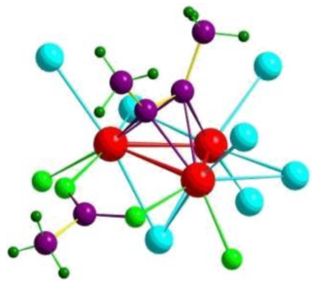


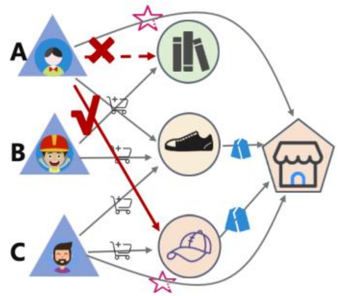
Image (Grid)



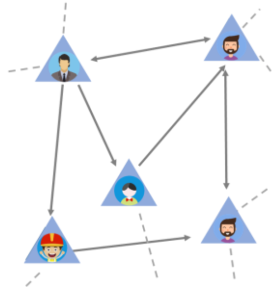
Language (Seq.)

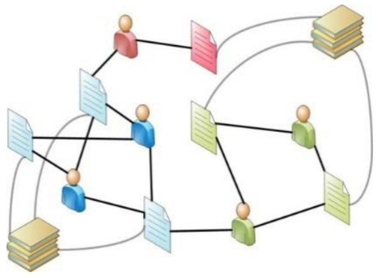

Social Networks

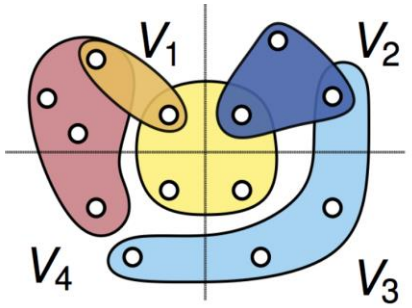

Molecules



E-commerce


Homogeneous


Heterogeneous

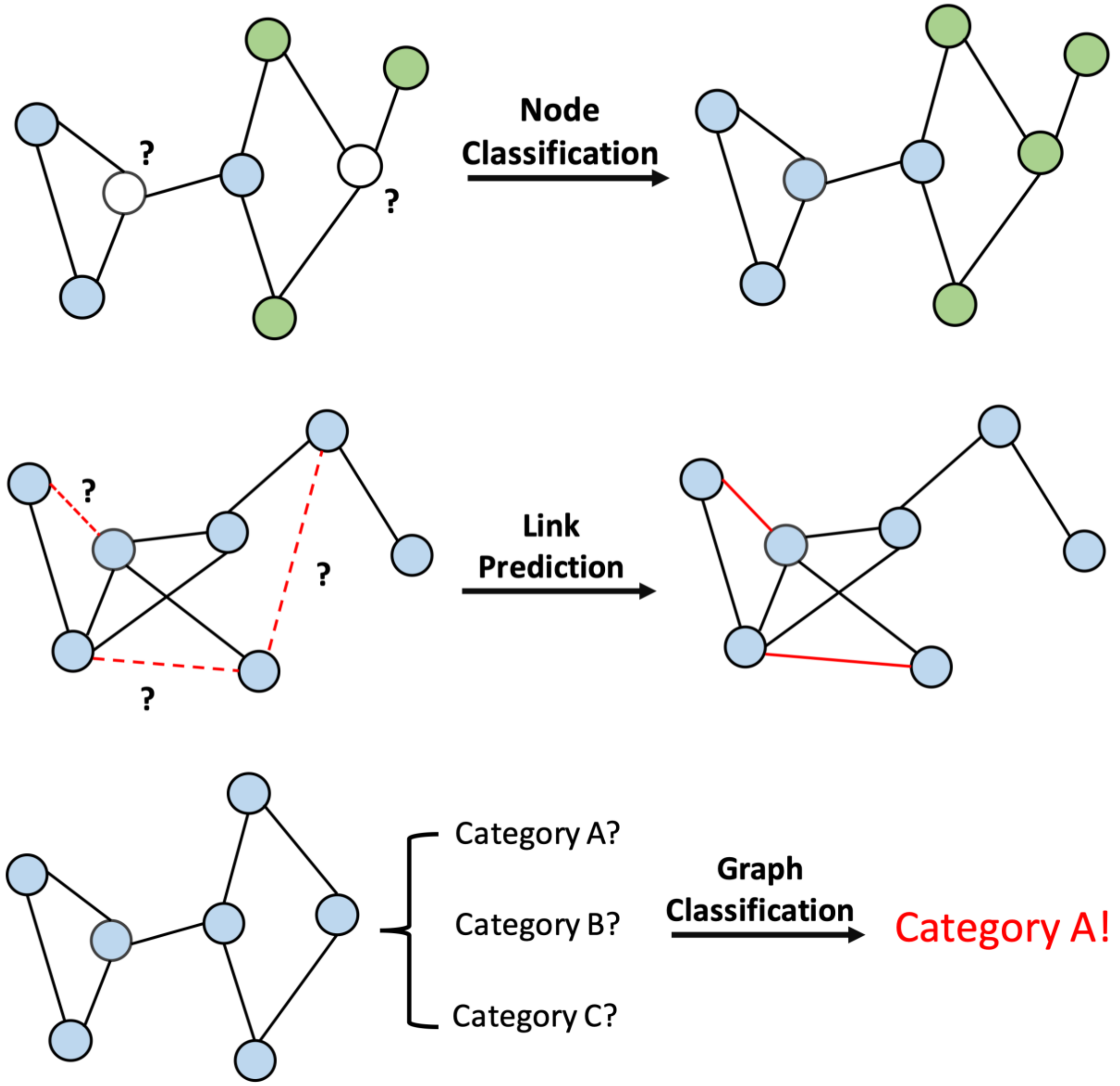


Hypergraph

Tasks in GNN

Downstream Tasks

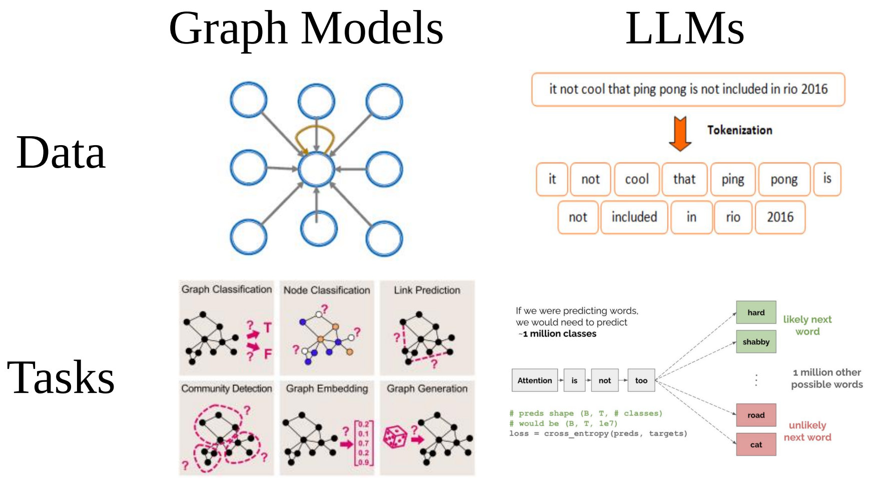
- **Node-level tasks**
  - node classification
  - node regression
  - node clustering…
- **Edge-level tasks**
  - link prediction
  - shortest path prediction
  - maximum flow prediction…
- **Graph-level tasks**
  - graph classification
  - graph generation
  - graph condensation…



Graph Models Meet Large Language Models

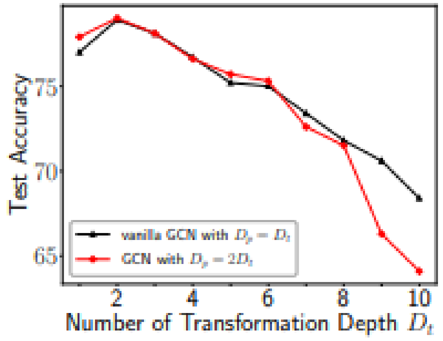
LLMs cannot solve graph-related problems.

- LLMs struggle to model graph structure semantics.
- LLMs struggle to handle diverse graph tasks.

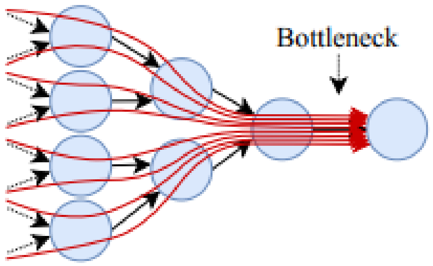


Graph models do not possess the capabilities of LLMs.

- Limited expressive power
- Deep GNNs: over-smoothing/over-squassion issues
- Lack emergence capability
- Cannot support multiple tasks



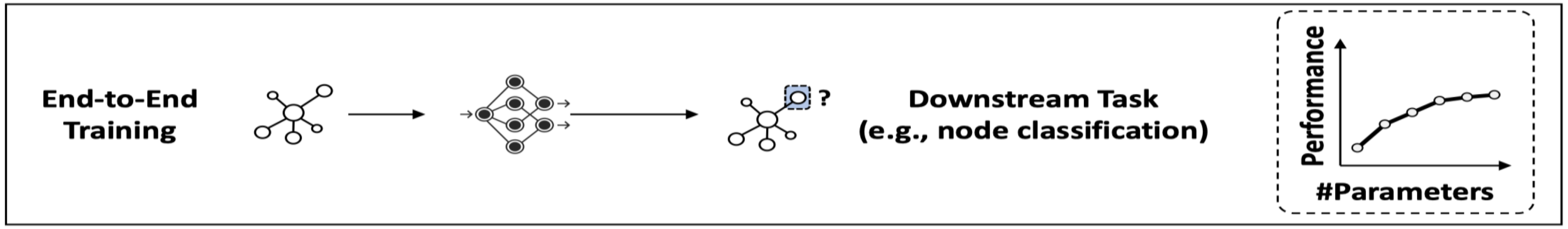
Performance Decline of Deep GNNs



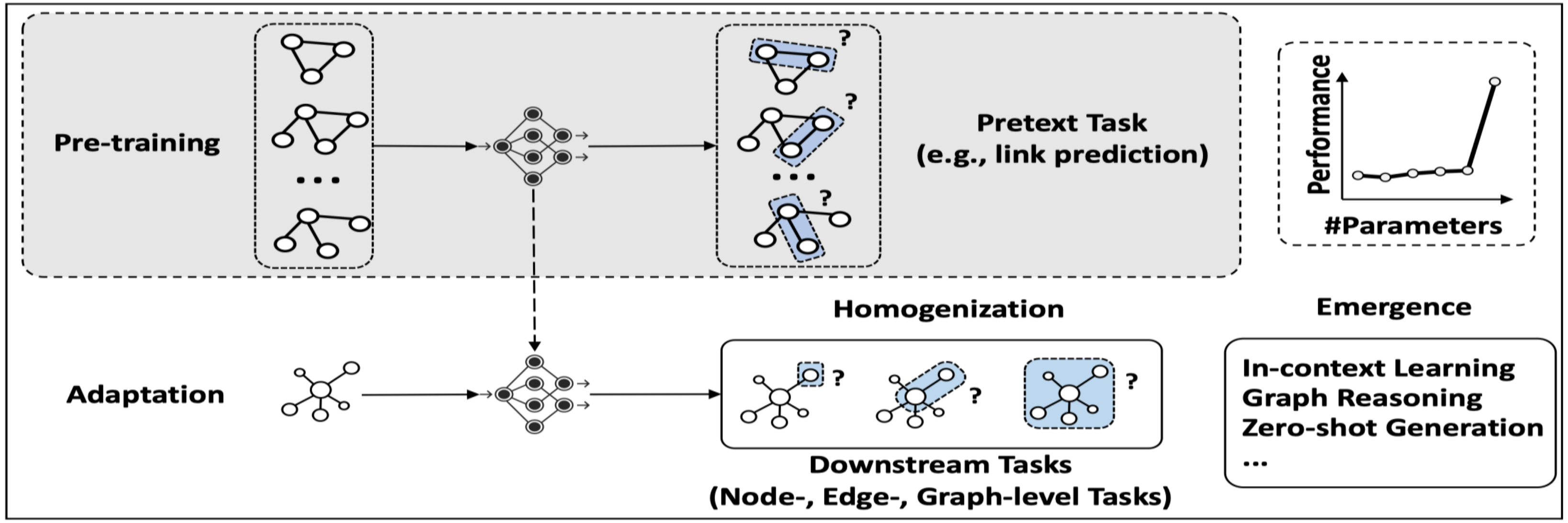
Information Bottleneck in GNNs

Graph Foundation Models

A graph foundation model (GFM) is a model pre-trained on extensive graph data, adapted for diverse downstream graph tasks.



(a) Deep Graph Learning.



(b) Graph Foundation Models.

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.

Characteristics of Graph Foundation Models

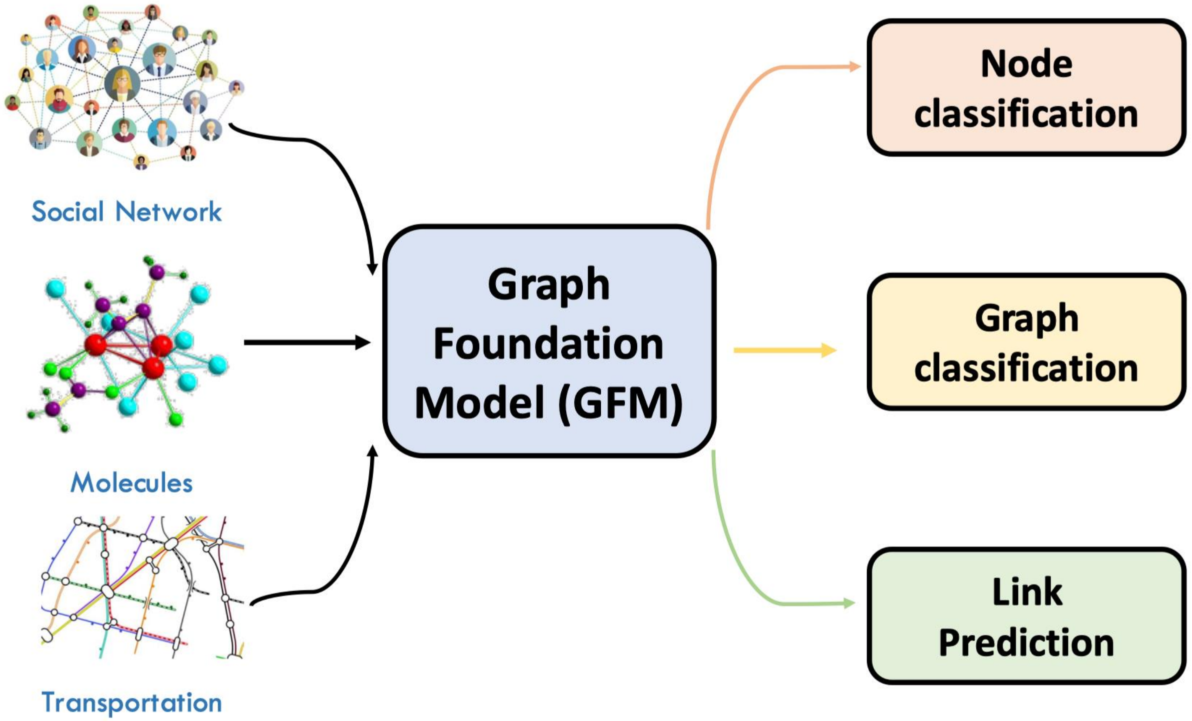
Two Characteristics

Emergence

- ➢ Novel capbility when larger model or more graph data
  - graph reasoning
  - graph generation…

Homogenization

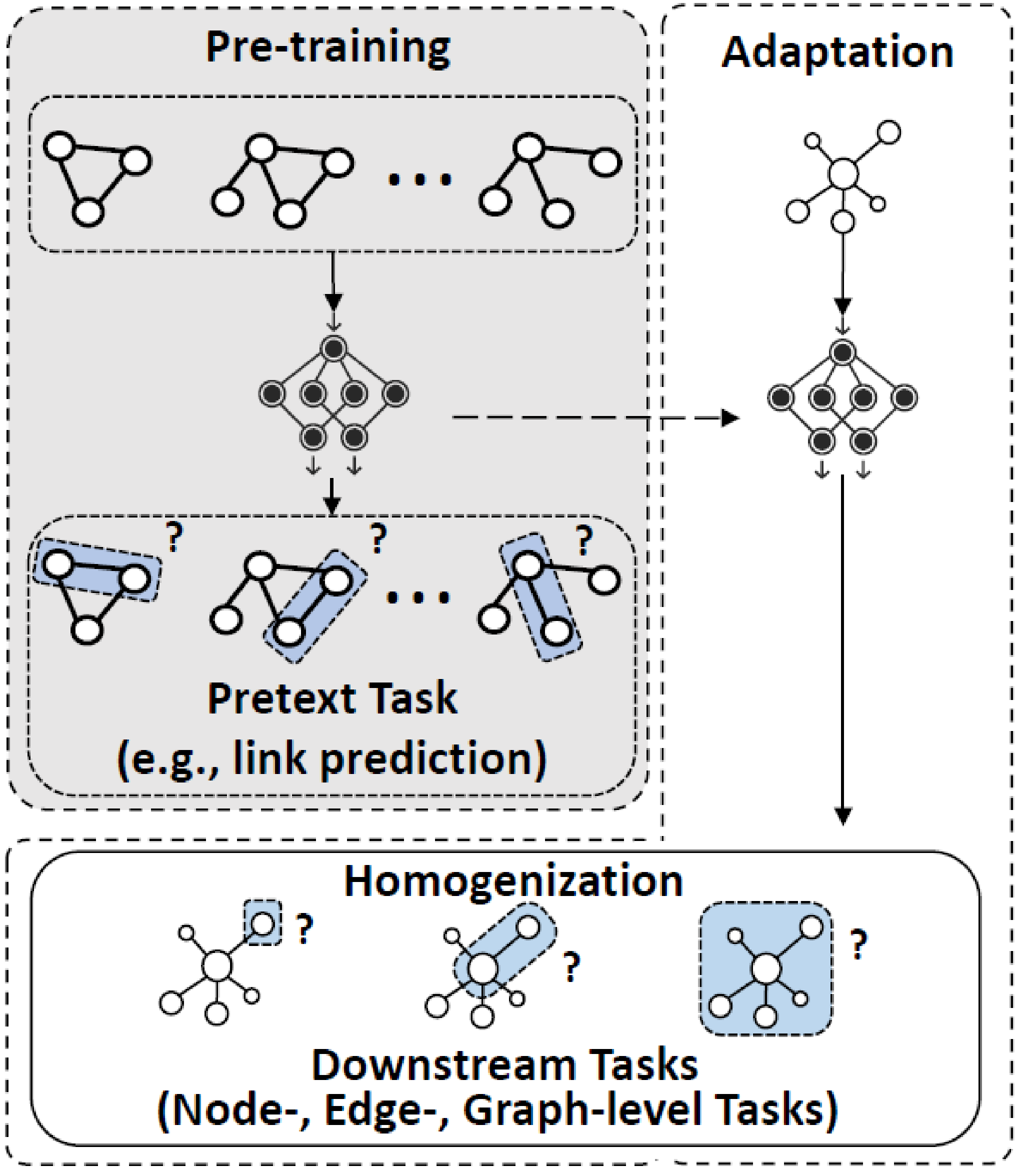
- ➢ Apply to different formats of tasks
  - node/edge/graph tasks



Key Techniques of Graph Foundation Models

Key Techniques of GFMs

- **Pre-training**: neural networks are trained on a large graph dataset in a self-supervised manner
  - contrastive pre-training: contrastive positive samples against negative samples
  - generative pre-training: reconstruct or predict original feature
- **Adaptation**: adapt pre-trained models to specific downstream tasks or domains to enhance their performance
  - fine-tuning
  - prompt-tuning



GFMs v.s. LLMs

Similarities: common goal and similar learning paradigm

Differences: (1) different data and tasks; (2) technological differences

		Language Foundation Model	Graph Foundation Model
Similarities	Goal	Enhancing the model's expressive power and its generalization across various tasks	
	Paradigm	Pre-training and Adaptation	
Intrinsic differences	Data	Euclidean data (text)	Non-Euclidean data (graphs) or a mixture of Euclidean (e.g., graph attributes) and non-Euclidean data
	Task	Many tasks, similar formats	Limited number of tasks, diverse formats
Extrinsic differences	Backbone Architectures	Mostly based on Transformer	No unified architecture
	Homogenization	Easy to homogenize	Difficult to homogenize
	Domain Generalization	Strong generalization capability	Weak generalization across datasets
	Emergence	Has demonstrated emergent abilities	No/unclear emergent abilities as of the time of writing

Outline

- Graph Foundation Models

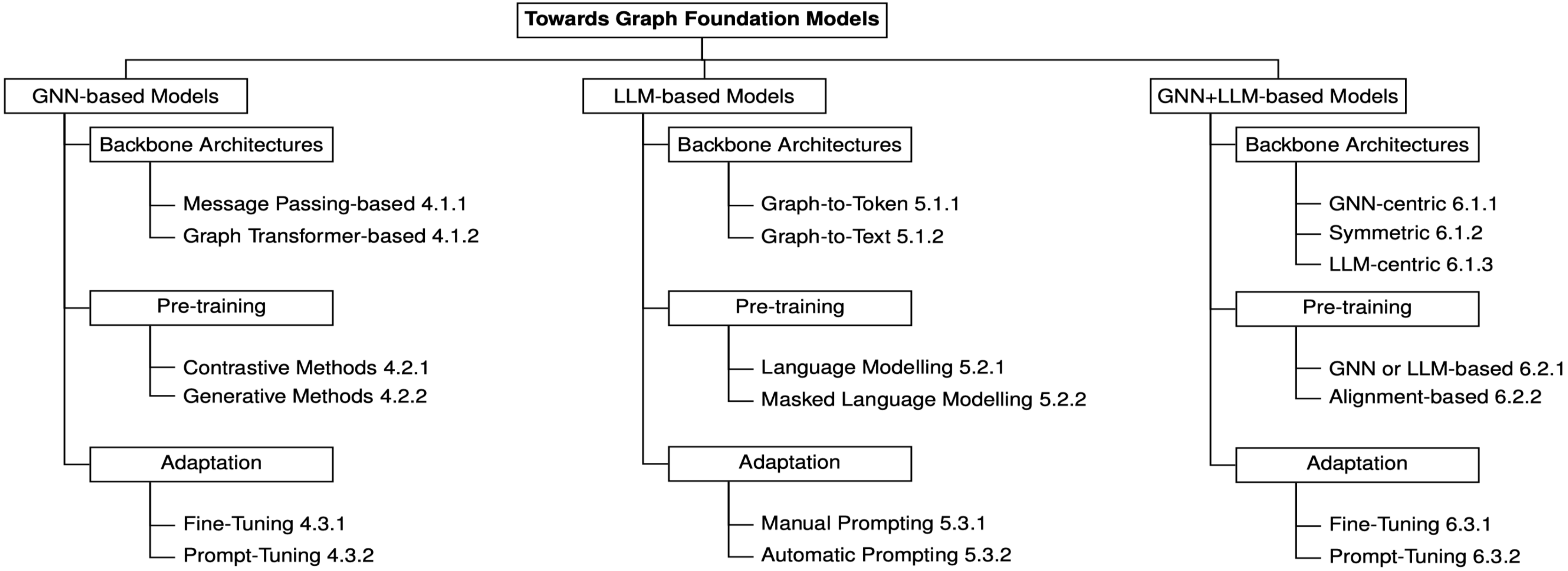
√ Progress in Related Work

- Challenges and Future Direction

Taxonomy of Related Work

No GFMs until now, but a lot of explorations is on the way.

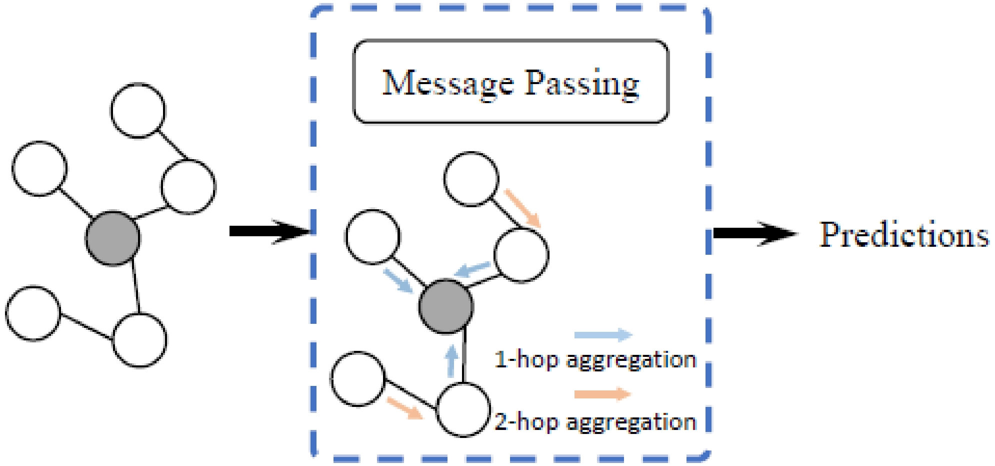
Categorize existing explorations into three distinct groups according to the dependence on GNNs and LLMs



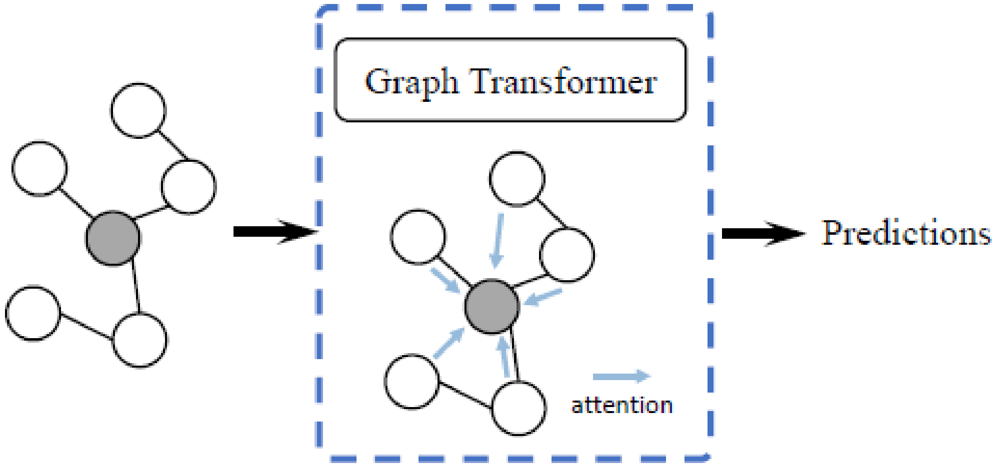
GNN-based Models

Seeking to enhance current graph learning through innovative approaches in GNN model architectures, pre-training, and adaptation.

- Architectures: Graph Transformer, e.g., Specformer (ICLR23), CoBFormer (ICML24)
- Pre-training: Graph Pretraining, e.g., PT-HGNN (KDD21), GraphPAR (WWW24)
- Adaptation: Graph Prompt, e.g., All In One (KDD23), MultiGPrompt (WWW24)



(a) Message Passing.

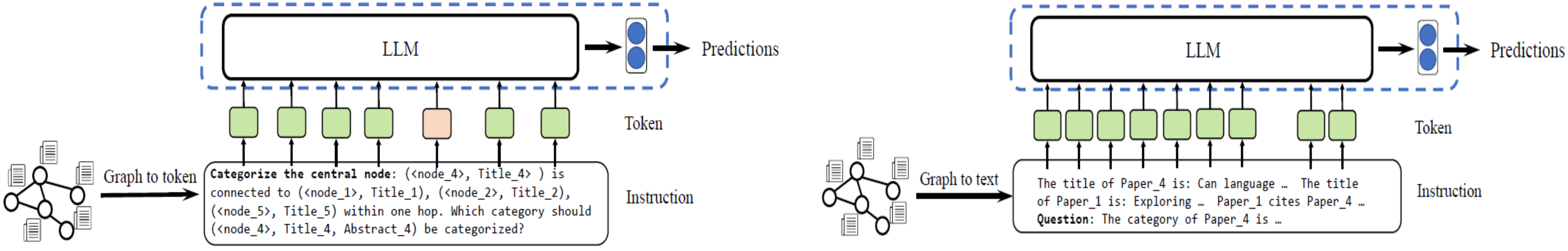


(b) Graph Transformer.

LLM-based Models

Exploring the feasibility of transforming graphs into text or tokens to leverage LLMs as foundation models.

- Graph-to-Token: transform graphs into tokens and then input them into LLMs
  - e.g., InstructGLM
- Graph-to-Text: transform graphs into texts and then input them into LLMs
  - e.g., NLGraph (NIPS24), LLM4Mol



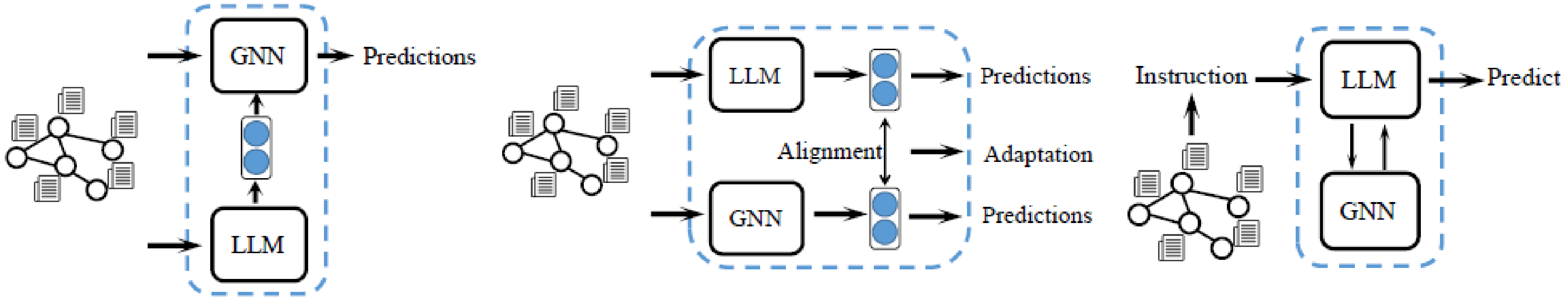
(a) Graph-to-token.

(b) Graph-to-text.

GNN+LLM-based Models

Exploring synergies between GNNs and LLMs to enhance graph learning.

- GNN-centric Models: utilize LLM to extract node feature and make predictions using GNN
  - e.g., SimTeG, TAPE
- Symmetric Models: align the embeddings of GNN and LLM
  - e.g., GraphTranslator (WWW24), G2P2 (SIGIR23), ConGrat
- LLM-centric Models: utilize GNNs to enhance the performance of LLM
  - e.g., Graph-Toolformer



Outline

- Graph Foundation Models
- Progress in Related Work

√ Challenges and Future Direction

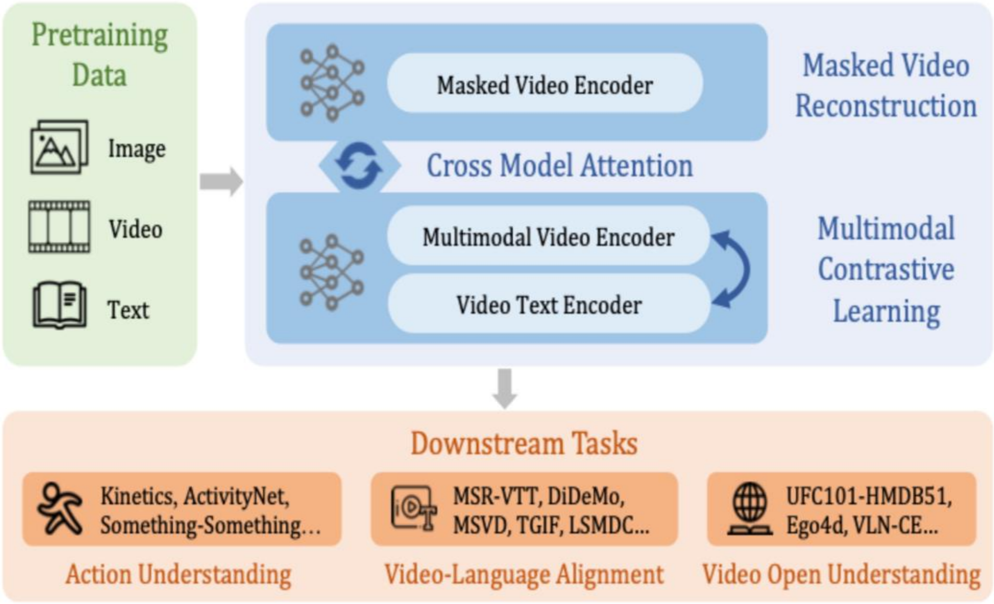
Challenges in Model

Model Architectures

- ➢ It remains unknown whether current architectures are optimal choices.
- ➢ Multimodal foundation models
  - Using graph to extend the multiple modalities…

Model Training

- ➢ Is there uniform pretext tasks for graph
- ➢ Some ideas from other directions
  - knowledge distillation
  - reinforcement learning from human feedback
  - model editing…

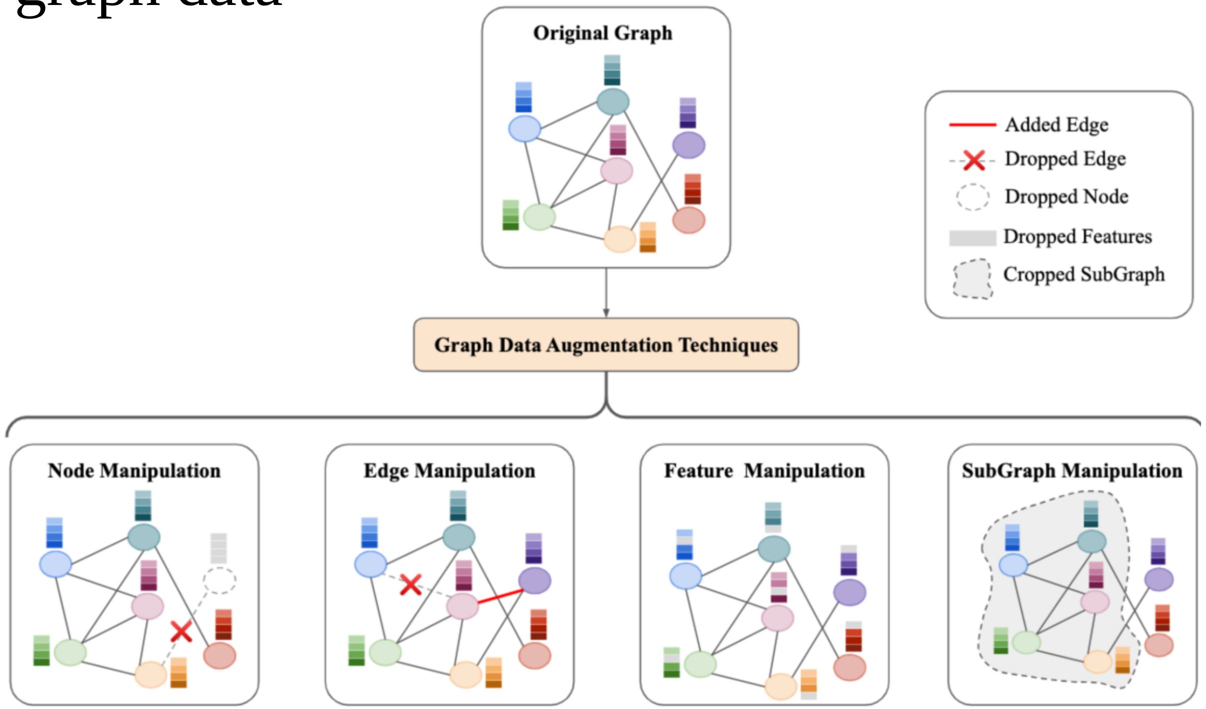


Multimodal Foundation Models

Challenges in Data and Evluation

Data Quantity and Quality

- ➢ Limited amount of open-source large-scale graph data
  - concentrated in a single domain
- ➢ Using augmentation strategies
  - graph structure learning
  - feature completion
  - label mixing…



Graph Augmentation

Evaluation

- ➢ Lacking labels in open-ended tasks
  - human evaluation
  - meta-evaluation
- ➢ Evaluating robustness, trustworthiness, holistic performance…

Challenges in Applications

Killer Applications

- It is not yet clear that graph foundation models can similarly catalyze groundbreaking applications in graph tasks.
- Promising fields
  - urban computing
  - drug development…

Safety

- Black-box nature introduces safety concerns.
  - hallucination
  - privacy leaks
- Promising technologies
  - counterfactual reasoning…

图数据挖掘和机器学习

扫码关注我们

www.shichuan.org

Thanks

Q&A

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

Towards Graph Foundation Models
Part II: GNN-based Methods

Cheng Yang

yangcheng@bupt.edu.cn

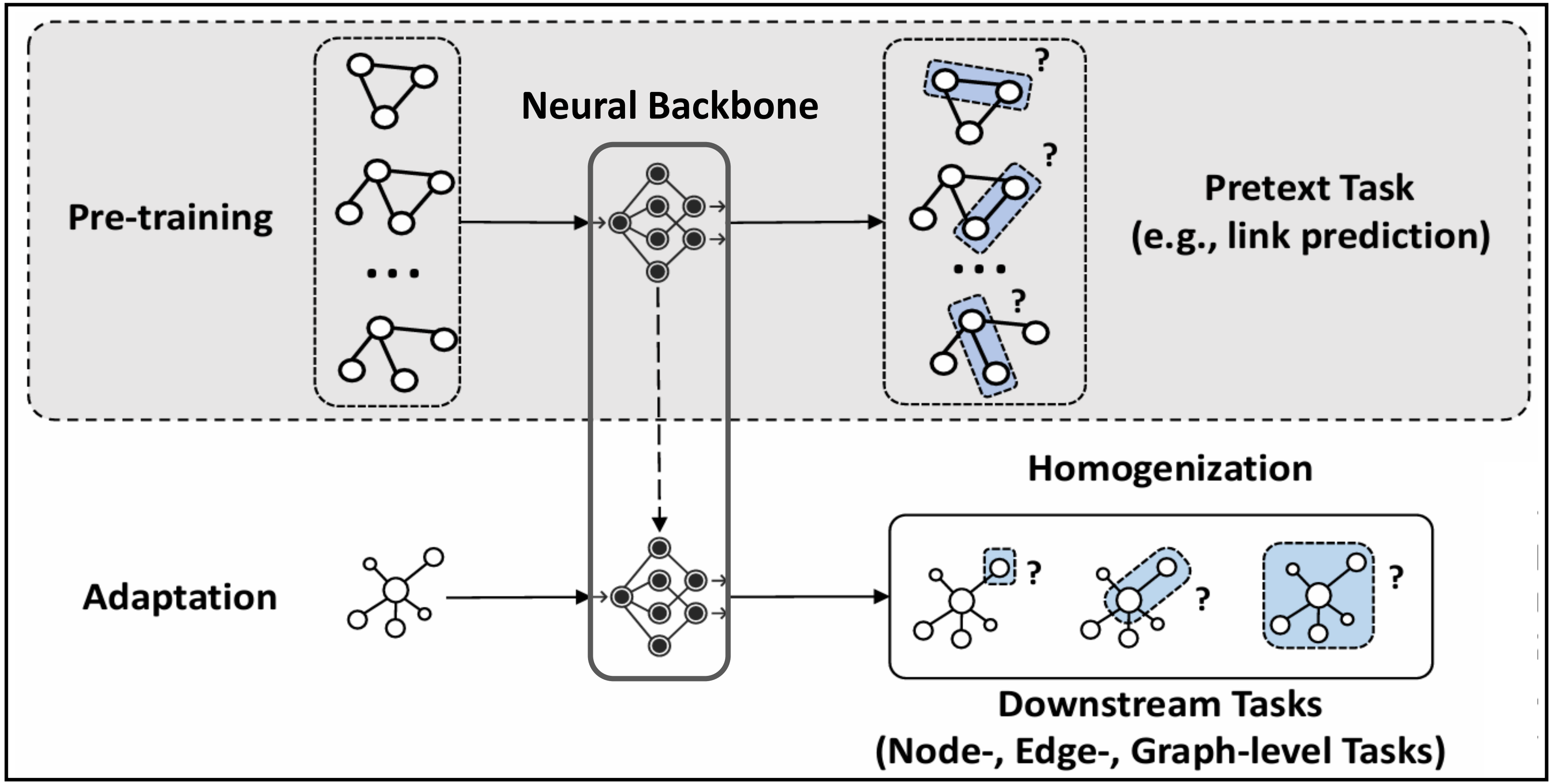
Beijing University of Posts and Telecommunications

GNN-based Methods

Backbone： No unified architecture

（Message Passing/Graph Transformer）

Paradigm： Pre-training + Adaptation



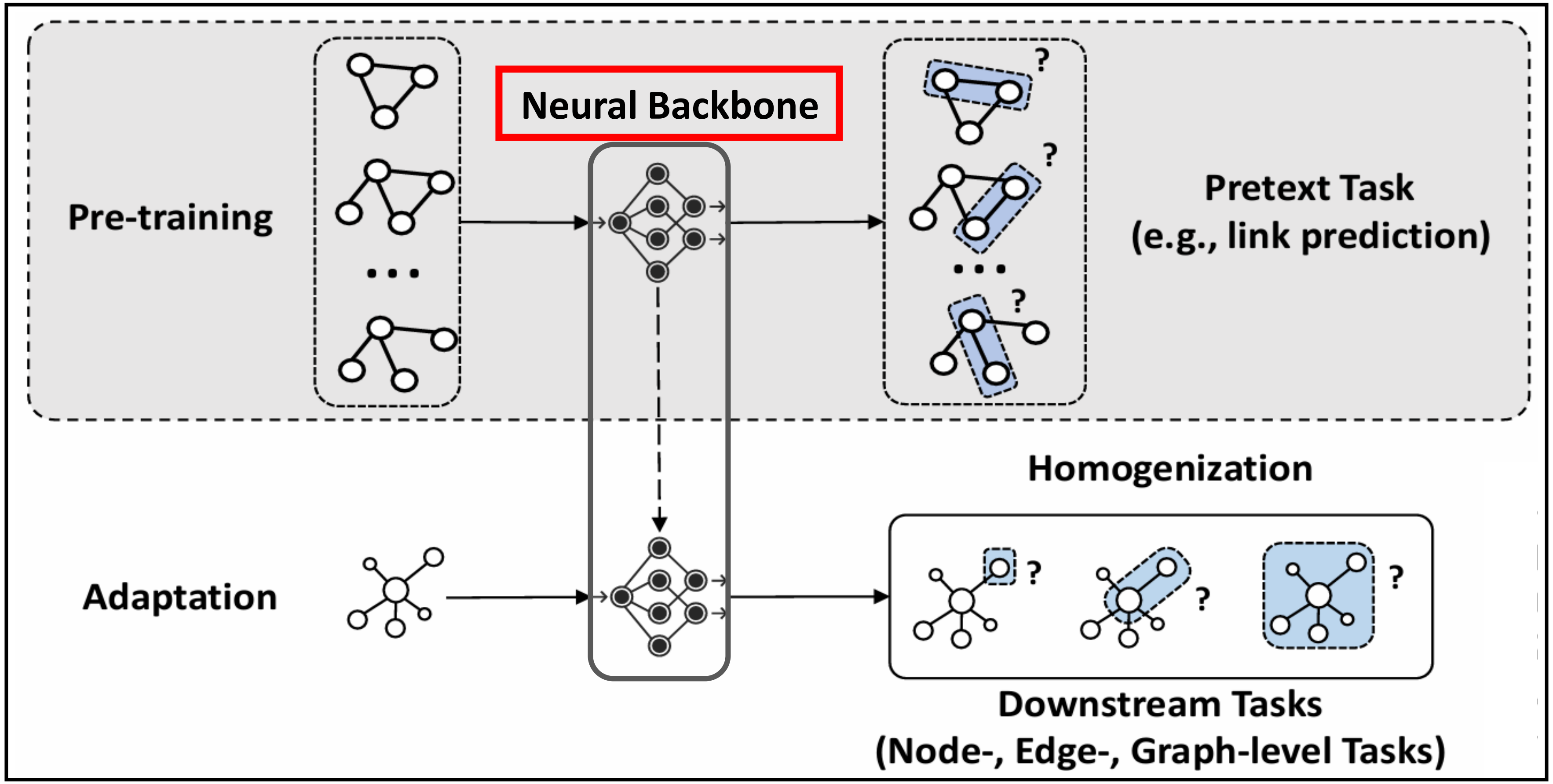
Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

GNN-based Methods

Backbone： No unified architecture

（Message Passing/Graph Transformer）

Paradigm： Pre-training + Adaptation

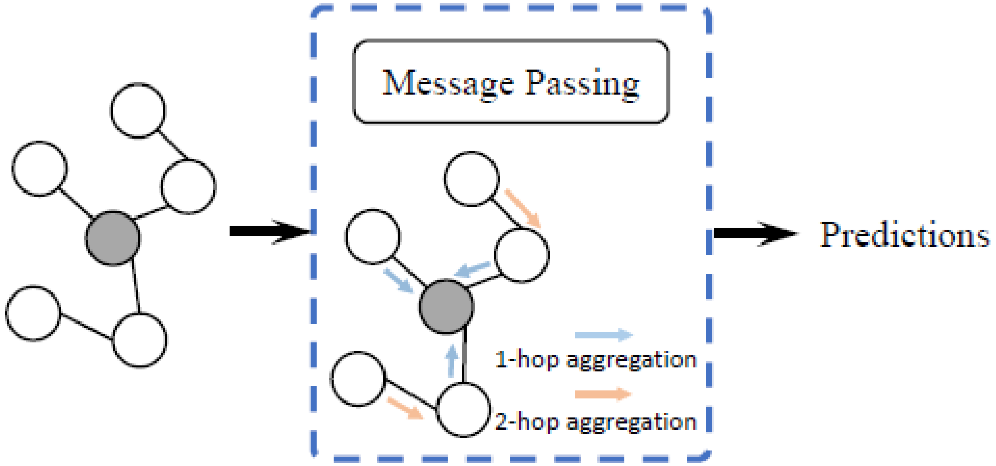


Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

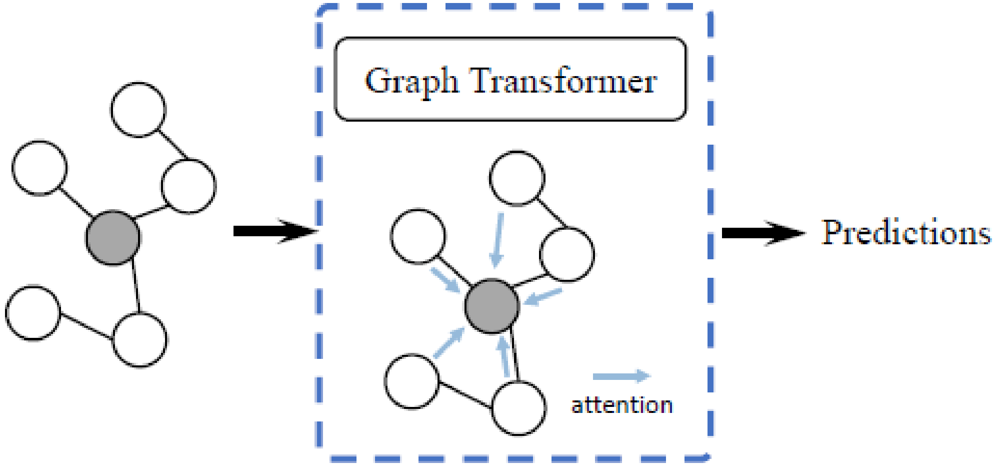
Backbone Architecture

Backbone Architecture

- Message Passing: propagates information between nodes that are explicitly connected in the graph structure
- Graph Transformer: considers and measures the similarity between every pair of nodes in the graph



(a) Message Passing.

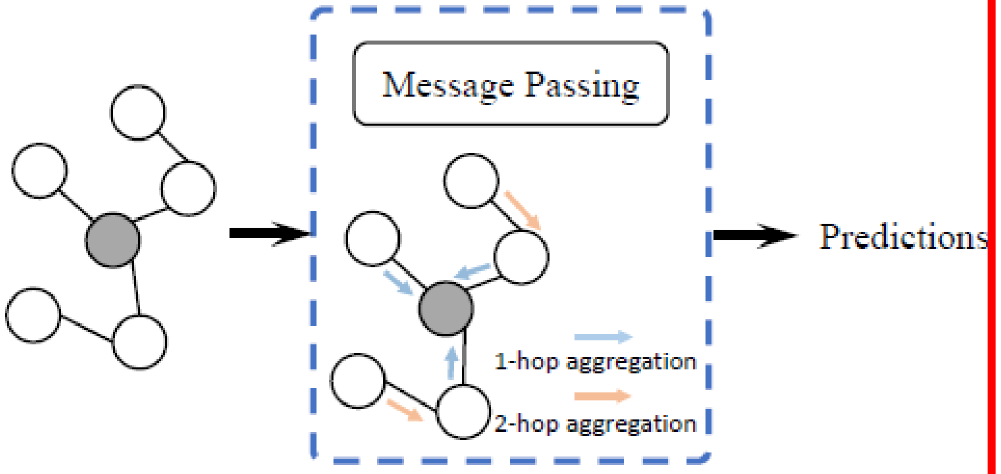


(b) Graph Transformer.

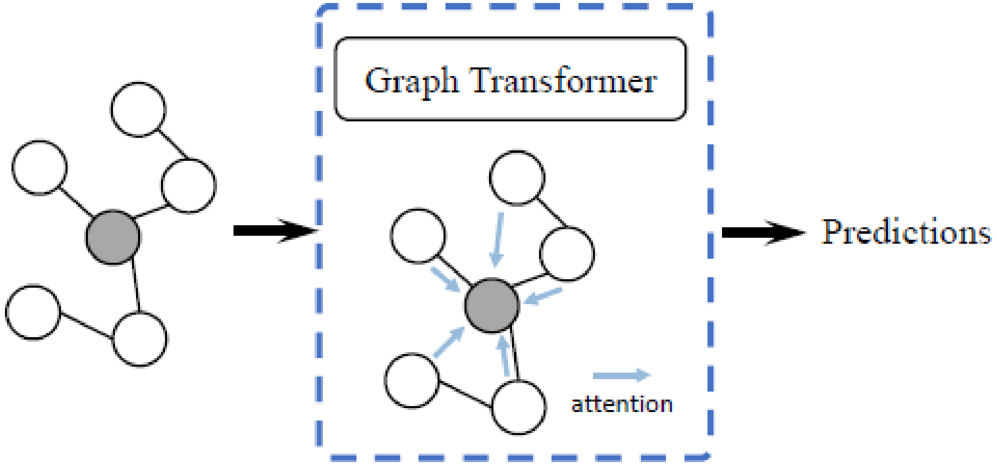
Backbone Architecture

Backbone Architecture

- Message Passing: propagates information between nodes that are explicitly connected in the graph structure
- Graph Transformer: considers and measures the similarity between every pair of nodes in the graph



(a) Message Passing.



(b) Graph Transformer.

Graphormer

Motivation：

- The Transformer is well acknowledged as the most powerful neural network in modelling sequential data, such as natural language and speech.
- Model variants built upon Transformer have also been shown great performance in computer vision and programming Language.
- However, Transformer has still not been the de-facto standard on public graph representation leaderboards.

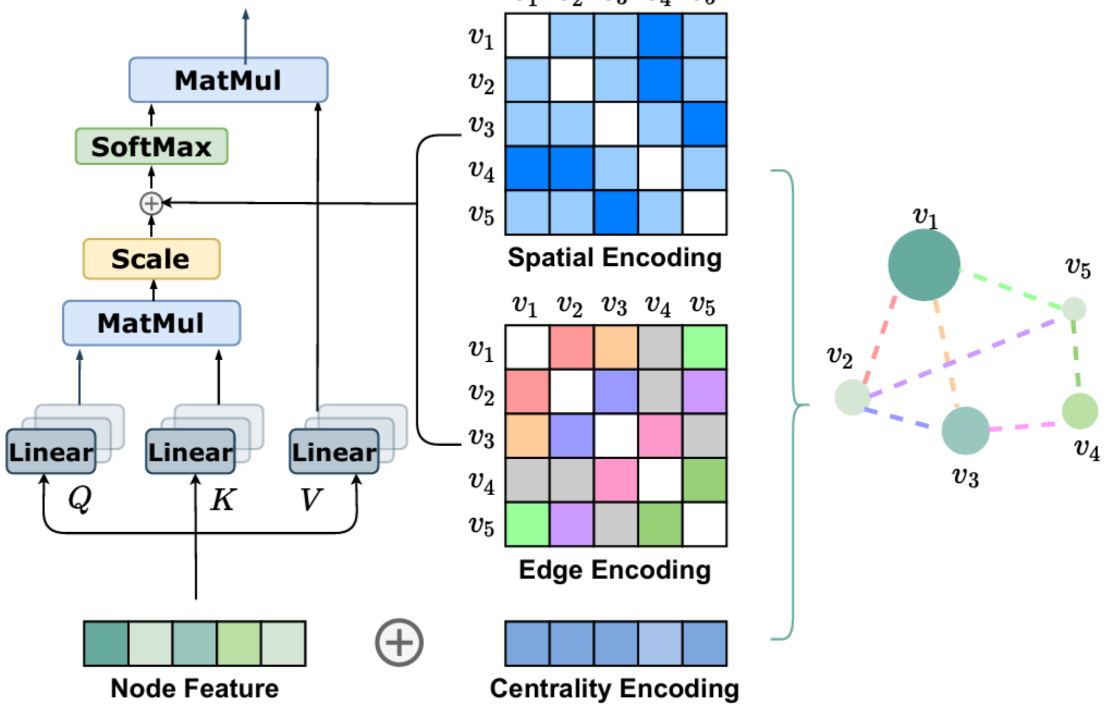
Whether Transformer architecture is suitable to model graphs and how to make it work in graph representation learning?

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? NeurIPS 2021.

Graphormer

Core idea：

- properly incorporate structural information of graphs into the model.
  - propose a Centrality Encoding to capture the node importance in the graph.
  - propose a novel Spatial Encoding to capture the structural relation between nodes
  - design a new Edge Encoding method to take such signal into the Transformer layers.



Graphormer

Structural Encodings in Graphormer：

- Centrality Encoding:
  - develop a Centrality Encoding which assigns each node two real-valued embedding vectors according to its indegree and outdegree.

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

- Spatial Encoding:
  - choose ϕ(vi, vj) to be the distance of the shortest path (SPD) between vi and vj.
- Edge Encoding:
  - find the shortest path SP_{ij} = (e1, e2, ..., eN) from vi to vj, and compute an average of the dot-products of the edge feature and a learnable embedding along the path.

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

GRAPH-BERT

Motivation：

- Traditional message passing-based models have limited representation capabilities.

- The inherently interconnected nature precludes large-sized graph parallelization, as memory constraints limit batching across the nodes.

- Existing GNN models have several serious learning performance problem, e.g., suspended animation problem and over-smoothing problem.

Zhang J, Zhang H, Xia C, et al. Graph-bert: Only attention is needed for learning graph representations. arXiv 2020[J]. arXiv preprint arXiv:2001.05140, 2001.

GRAPH-BERT

Part 1: linkless subgraph batching instead of the complete graph

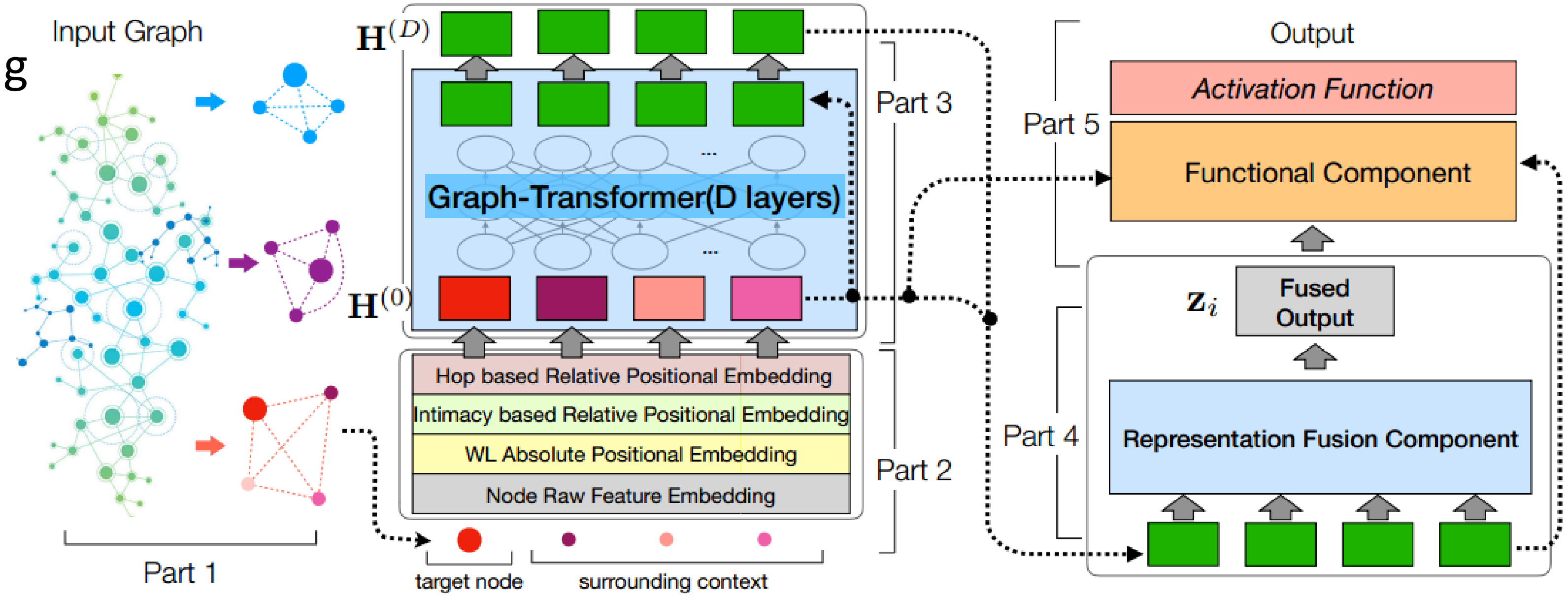
Part 2: Node Input Vector Embeddings

1. raw feature vector embedding
2. Weisfeiler-Lehman absolute role embedding
3. intimacy-based relative positional embedding
4. hop-based relative distance embedding

Part 3: graph transformer based encoder

Part 4: representation fusion

Part 5: functional component



GRAPH-BERT

How to handle large-scale graph input?

- Linkless Subgraph Sampling and Batching
  - Introduce the top-k intimacy sampling approach and trained with linkless subgraph batches sampled from the input graph instead of complete graph

How to deal with insufficient model representation?

- More node-related information as node initial features

1. raw feature vector embedding

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

2. Weisfeiler-Lehman absolute role embedding

$$\begin{aligned}\mathbf{e}_j^{(r)} &= \text{Position-Embed}\left(\text{WL}(v_j)\right)\\ &= \left[sin\left(\frac{\text{WL}(v_j)}{10000^{\frac{2l}{d_h}}}\right), cos\left(\frac{\text{WL}(v_j)}{10000^{\frac{2l+1}{d_h}}}\right)\right]_{l=0}^{\left\lfloor \frac{d_h}{2} \right\rfloor}\end{aligned}$$

3. intimacy-based relative positional embedding

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

4. hop-based relative distance embedding

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

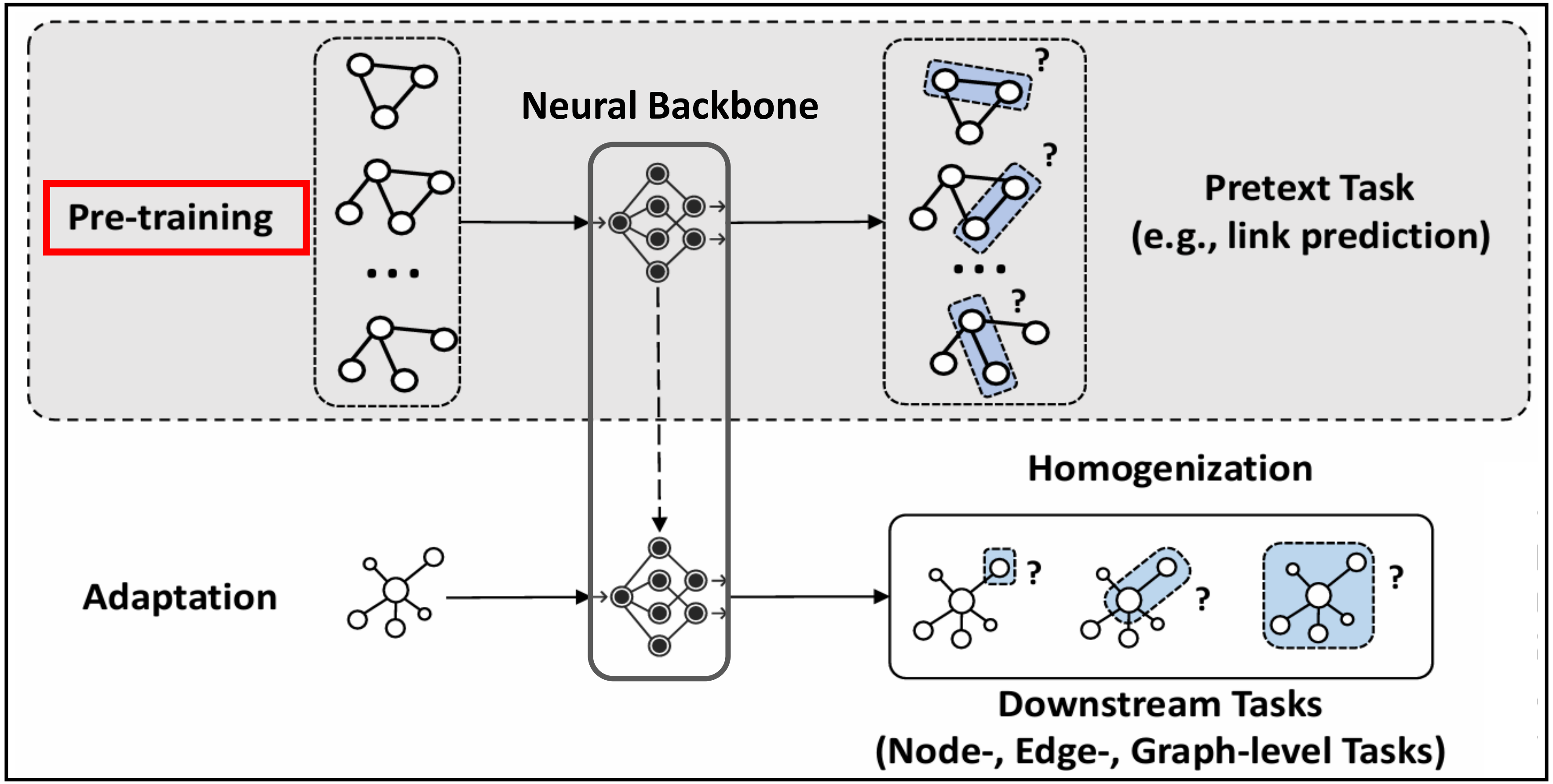
$$\mathbf{h}_j^{(0)} = \text{Aggregate}(\mathbf{e}_j^{(x)}, \mathbf{e}_j^{(r)}, \mathbf{e}_j^{(p)}, \mathbf{e}_j^{(d)})$$

GNN-based Methods

Backbone： No unified architecture

（Message Passing/Graph Transformer）

Paradigm： Pre-training + Adaptation



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

Pre-training

Pre-training

- Generative methods:
    - graph reconstruction
    - property prediction

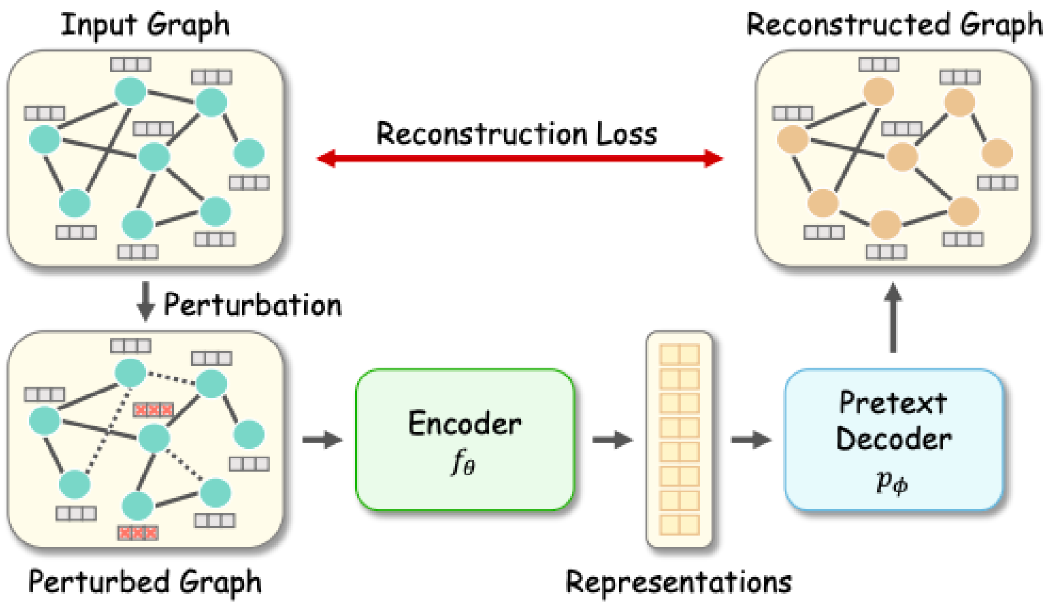
- Contrastive methods:
    - same-scale contrastive learning
    - cross-scale contrastive learning

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.

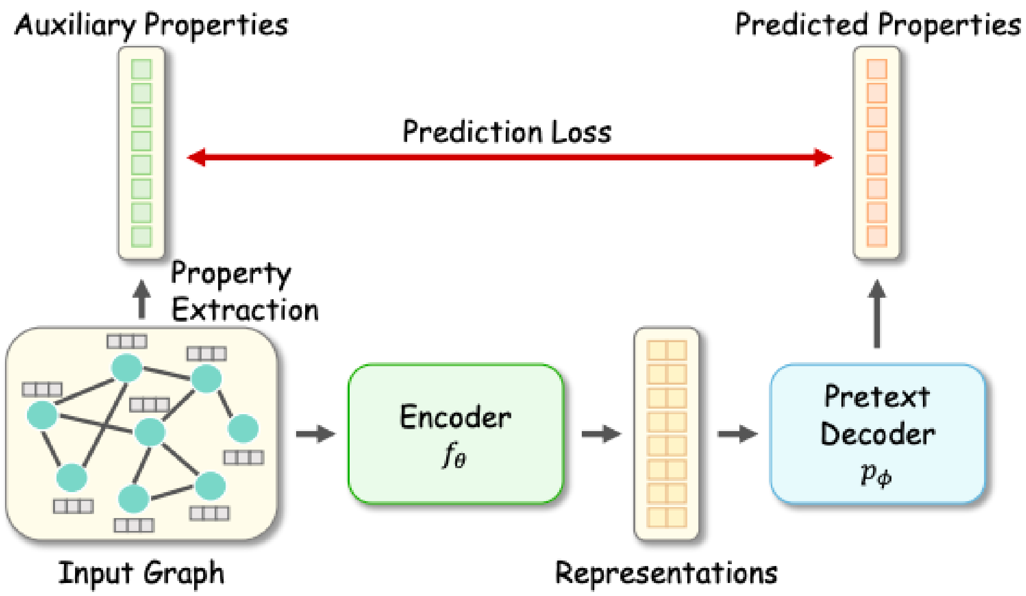
Pre-training

Pre-training

- Generative methods: graph reconstruction, property prediction



graph reconstruction



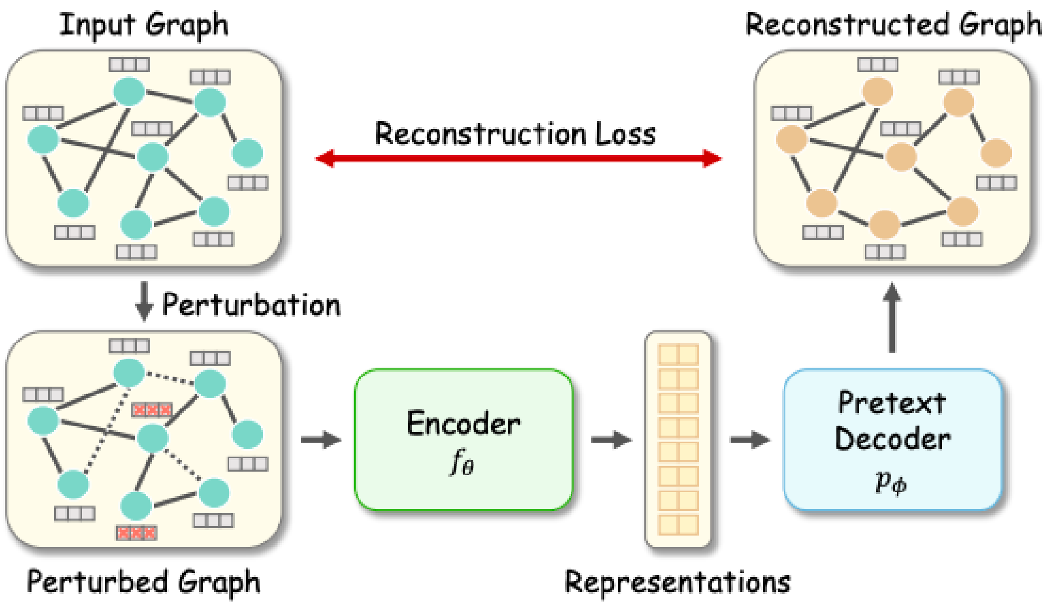
property prediction

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.

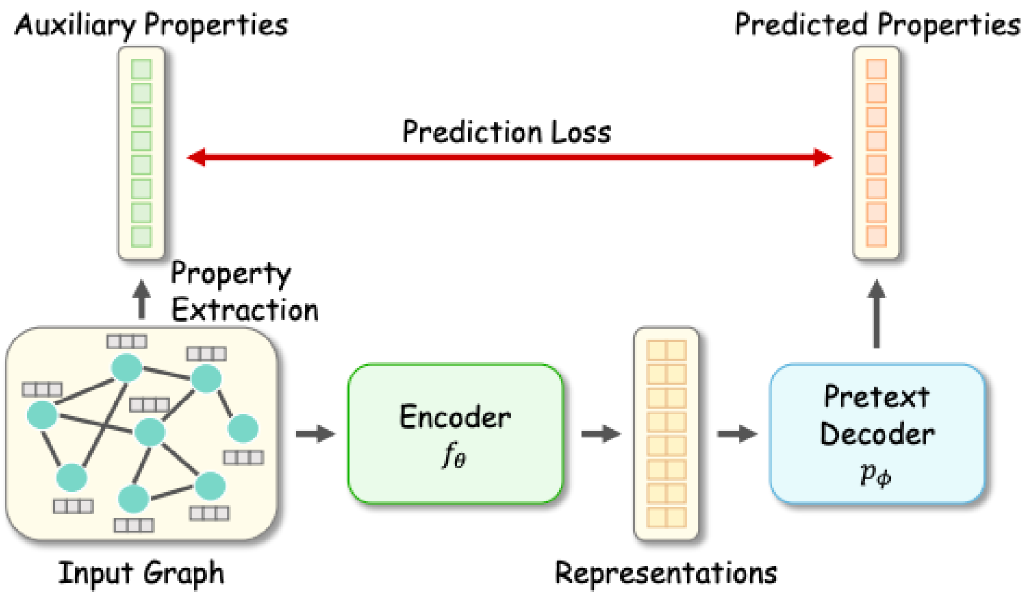
Pre-training

Pre-training

- Generative methods: graph reconstruction, property prediction



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.

GPT-GNN

Motivations:

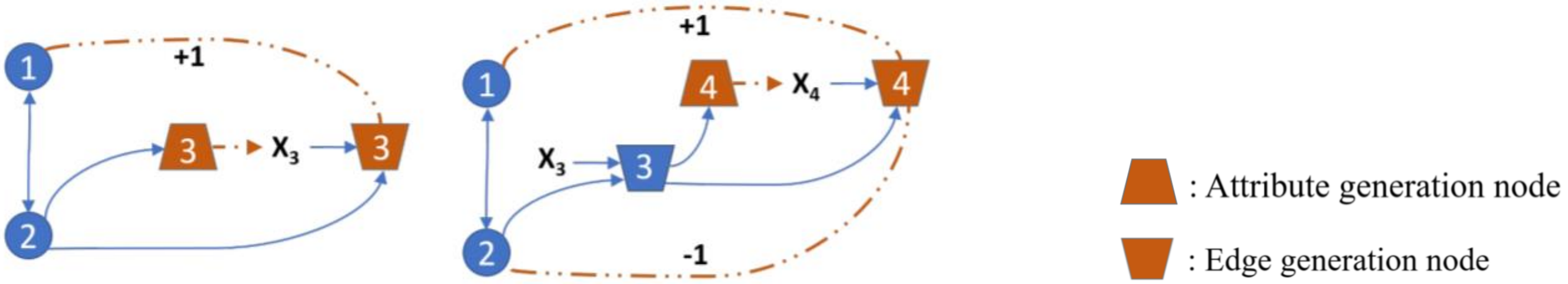
- **Scarcity of Labeled Data**: Adequate labeled data is often unavailable for training GNNs on specific tasks.
- **Proven Success of Pre-training**: Pre-training has significantly enhanced performance in domains like NLP and computer vision.
- **Need for Generalization**: Pre-training GNNs can help them generalize across various tasks with minimal customization.

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

GPT-GNN

Pre-train large-scale graph with **reconstructing** the input graph. Decompose the reconstruction process into two coupled steps:

- Attribute generation: given observed edges, generate **node attributes**
- Edge generation: given observed edges and generated attributes, generate **masked edges**



(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.

Adaptive Graph Encoder

Motivations:

- Reconstructing the adjacency matrix = contrasting adjacent nodes
- Assumption: A node is similar to its neighbors.

Reasonable?

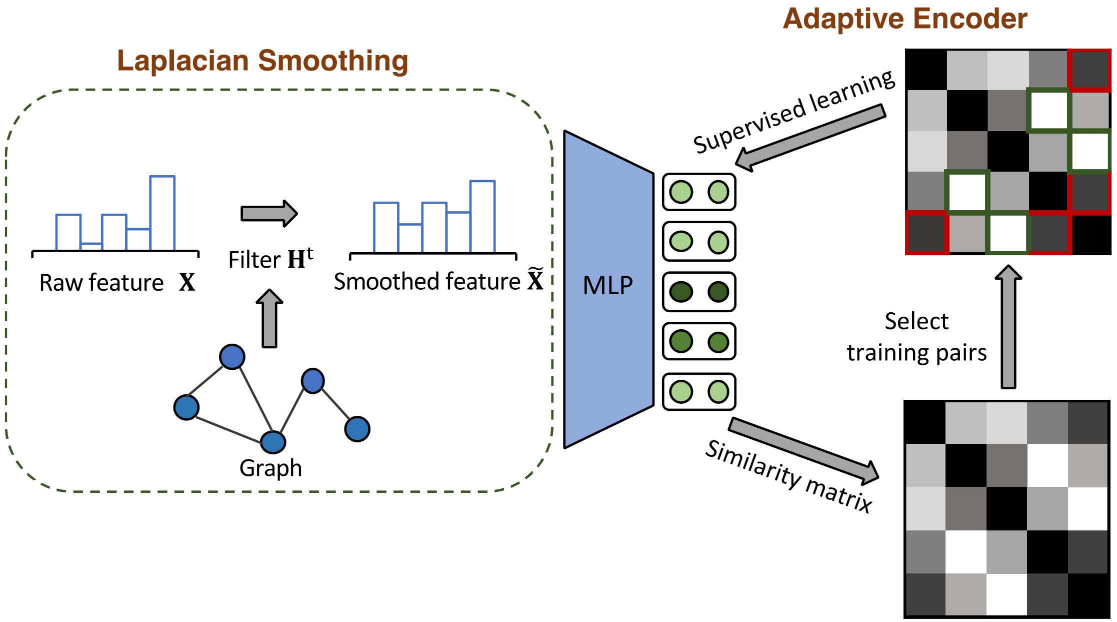
The three main drawbacks of GAE:

- **Entangled Architecture**: Combines multiple layers in a way that complicates training without improving performance.
- **Ineffective Filters**: The graph convolutional filters used are not optimal for filtering out high-frequency noise.
- **Unsuitable Objectives**: The training goals of reconstructing adjacency and feature matrices are not practical, as they can either overlook key data or retain unwanted noise.

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

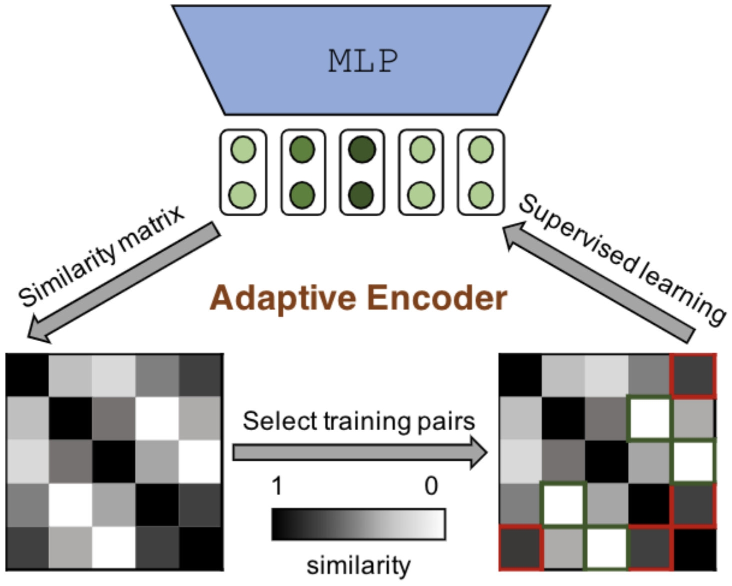
Adaptive Graph Encoder

- **Laplacian Smoothing**: Design appropriate Laplacian smoothing filters to filter out high-frequency noise.
- **Adaptive Encoder**: Adaptively select training node pairs from the node similarity and adjust graph representations accordingly.



Adaptive Graph Encoder

- How to set training objectives to learn graph representations?
  - The adjacency matrix records only one-hop structural information.
  - Smoothed features or trained representations integrate both structural and feature information.
- Adaptively select training node pairs:
  - High similarity pairs as positive examples.
  - Low similarity pairs as negative examples.
- Select Strategy
  - Calculate the cosine similarity matrix S.
  - Sort all node pairs and select those whose similarity is above/below a certain threshold.
  - Dynamically update the threshold.

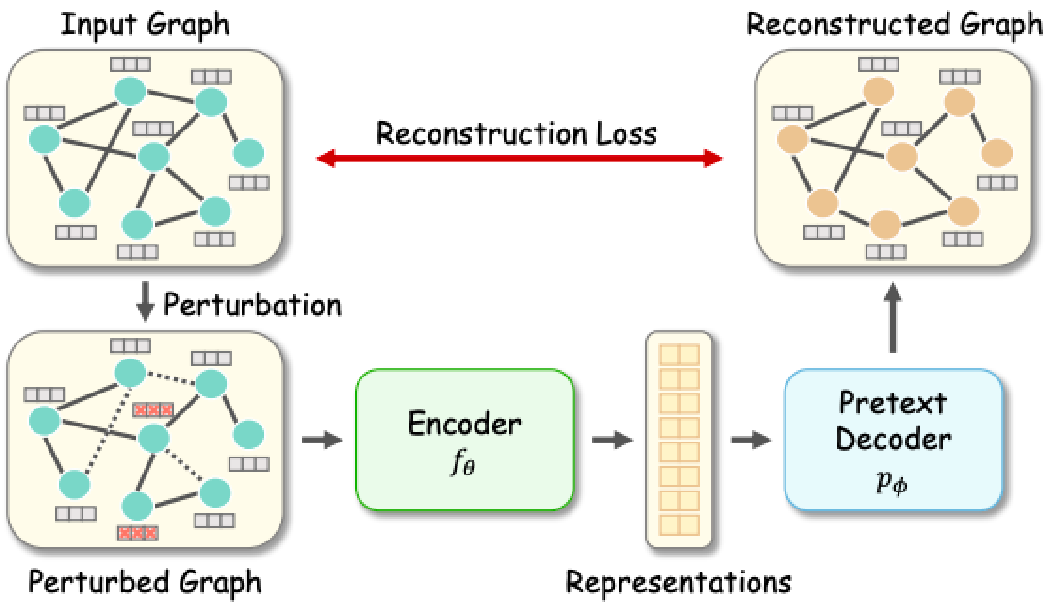


Cui G, Zhou J, Yang C, et al. Adaptive graph encoder for attributed graph embedding. KDD 2020

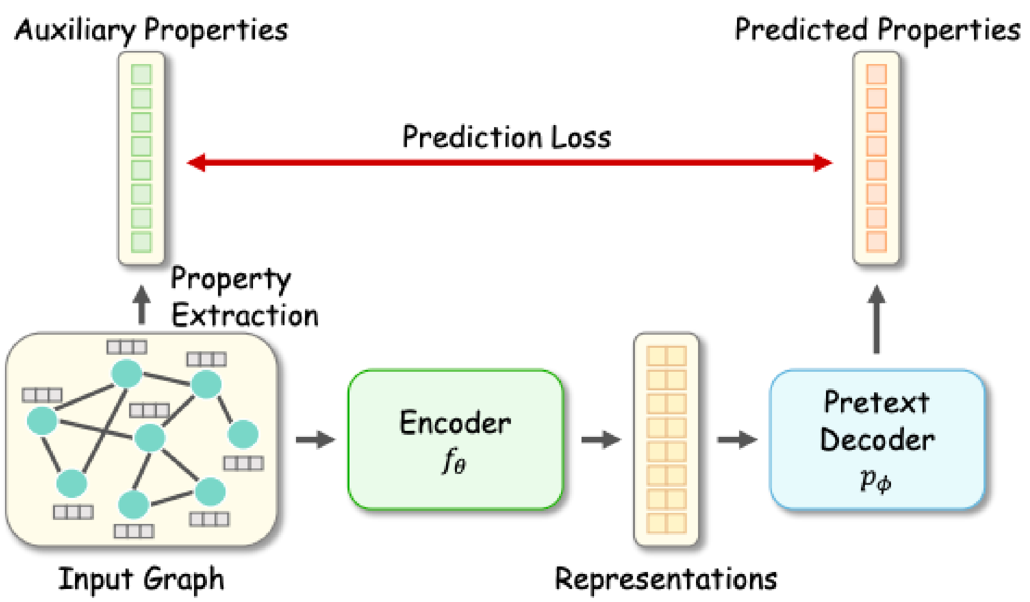
Pre-training

Pre-training

- Generative methods: graph reconstruction, property prediction



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.

GraphMAE

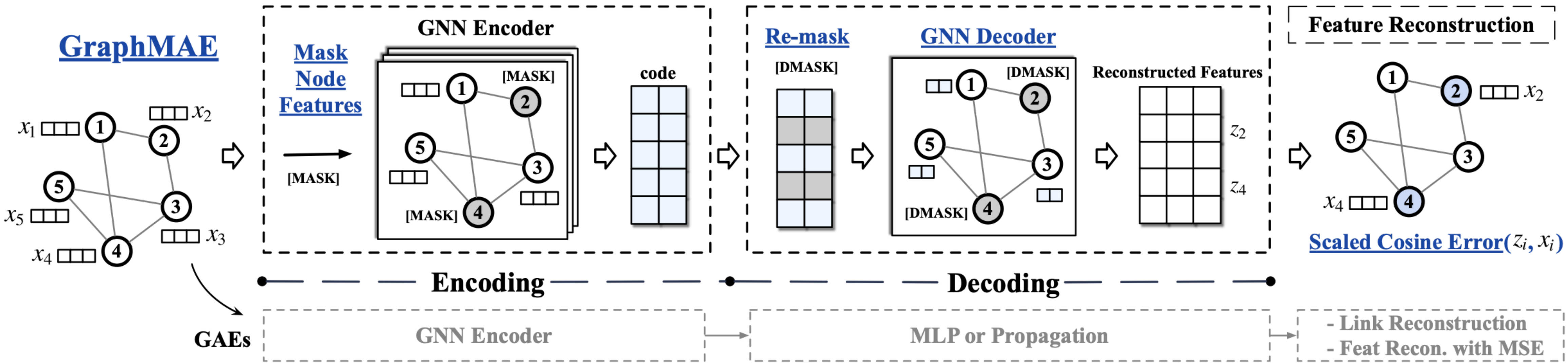
Motivations:

- **Lagging Development of GAEs**: GAEs lag behind contrastive methods in critical tasks like node and graph classifications, highlighting a need for enhanced models.
- **Challenges in Current GAE Approaches**: Existing GAEs struggle with issues like non-robust feature reconstruction and sensitivity to MSE, prompting the need for methodological improvements.
- **Decoder Limitations**: The simple MLP decoders commonly used in GAEs are inadequate for complex graph data, suggesting a need for more expressive architectures.

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.

GraphMAE

Masked Feature Reconstruction: Focuses on node feature reconstruction with masking, proven effective in enhancing performance.
Scaled Cosine Error: Uses a scaled cosine error for better handling of feature magnitude variations and sample difficulty imbalances.
Re-mask Decoding: Employs re-masking of encoded node embeddings to improve decoding accuracy.
Advanced Decoder Architecture: Incorporates complex GNNs in the decoder for improved expressiveness.



GROVER

Motivations:

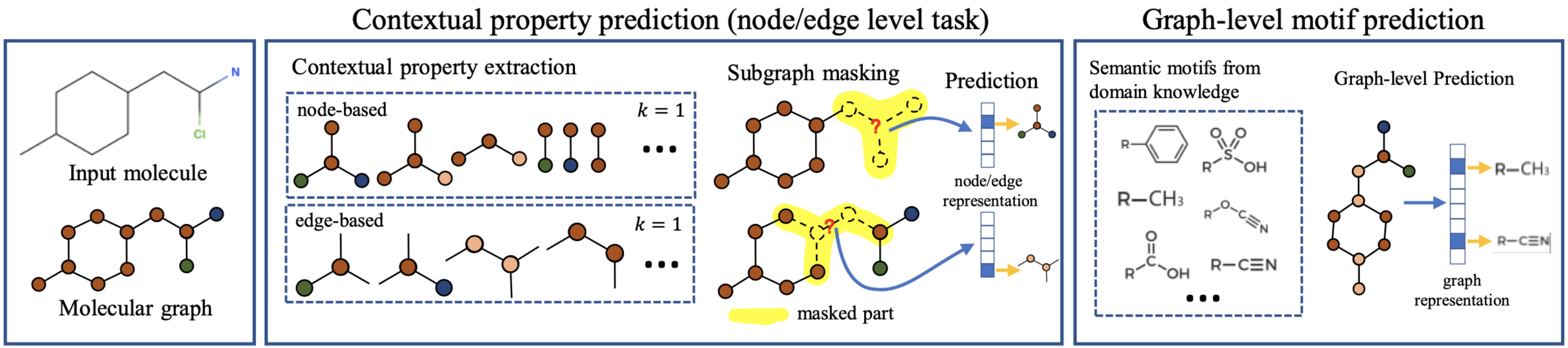
- **Scarcity of Labeled Data**: There is a significant lack of labeled molecular data, making it challenging to apply traditional supervised learning effectively in drug discovery.
- **Limitations of Current Methods**: Existing molecular representation methods, like SMILES, fail to adequately capture the complex topological information of molecules.
- **Need for Novel Strategies**: There is a pressing need for novel computational strategies that can efficiently exploit vast amounts of unlabeled molecular data to improve prediction accuracy and model generalization.

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

GROVER

Designed self-supervised tasks in node-, edge- and graph-level, learn rich **structural** and **semantic** information of molecules

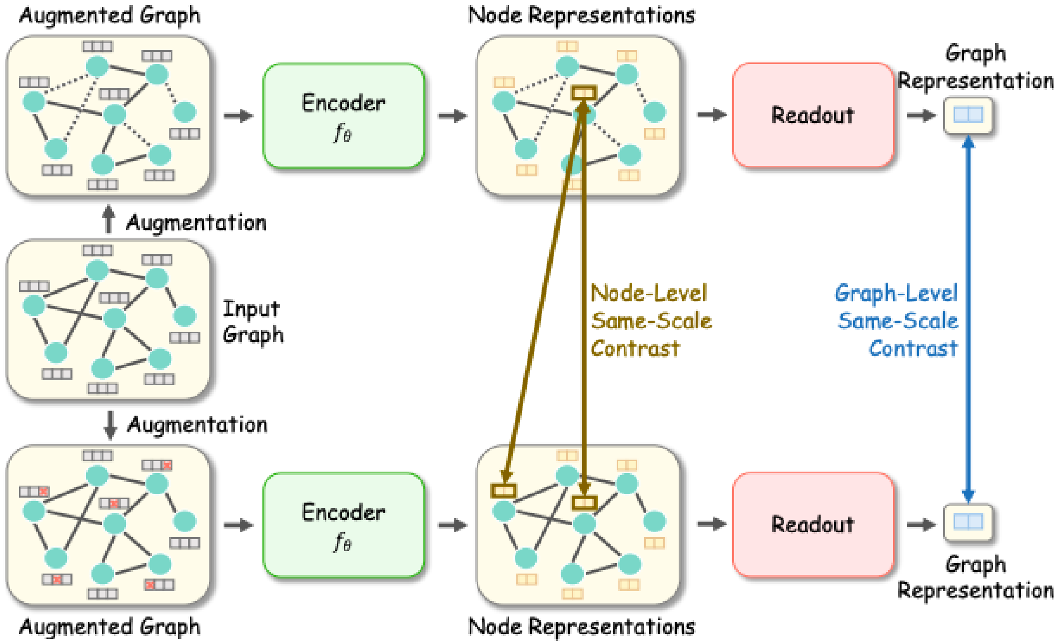
- Contextual property prediction: predict **masked node/edge** attributes set
- Motif prediction : predict the classes of **the motif** that occur in a given molecule



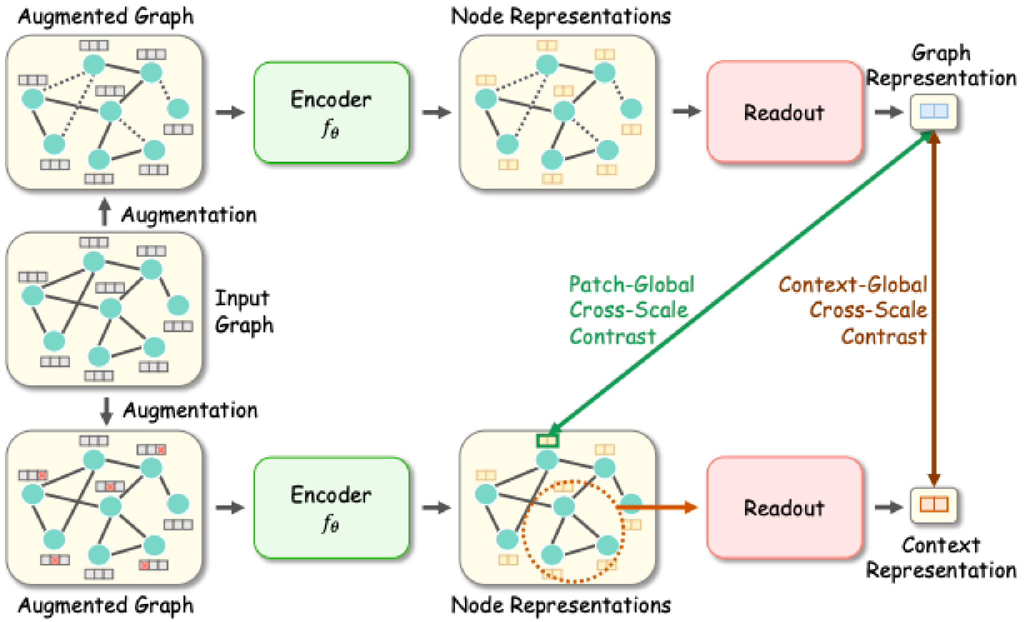
GNN-based Models

Pre-training

- Contrastive methods: same-scale contrastive learning, cross-scale contrastive learning



same-scale contrastive learning



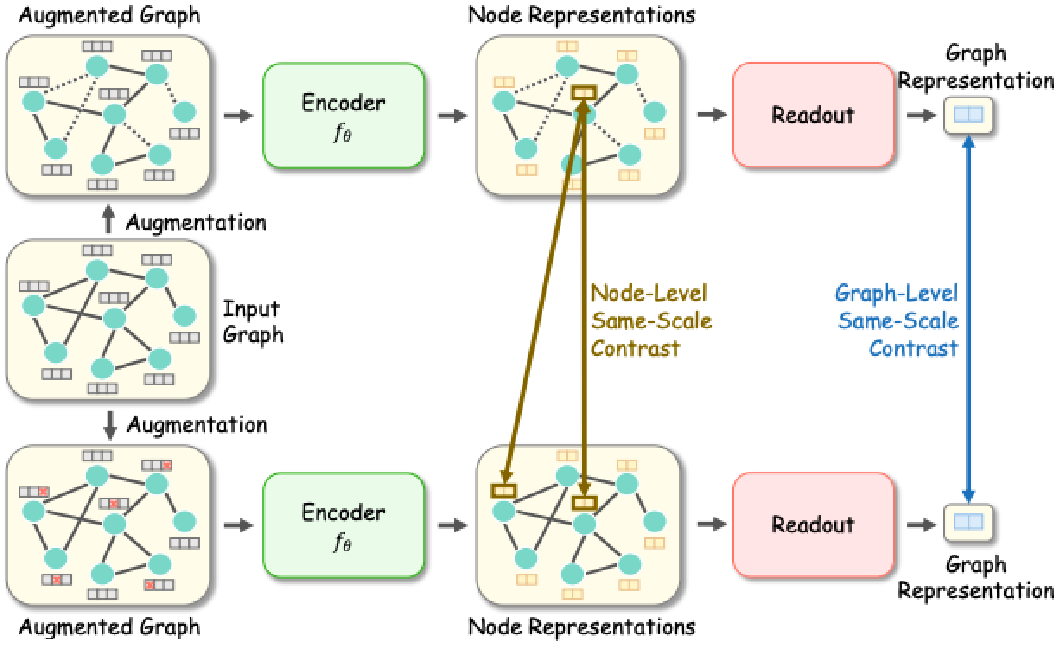
cross-scale contrastive learning

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.

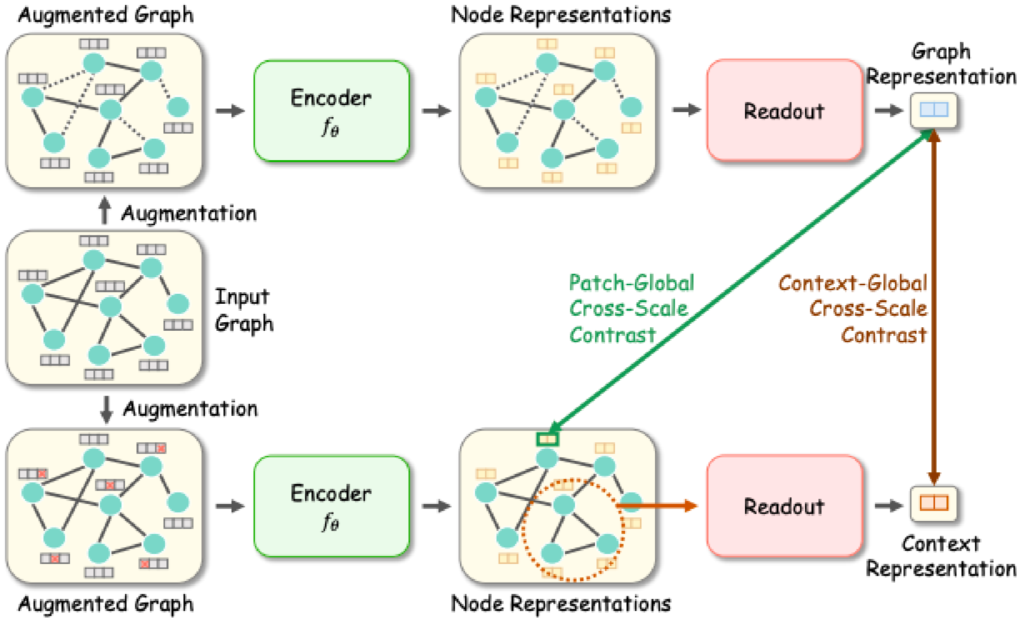
GNN-based Models

Pre-training

- Contrastive methods: same-scale contrastive learning, cross-scale contrastive learning



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.

GraphCL

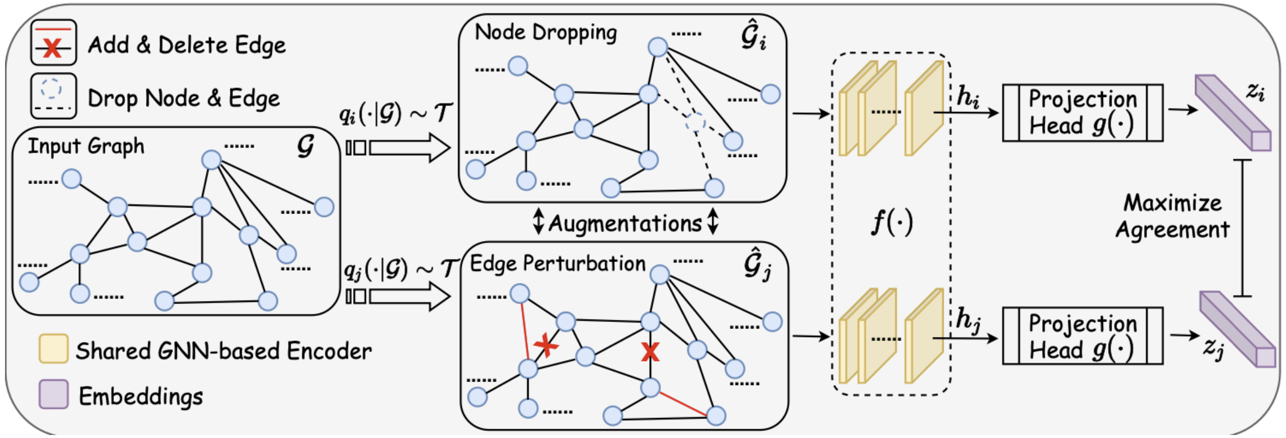
Motivations

- **Label Scarcity:** In fields like biology and chemistry, acquiring labels is costly and slow, making pre-training a valuable strategy to enhance GNNs, akin to its use in CNNs.
- **Complex Graph Data:** The diverse and complex nature of graph data makes designing effective pre-training schemes challenging, as simple methods like adjacency reconstruction may fall short.
- **Contrastive Learning Potential:** Contrastive learning could potentially overcome the limitations of proximity-focused pre-training by promoting feature consistency across different views.

Yuning You, Tianlong Chen, Yongduo Sui, et al. Graph Contrastive Learning with Augmentations. NeurIPS2020

GraphCL

Design four types of **graph augmentations** to incorporate various impacts in four different settings: semi-supervised, unsupervised, transfer learning and adversarial attacks.



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

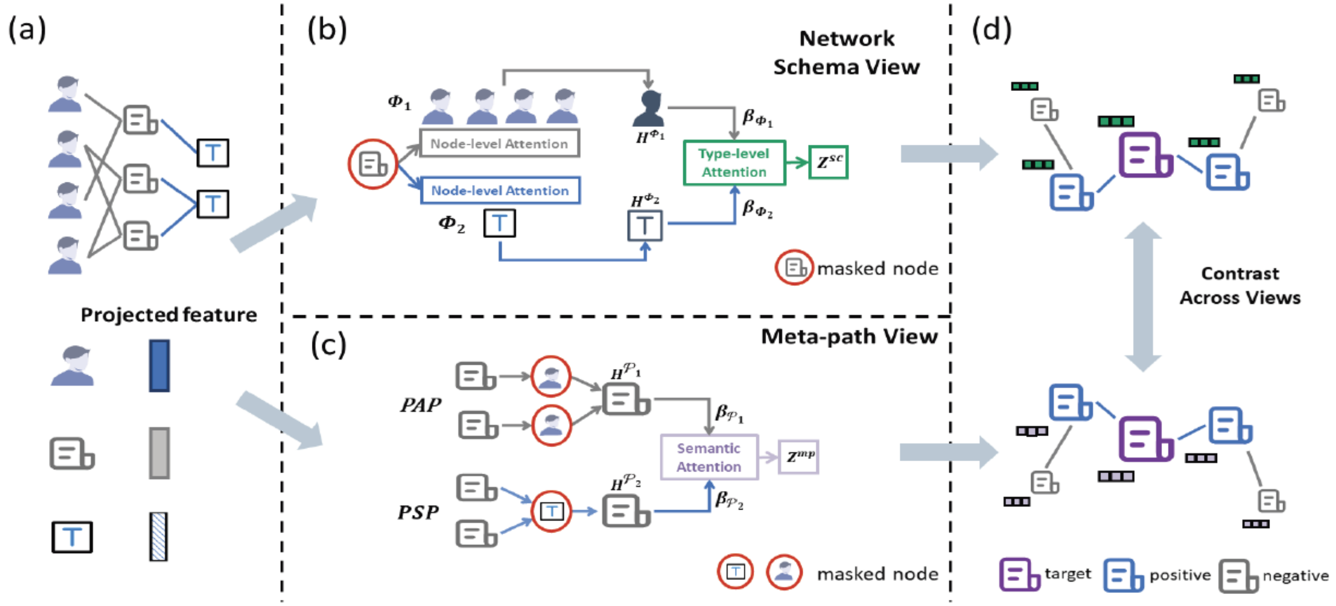
Contrast augmentation of same&diffferent graphs

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.

HeCo

Core idea:

- Two views of a HIN (network schema and meta-path views) are proposed to capture both of local and high-order structures simultaneously.
- The cross-view contrastive learning is proposed to extract the positive and negative embeddings from two views.
- The two views to collaboratively supervise each other and finally learn high-level node embeddings.



Wang X, Liu N, Han H, et al. Self-supervised heterogeneous graph neural network with co-contrastive learning. KDD 2021

HeCo

Network Schema View Guided Encoder

Node-level attention

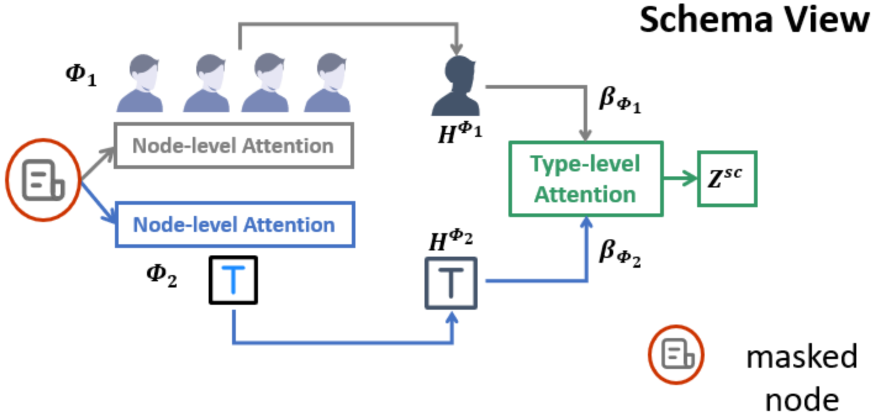
For target paper

1. randomly sample Φ_m-type neighbors with threshold T_{Φ_m}
2. aggregate them with attention to get embedding of type Φ_m

Type-level aggregation

Aggregate different type of nodes with attention mechanism

Network Schema View



Meta-path View Guided Encoder

Meta-path specific GCN

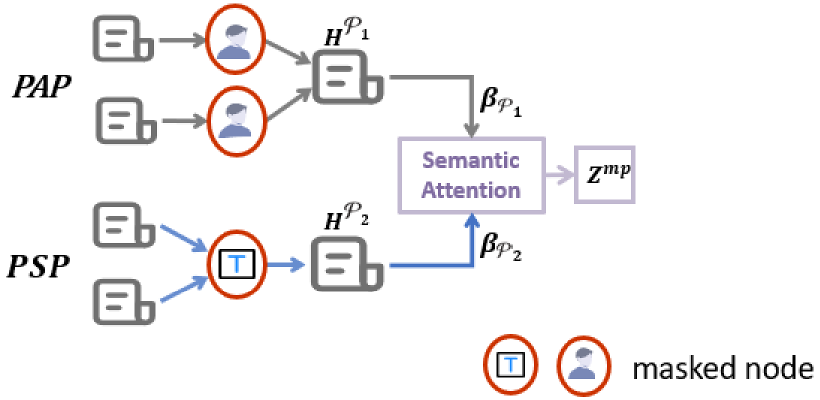
For target paper

1. find its neighbors based on different meta-paths
2. aggregate them with meta-path based vanilla GCN

Semantic-level aggregation

Aggregate embedding under different meta-path with attention mechanism

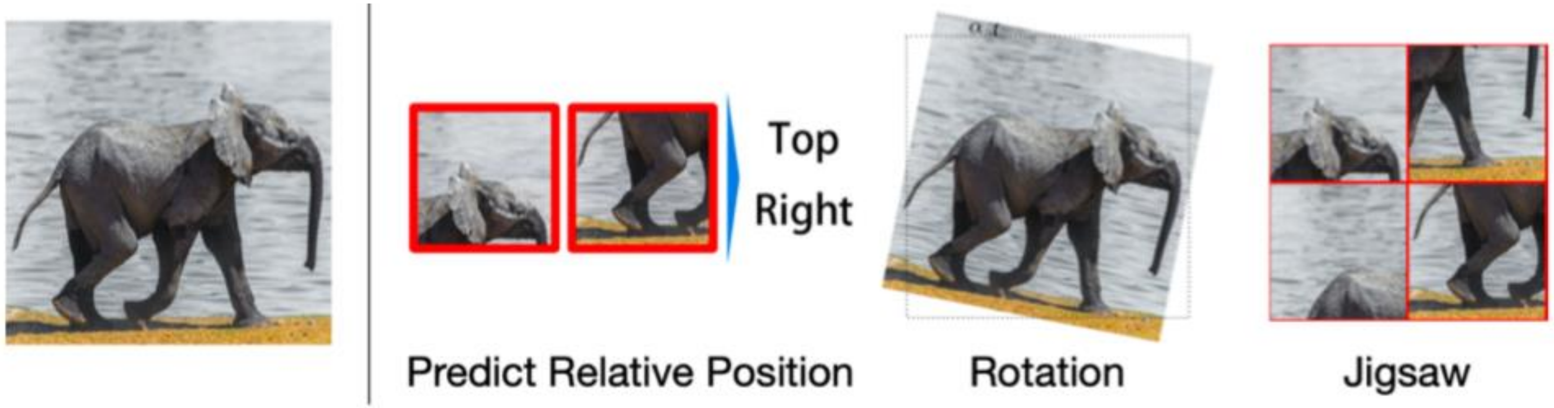
Meta-path View



MA-GCL

Motivation

- Contrastive learning captures invariant information among different augmentation views.
- Good augmentations should introduce as much perturbation as possible without changing the core semantics.



- However, in graph contrastive learning (GCL), we have few prior knowledge on how to generate such good augmentations.

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

MA-GCL

Core idea

- We interpret a GNN as a sequence of propagation operator *g* and transformation operator *h*:
  - propagation operator *g* is typically the non-parametric graph filter.
  - transformation operator *h* is typically a weight matrix with a non-linear function.

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

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

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

- Intuition: different architectures (i.e., operator sequences) won't affect the core semantics.
- Thus we perturb the neural architecture of graph encoder as model augmentations.

MA-GCL

We propose three strategies to introduce perturbations:

- Asymmetric strategy
  - Use the same number of operator h with shared parameters for different views
  - Use different numbers of operator g for different views
- Random strategy
  - Randomly vary the number of propagation operator g in every training epoch
- Shuffling strategy
  - Randomly shuffle the permutation of propagation and transformation operators

MA-GCL

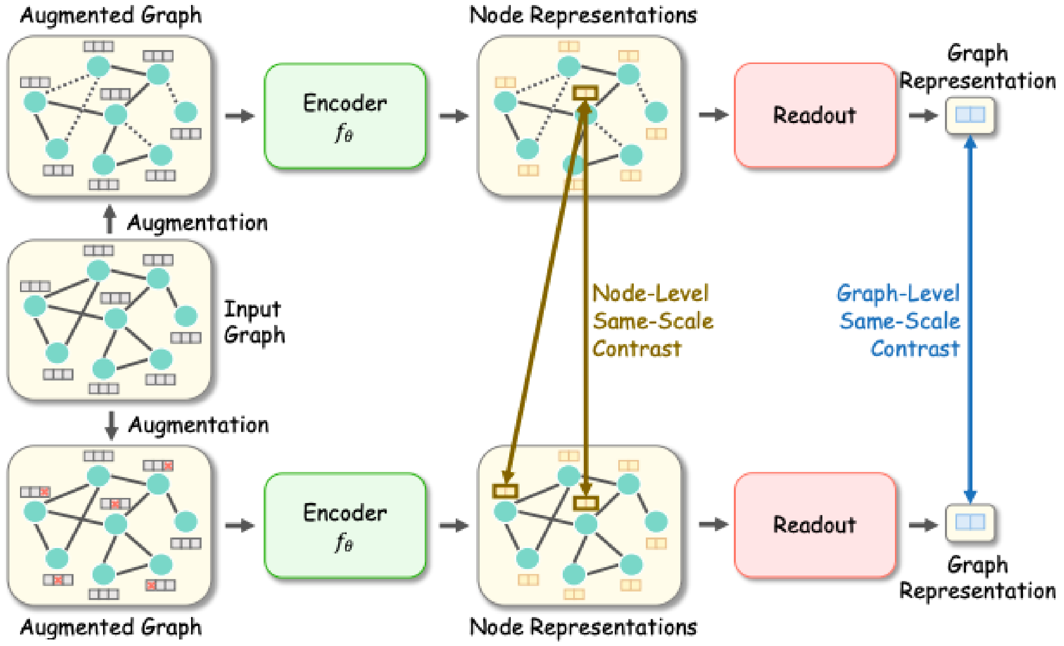
We conducted extensive experiments on node/graph classification/clustering.

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	71.5 ± 0.5	80.7 ± 0.4	92.93 ± 0.0	87.46 ± 0.2	92.15 ± 0.2	84.44	6.5
MVGRL	83.4 ± 0.3	73.0 ± 0.3	80.1 ± 0.6	92.11 ± 0.1	87.52 ± 0.1	91.74 ± 0.0	84.63	6.5
BGRL	81.7 ± 0.5	72.1 ± 0.5	80.2 ± 0.4	93.01 ± 0.2	88.23 ± 0.3	<u>92.57 ± 0.3</u>	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	<u>93.45 ± 0.4</u>	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	82.40	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**	<u>73.1 ± 0.3</u>	<u>81.3 ± 0.4</u>	93.37 ± 0.2	<u>88.42 ± 0.3</u>	92.44 ± 0.1	<u>85.42</u>	<u>2.3</u>
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**

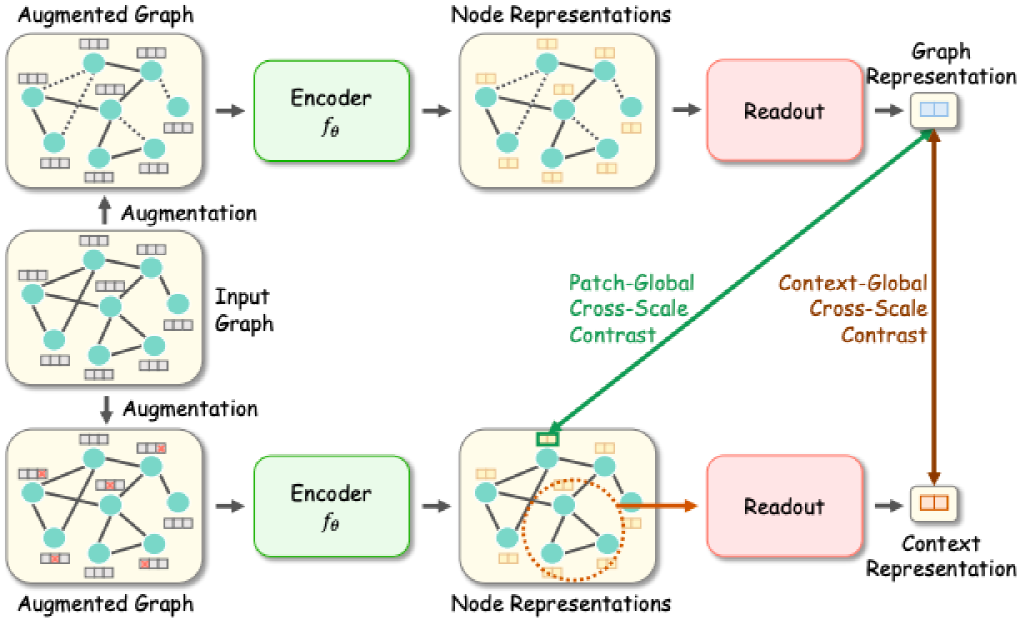
GNN-based Models

Pre-training

- Contrastive methods: same-scale contrastive learning, cross-scale contrastive learning



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.

Deep Graph Infomax

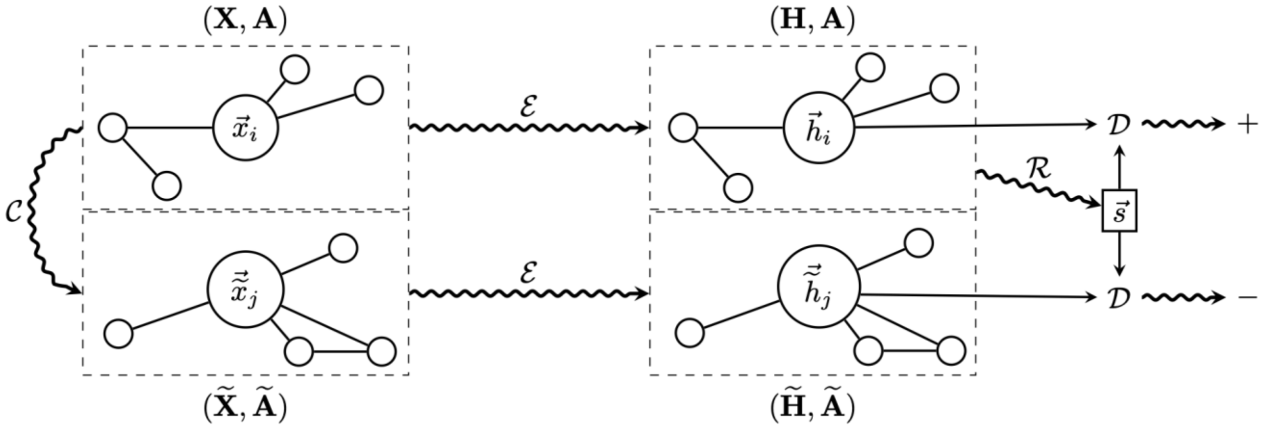
Motivations:

- **Label Scarcity**: Most real-world graph data lacks labels, restricting the use of supervised methods.
- **Structure Discovery**: Unsupervised learning is vital for uncovering new structures in large-scale graphs.
- **Current Method Limitations**: Existing methods like random walks over-emphasize proximity and may neglect broader structural details.

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

Deep Graph Infomax

- **Node Representation**: GCN generates a representation for each node in the graph.
- **Graph Representation**: The global representation of the graph is produced by aggregating all node representations, typically through summation or averaging.
- **Negative Sampling**: Perturbed versions of the graph are generated, for example, by shuffling node features or edges to create negative samples.
- **Maximization of Mutual Information**: The network is trained by maximizing the mutual information between node representations and the global representation in the positive samples (original graph) and minimizing it in the negative samples (perturbed graph).

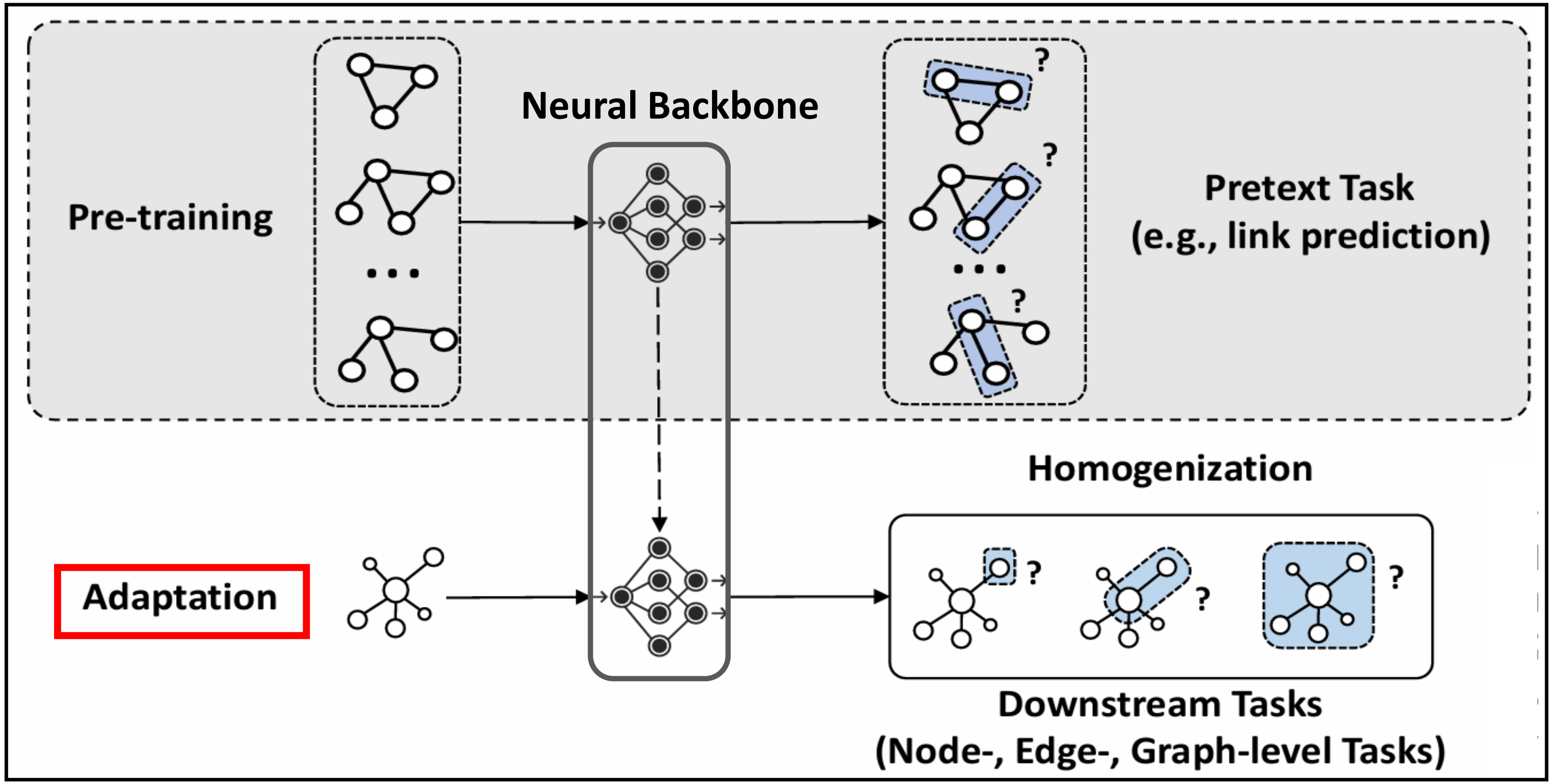


GNN-based Methods

Backbone： No unified architecture

（Message Passing/Graph Transformer）

Paradigm： Pre-training + Adaptation

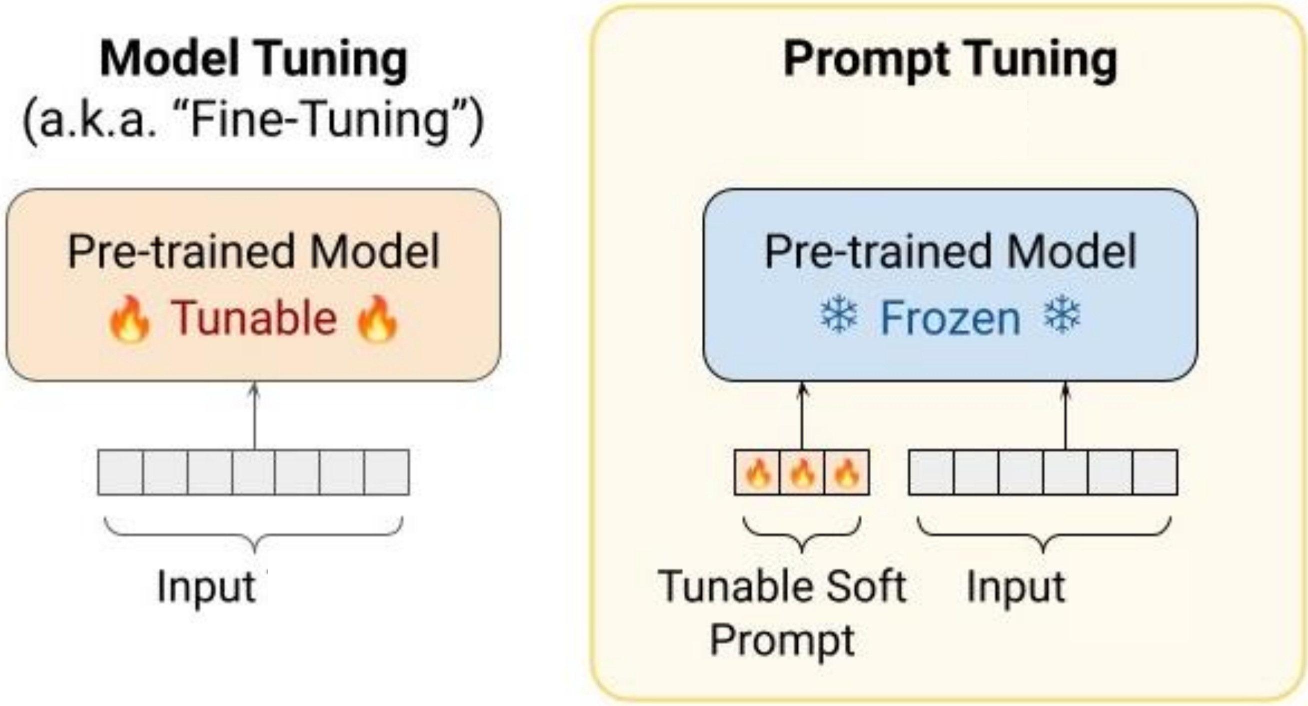


Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, Chuan Shi. Towards Graph Foundation Models: A Survey and Beyond. arXiv 2023

Adaptation

Downstream Adaptation

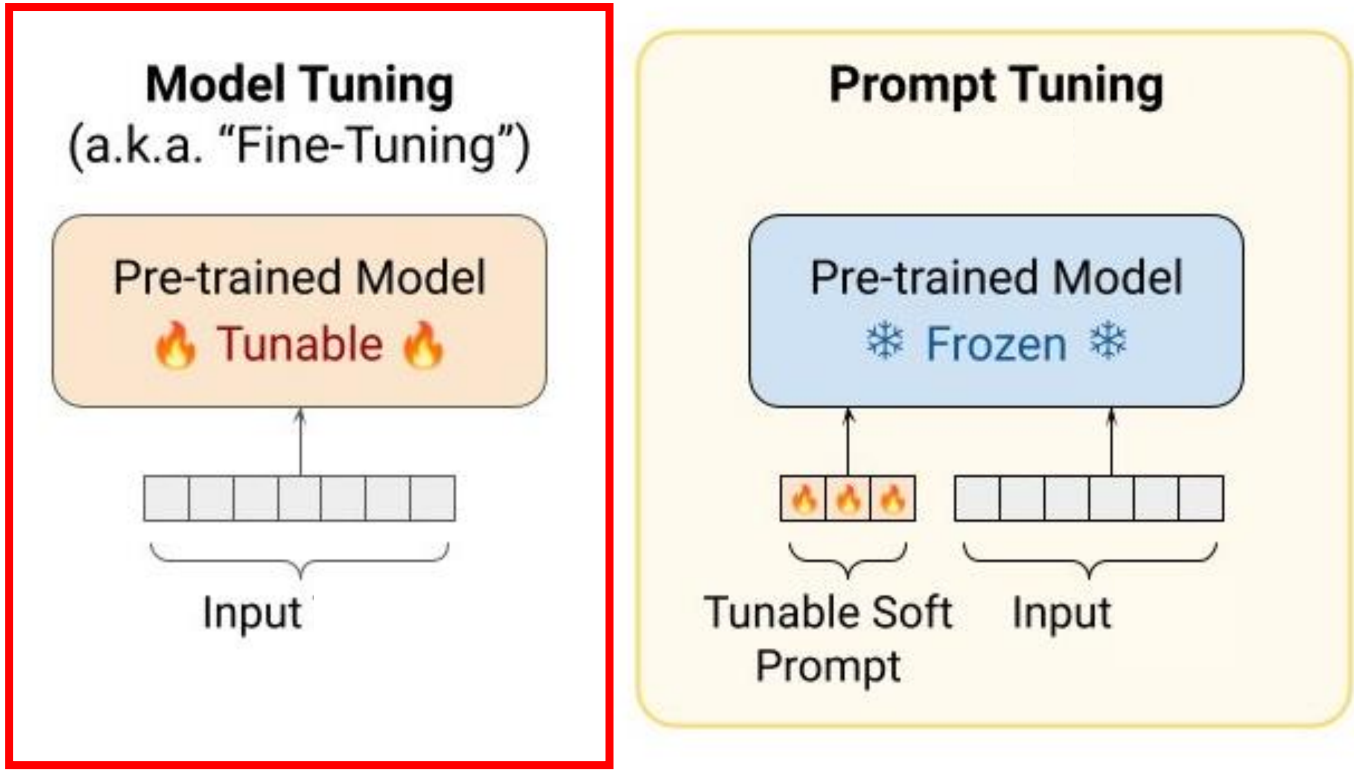
- Fine-tuning: keep input graph intact, modify model parameters accordingly
- Prompt-tuning: keep pre-trained model intact, modify input graph or output embedding



Adaptation

Downstream Adaptation

- Fine-tuning: keep input graph intact, modify model parameters accordingly
  - Parameter-efficient Fine-tuning (PEFT): only tune a small portion of parameters

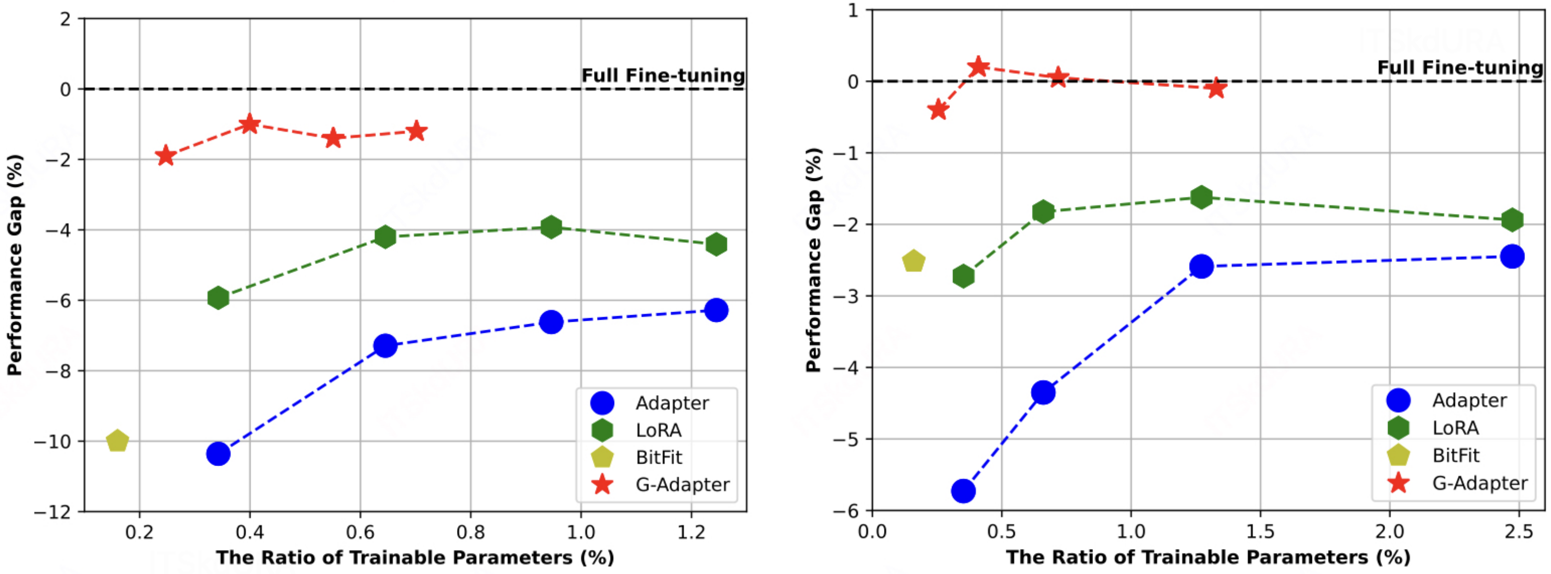


G-Adapter

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?



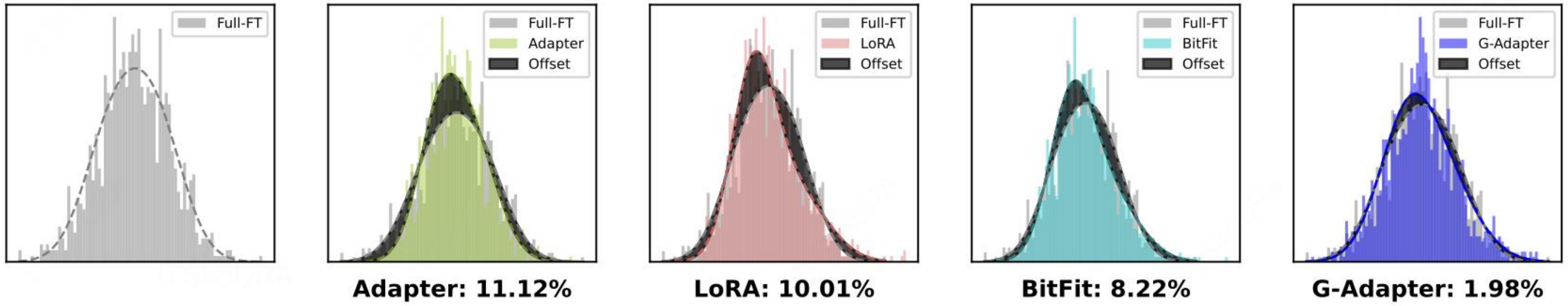
(a) On large-scale datasets. (b) On small-scale datasets.

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

G-Adapter

Method

- Exploration in this paper reveals the feature distribution shift issue due to the absence of graph structure in the fine-tuning process.
- To alleviate these concerns, a novel structure-aware PEFT method, G-Adapter, is proposed, which leverages graph convolution operation to introduce graph structure as the inductive bias to guide the updating process.
- They apply the low-rank decomposition to the learnable weights, which makes G-Adapter highly lightweight.



AdapterGNN

Motivation

- 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?

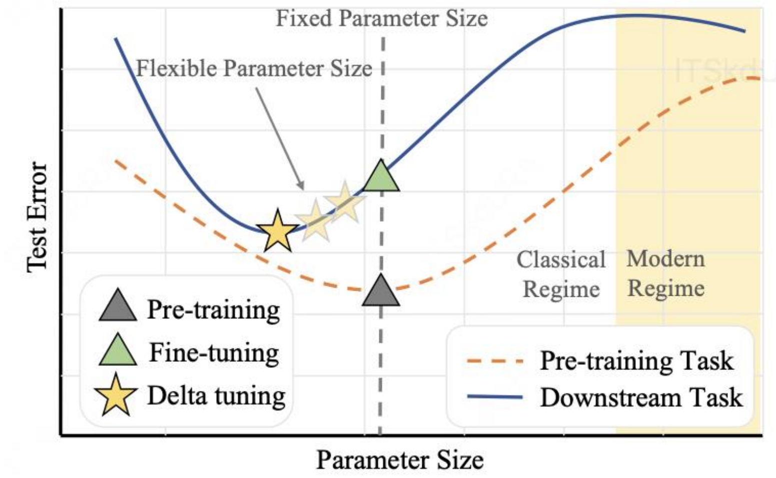
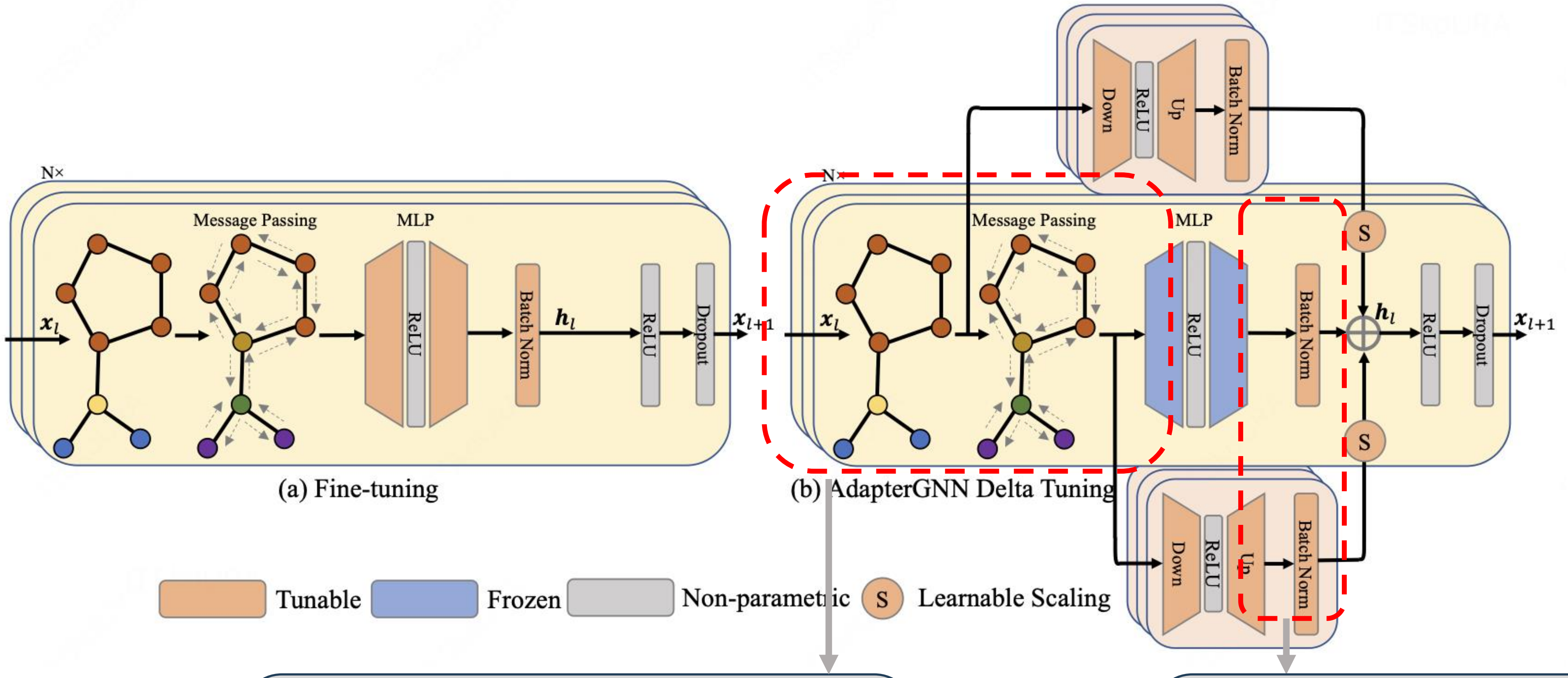


Figure 1: A large model is often employed for pre-training ▲ when sufficient data is available. However, for downstream tasks with limited data, a smaller model is optimal in the classical regime. Compared with fine-tuning ▲, delta tuning ★ preserves expressivity while reducing the size of parameter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

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

AdapterGNN



These adapters utilize bottleneck architecture to significantly reduce the number of tunable parameters by reducing intermediate dimensions.

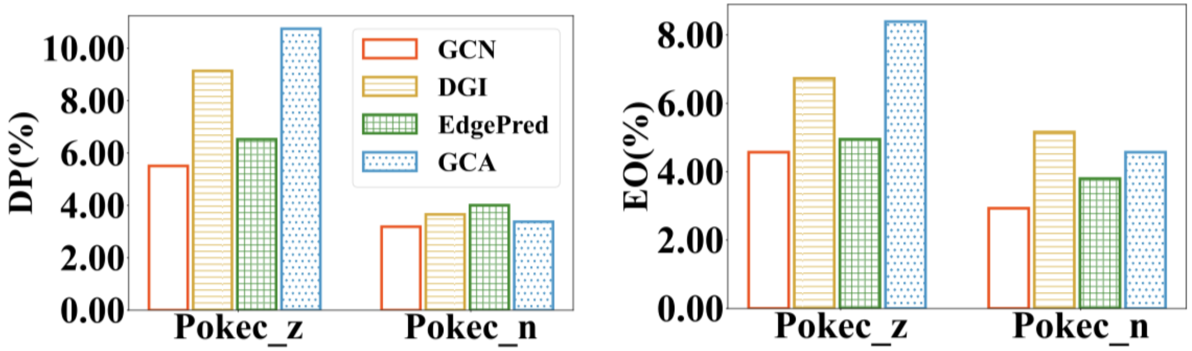
AdapterGNN introduces trainable BN layers in each adapter to maintain consistency with the output of the backbone network.

GraphPAR

Background

- 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

GraphPAR

Existing fair methods is inflexible and inefficient.

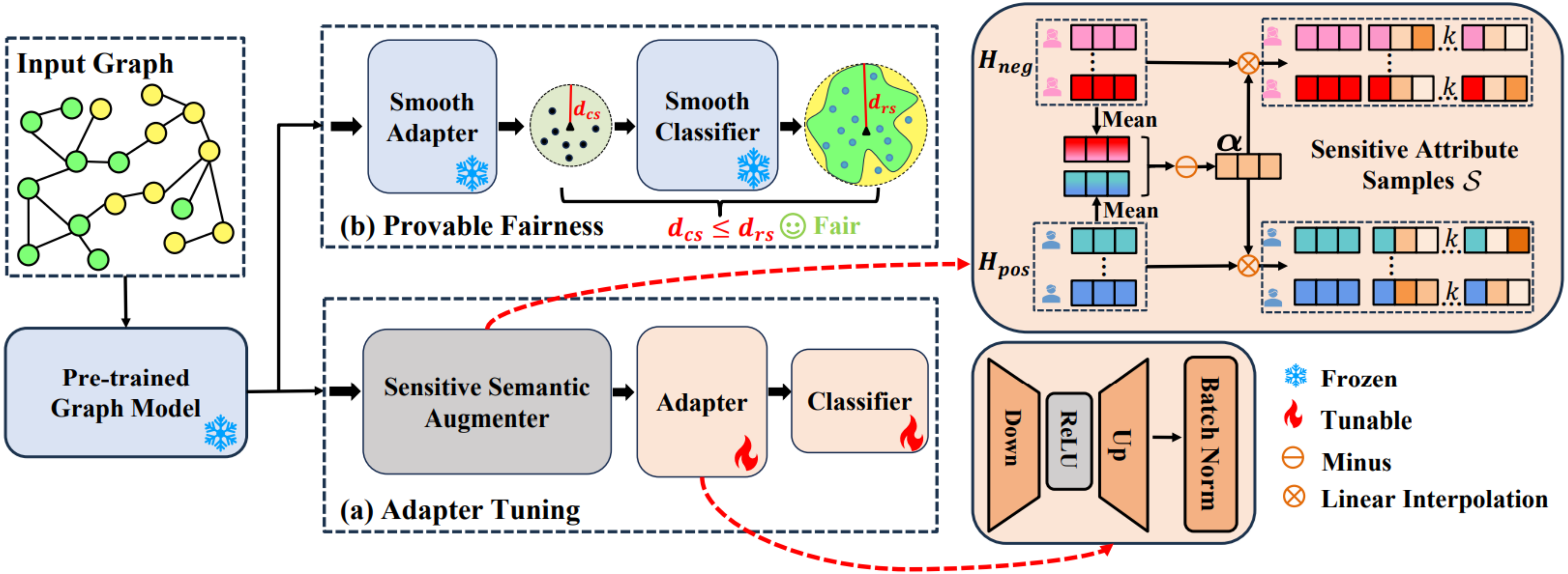
- Existing works generally train a fair GNN for a specific task.
- Debiasing for a specific task in the pre-training phase is inflexible, and maintaining a specific PGM for each task is inefficient.

Existing fair methods lack theoretical guarantees.

- Provable lower bounds on the fairness of model prediction.

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

GraphPAR



Augmenting sensitive attribute semantics

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

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

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

Training adapter for PGMs fairness

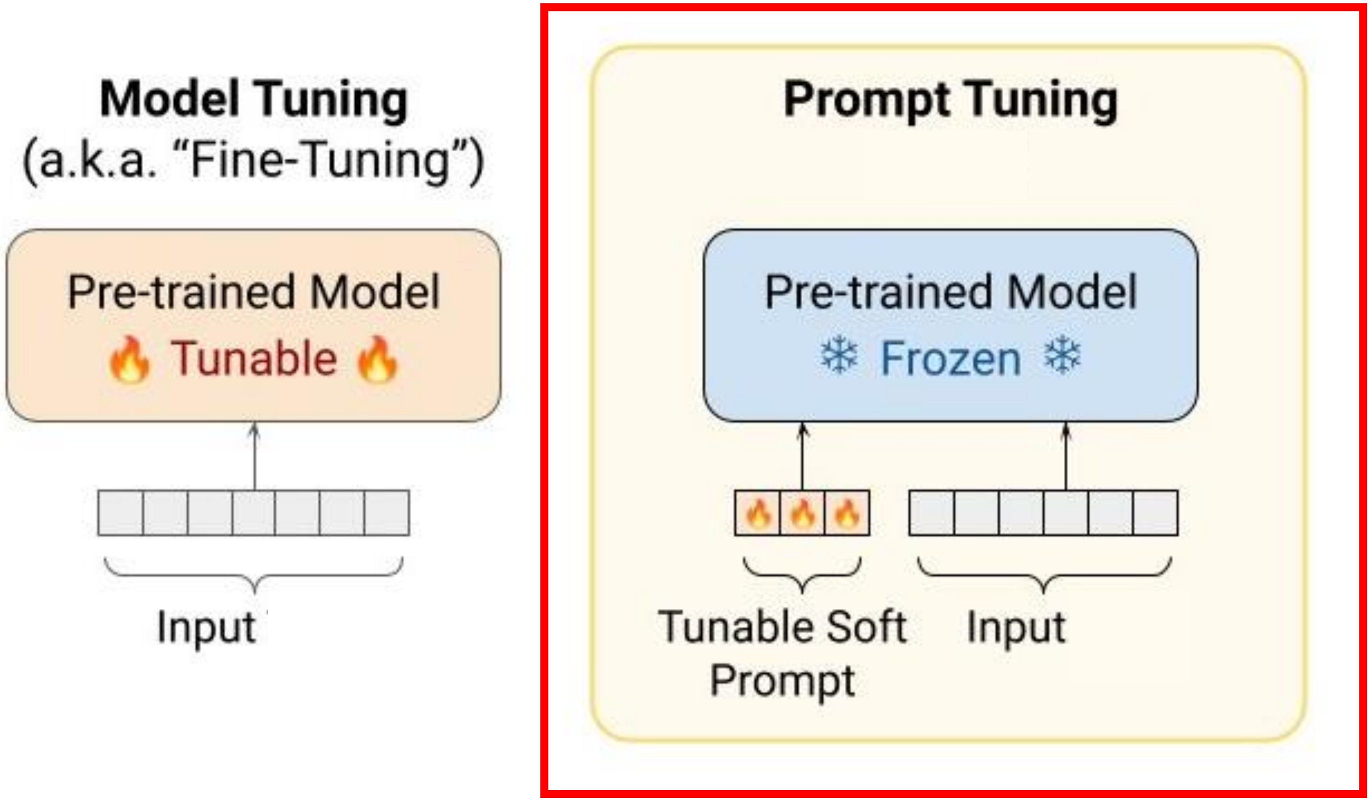
$$\mathcal{L}_{\text{RandAT}} = \mathbb{E}_{i \in \mathcal{V}_L} \left[\mathbb{E}_{\mathbf{h}'_i \in \hat{\mathcal{S}}_i} \left[\ell(d \circ g(\mathbf{h}'_i), y_i) \right] \right],$$

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

Adaptation

Downstream Adaptation

- Prompt-tuning: keep pre-trained model intact
  - pre-prompt: modify input graph
  - post-prompt: modify output embedding



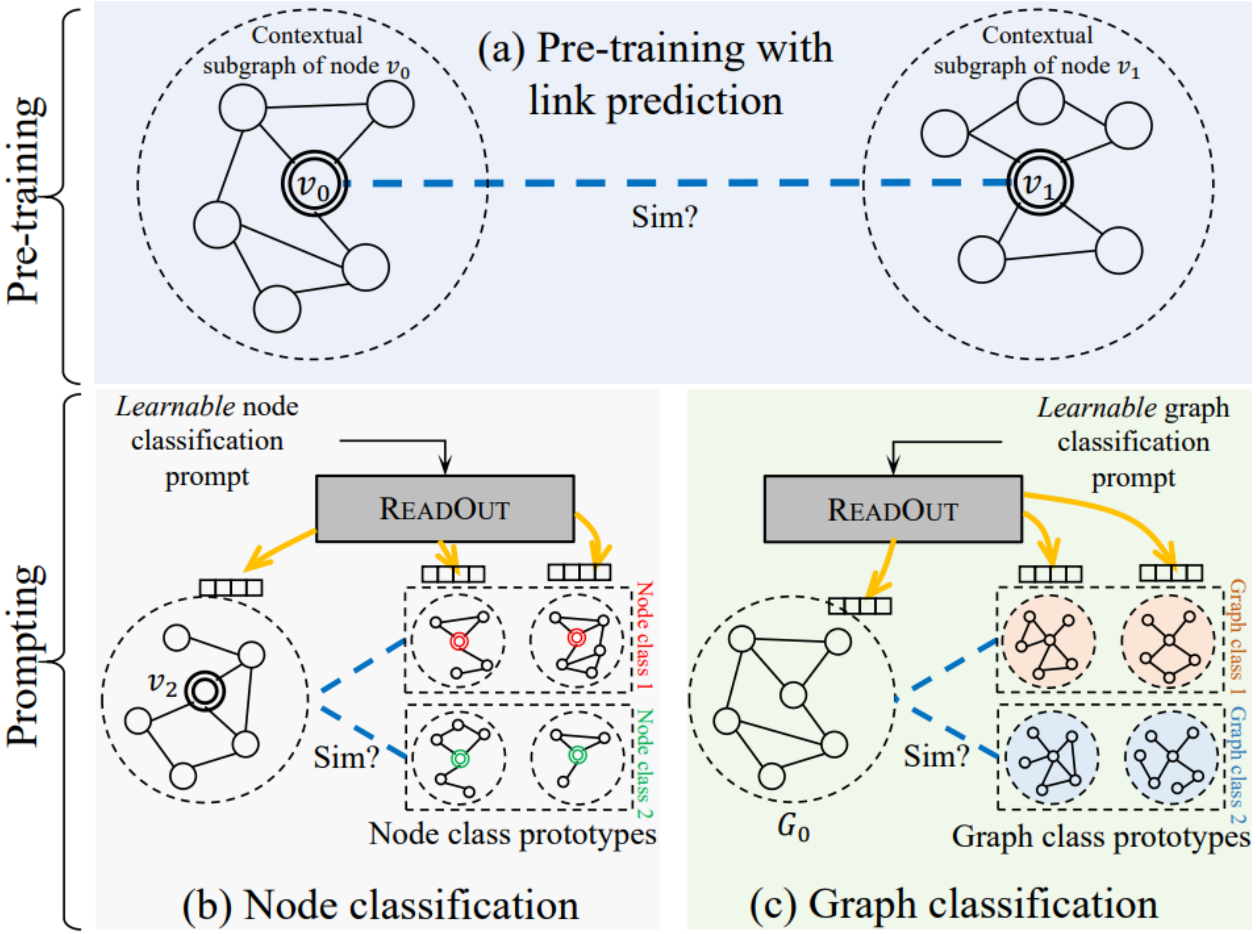
GraphPrompt

Problem:

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

Insights：

- 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

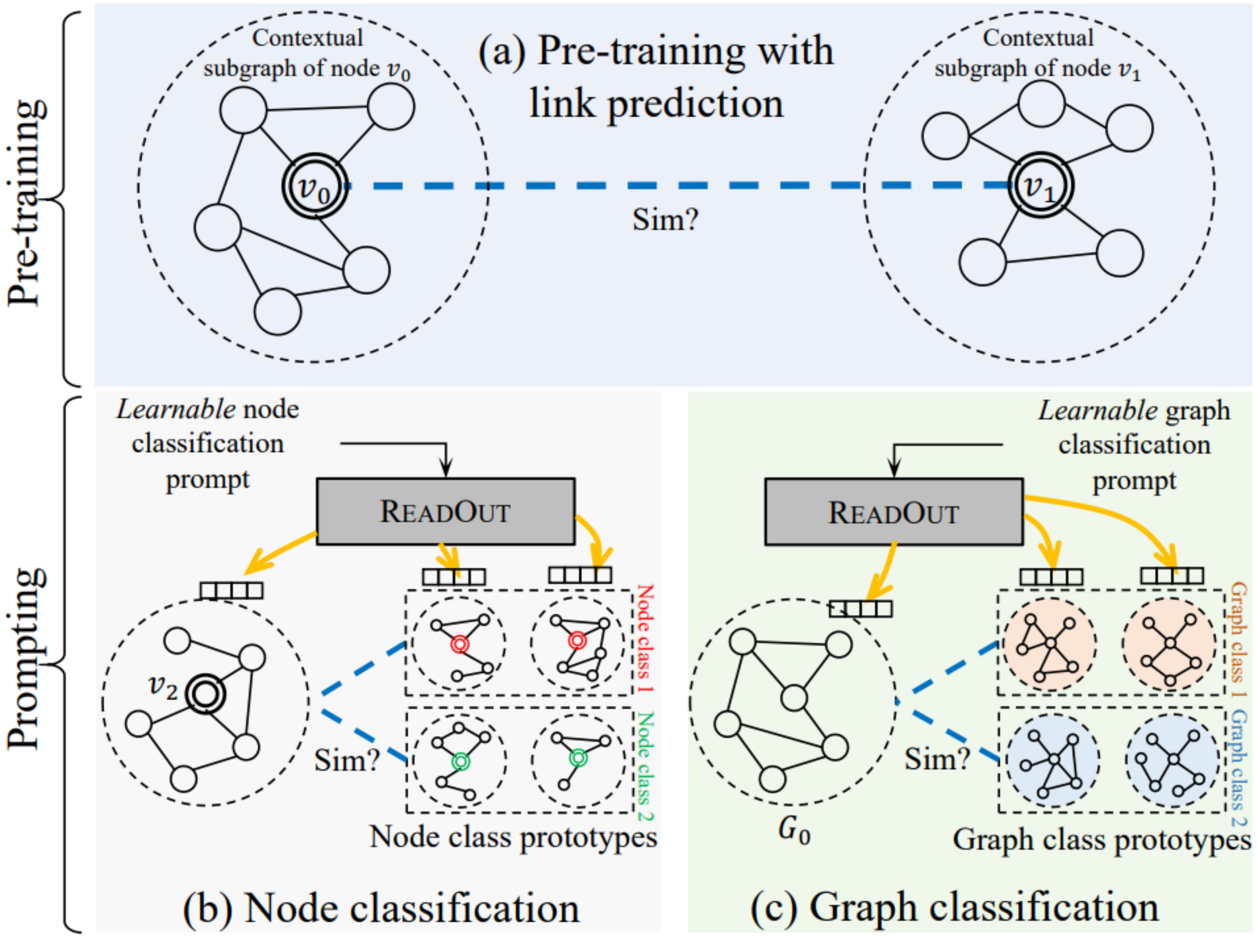
GraphPrompt

Prompt design:

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

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

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



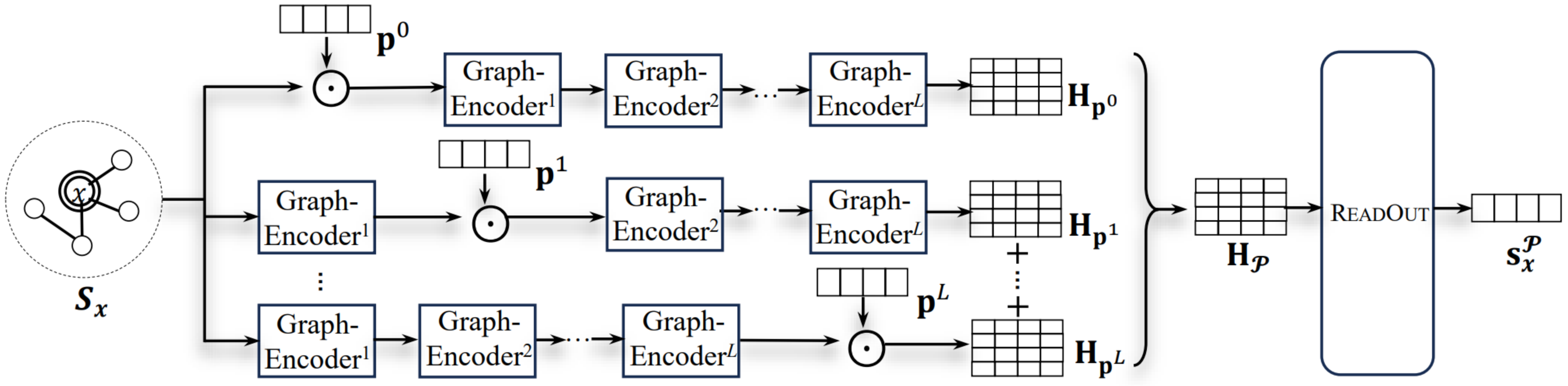
$\mathbf{s}_{t,x}$: (sub)graph embedding of x for a task t
$\mathbf{h}_v$: node v's embedding vector
$\mathbf{p}_t$ or $\mathbf{P}_t$: learnable prompt vector or matrix for task t

Generalized Graph Prompt

Support more pre-training tasks beyond link prediction：

- DGI, InfoGraph, GraphCL, GCC, …

Layer-wise prompts



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.

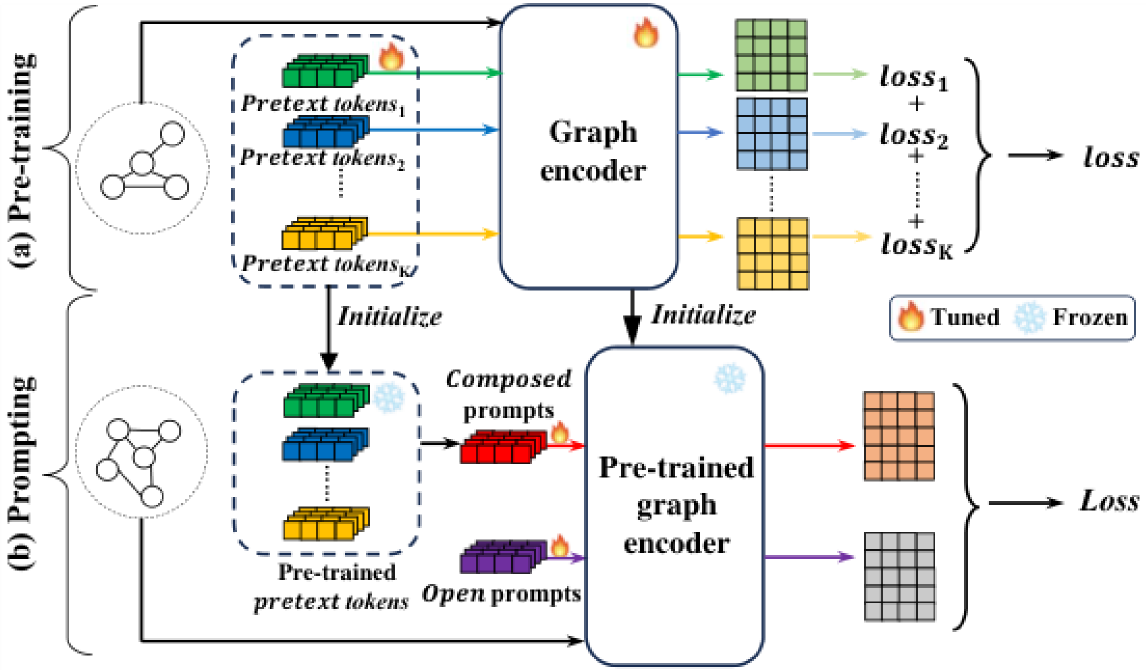
MultiGPrompt

Problem:

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

Challenges:

- 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

Yu, X., Zhou, C., Fang, Y., & Zhang, X. MultiGPrompt for Multi-Task Pre-Training and Prompting on Graphs. WWW2024

MultiGPrompt

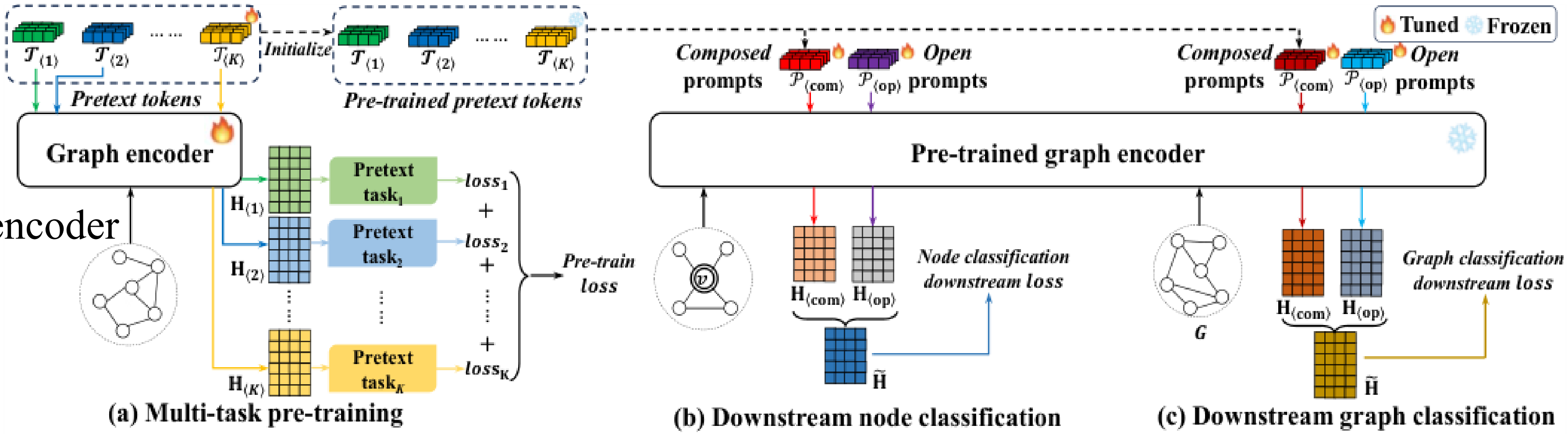
Multi-task pre-training

Pretext tokens

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

Add token to each layer of graph encoder

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



Graph encoder output embedding

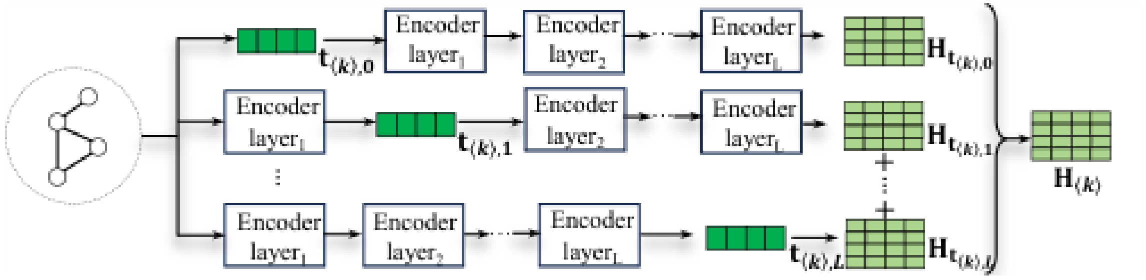
$$\mathbf{H}_{\mathbf{t}} = \textsc{GraphEncoder}_{\mathbf{t}}(\mathbf{X}, \mathbf{A}; \Theta)$$

Overall embedding

$$\mathbf{H}_{\langle k \rangle} = \sum_{l=0}^{L} \alpha_l \mathbf{H}_{\mathbf{t}_{\langle k \rangle,l}}$$

Pre-Training Objective

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



Prompt tuning

Composed prompt

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

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

Open prompt

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

Add prompt to each layer of graph encoder

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

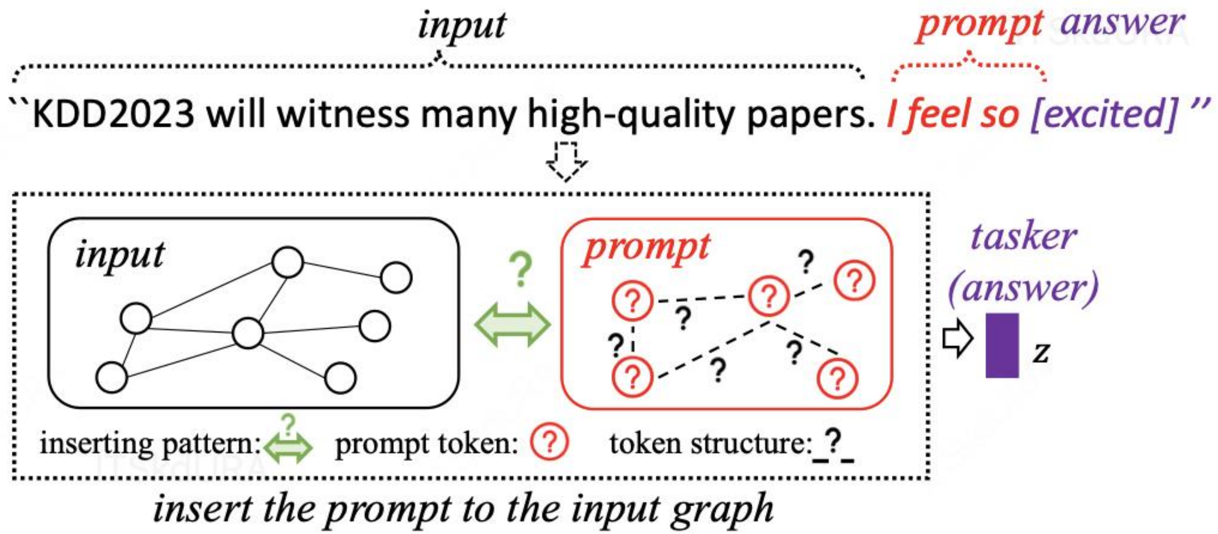
Aggregate dual prompt

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

All in One

Challenges：

- 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.



Sun, X., Cheng, H., Li, J., Liu, B., & Guan, J. All in one: Multi-task prompting for graph neural networks. KDD2023

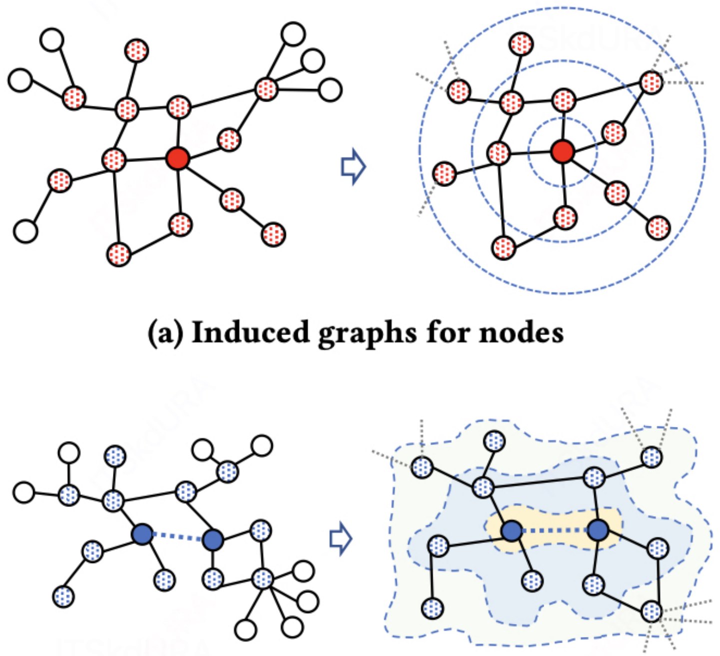
All in One

Reformulate Downstream Tasks：

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

Prompt Graph Design：

- 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

HGPrompt: Extending to heterogeneous graphs

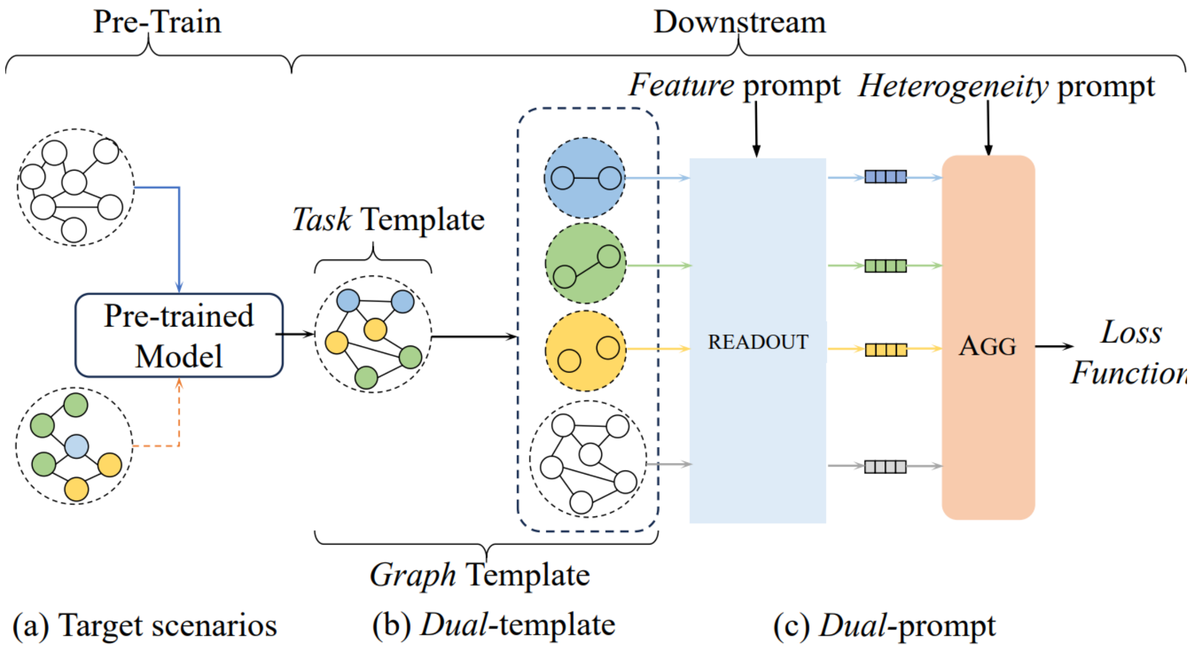
Problem:

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

Insights：

- Dual-template:
  Task + Graph template

- Dual-prompt:
  Feature + Heterogeneity prompt



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

GPF

Challenges：

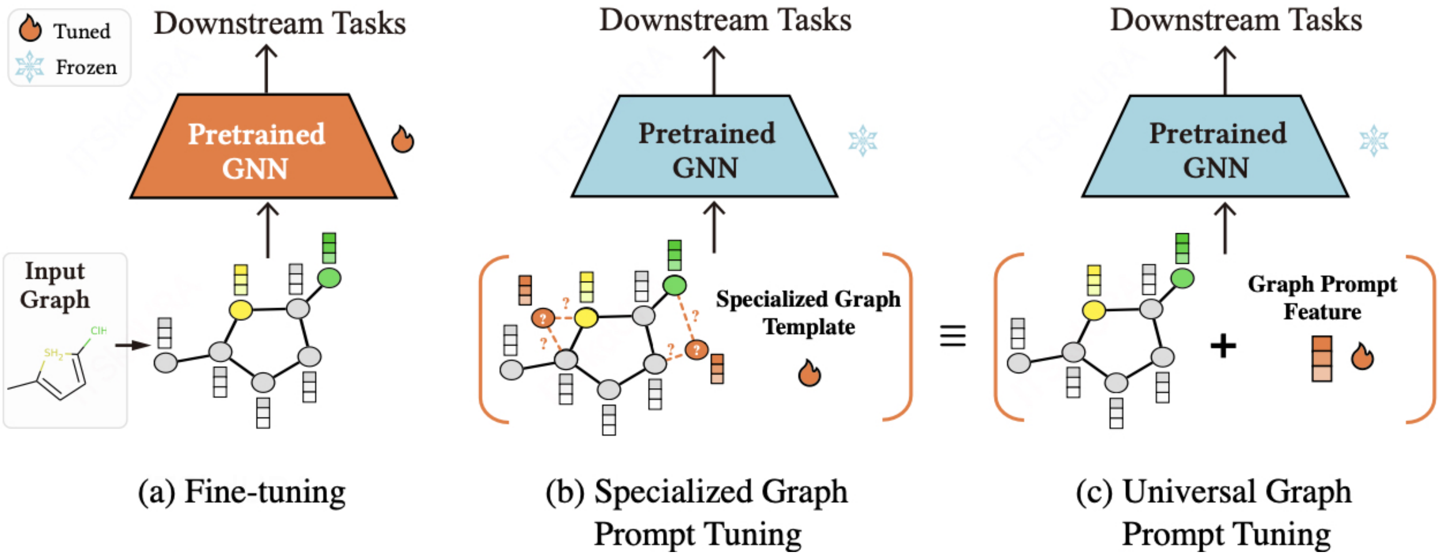
- Diverse pre-training strategies employed on graphs make it difficult to design suitable prompting functions.
- Existing prompt-based tuning methods for GNN models are predominantly designed based on intuition, lacking theoretical guarantees for their effectiveness.

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

GPF

Method：

- 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.

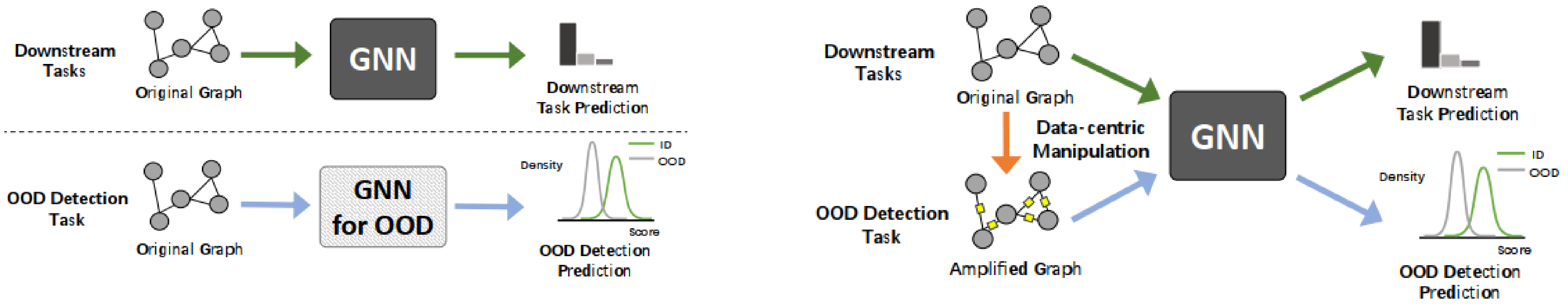


AAGOD

Motivation

- 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?



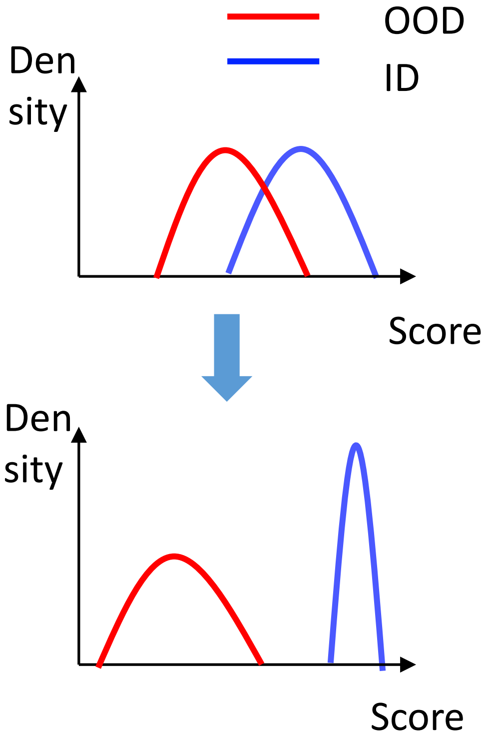
(1) Traditional works

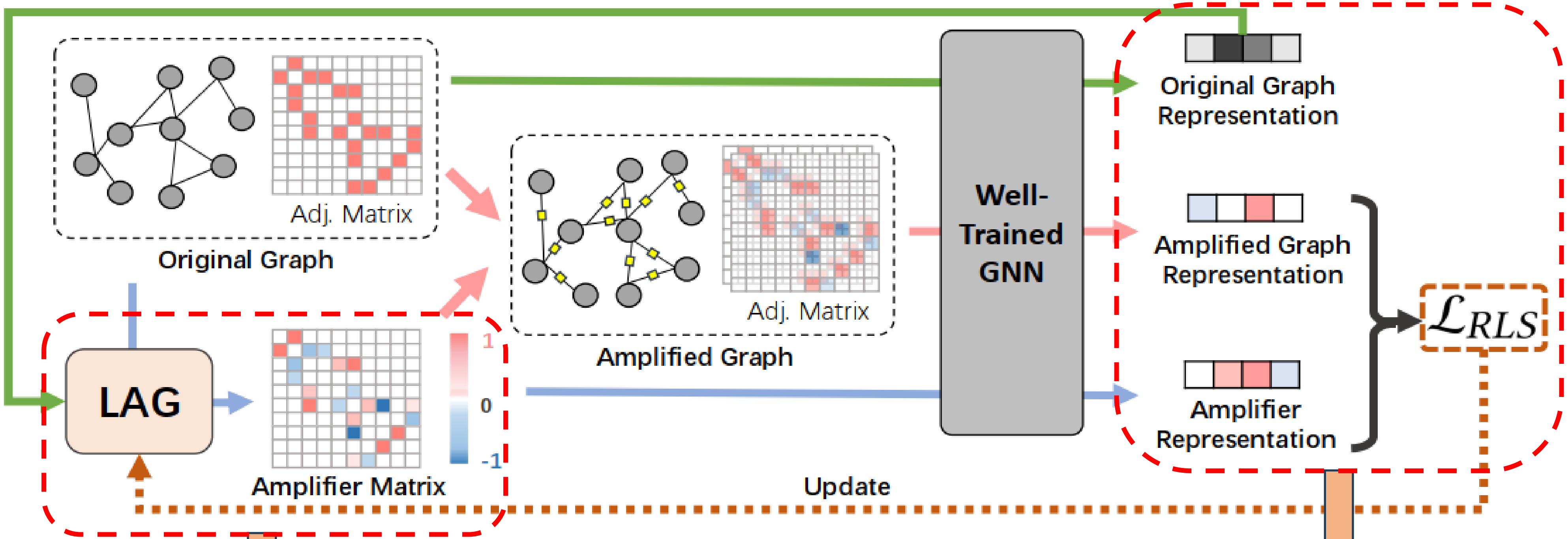
(2) Our proposed framework

Guo, Y., Yang, C., Chen, Y., Liu, J., Shi, C., & Du, J. A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability. KDD2023

AAGOD

We modify edge weights as prompts to highlight the latent pattern of ID graphs, and thus enlarge the score gap between OOD and ID graphs.





LAG adaptively generates graph-specific amplifiers by converting node representations into edge weights.

RLS encourages high scores for amplified ID graphs and expects low scores when only seeing the amplifiers.

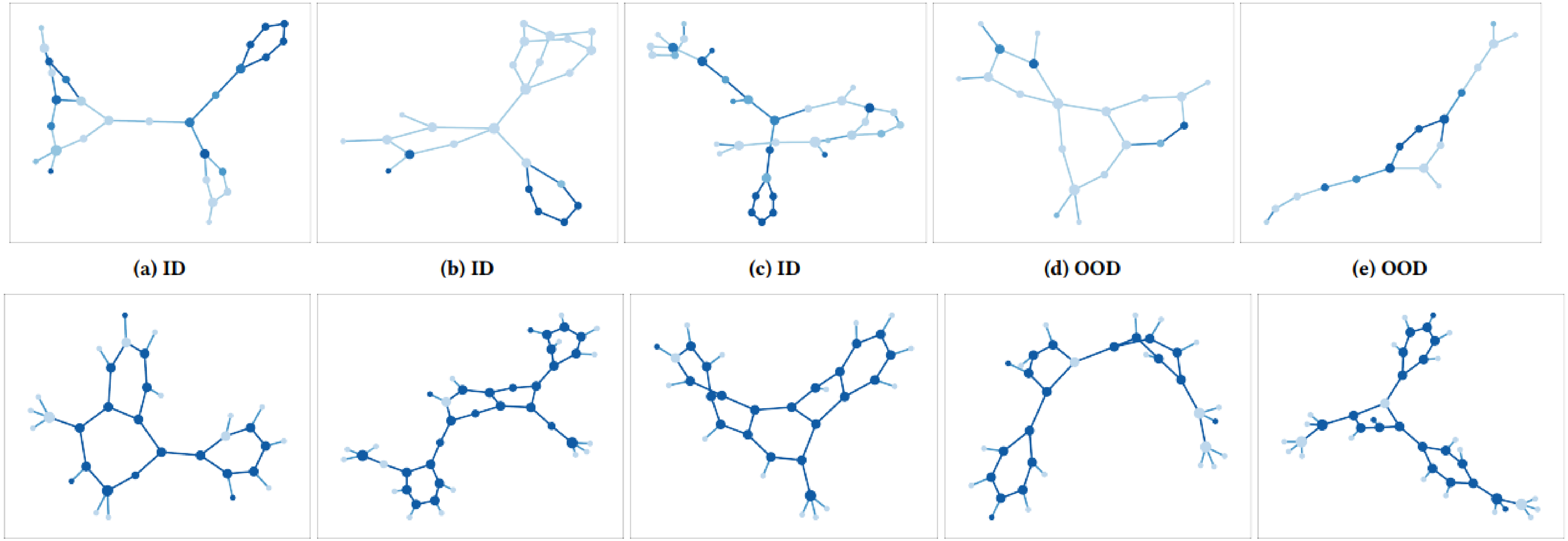
AAGOD

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 $\uparrow$	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 $\uparrow$	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 $\downarrow$	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 $\uparrow$	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 $\uparrow$	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 $\downarrow$	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 $\uparrow$	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 $\uparrow$	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 $\downarrow$	47.50	**15.00**	-68.42%	92.50	**80.00**	-13.51%	37.50	**12.50**	-66.67%	**97.50**	**97.50**	0.00%
TOX21	SIDER	AUC $\uparrow$	68.04	**71.27**	+4.75%	53.44	**58.25**	+9.00%	53.46	**69.39**	+29.80%	53.64	**55.67**	+3.78%
		AUPR $\uparrow$	69.28	**73.52**	+6.12%	56.81	**59.58**	+4.88%	56.02	**71.01**	+26.76%	**56.02**	**56.02**	0.00%
		FPR95 $\downarrow$	90.42	**89.53**	-0.98%	94.25	**92.72**	-1.62%	95.66	**90.55**	-5.34%	95.66	**89.66**	-6.27%
BBBP	BACE	AUC $\uparrow$	77.07	**80.64**	+4.63%	46.74	**50.53**	+8.11%	75.48	**78.54**	+4.05%	43.96	**51.28**	+16.65%
		AUPR $\uparrow$	68.41	**72.60**	+6.12%	45.35	**46.49**	+2.51%	69.32	**74.06**	+6.84%	44.77	**48.32**	+7.93%
		FPR95 $\downarrow$	71.92	**60.59**	-15.75%	92.12	**86.70**	-5.88%	76.85	**69.46**	-9.62%	94.09	**92.61**	-1.57%

AAGOD

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

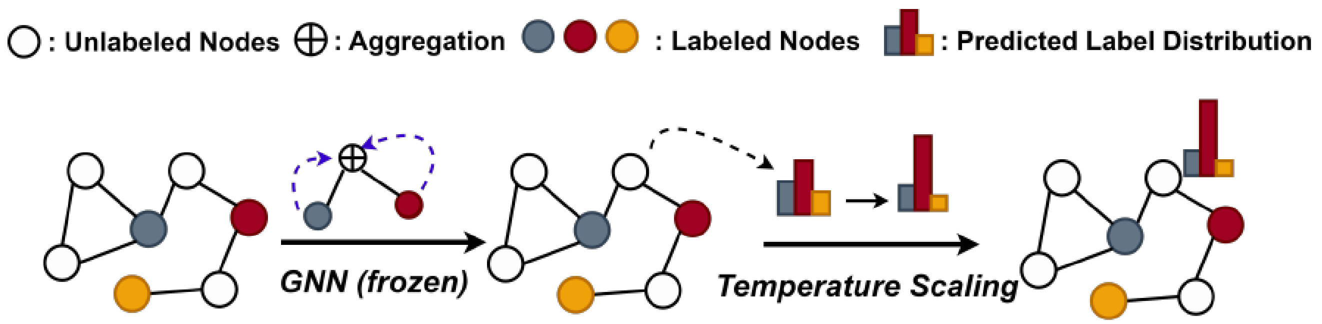


(a) ID (b) ID (c) ID (d) OOD (e) OOD

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

DCGC

- 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

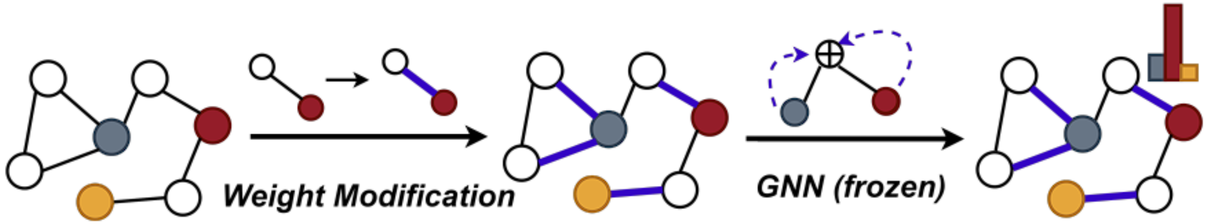
- Through evaluating the expected calibration error (ECE) on Cora and Photo datasets with five different GNNs, we find that the ECEs on Cora (10.25%-18.02%) are always larger than those on Photo (4.38%-8.27%), indicating that **the calibration performance depends more on the datasets instead of GNN model**.

Yang, C., Yang, C., Shi, C., Li, Y., Zhang, Z., & Zhou, J. Calibrating Graph Neural Networks from a Data-centric Perspective. WWW2024

DCGC

- Inspired by this phenomenon, we innovatively propose to calibrate GNNs from a data-centric perspective:

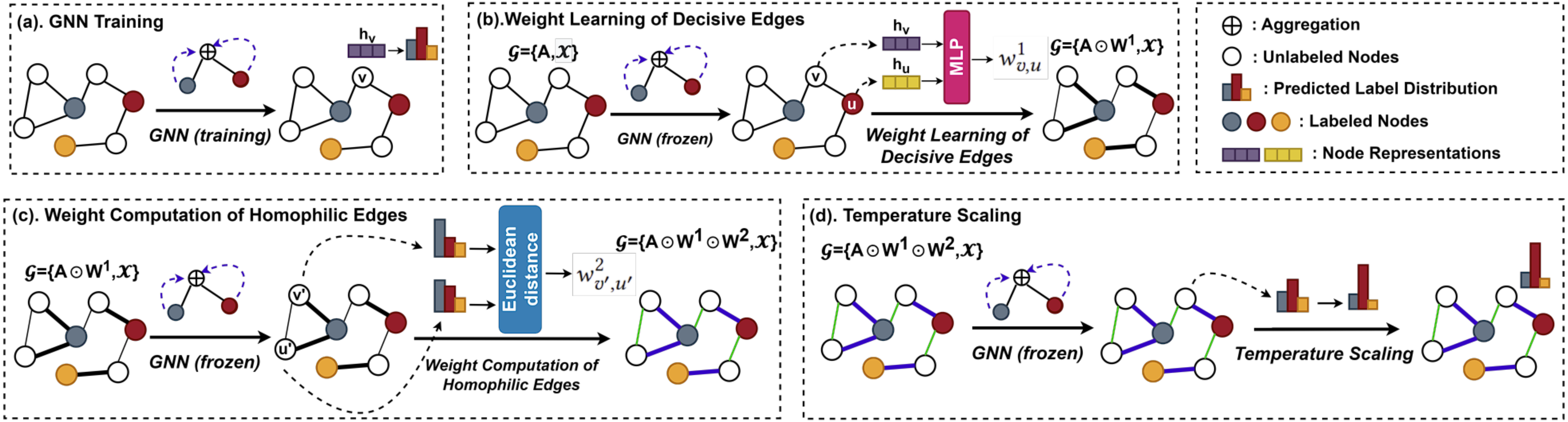
Can we modify the graph data instead for better calibration performance without losing accuracy?



(b) Data-centric calibration

DCGC

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



Summary

GNN-based models compares to foundation models with LLMs

- Advantage：
  - small parameter size, resulting in low-cost training
  - possess essential properties like permutation invariance
  - exhibit strong performance in scenarios without textual attributes
- Disadvantage：
  - limited capacity to harness extensive knowledge and can struggle to manifest emergent abilities
  - underutilize the information stored in textual data

Thanks
Q&A

Towards Graph Foundation Models

WWW 2024 Tutorial

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

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Towards Graph Foundation Models
Part III: LLM & GNN+LLM Models

Presented by **Yuan Fang**, Singapore Management University

yfang@smu.edu.sg | www.yfang.site

Prepared by **Yuxia Wu**, Singapore Management University

Outline

- **LLM based Models**
  - Backbone Architecutures
  - Pre-training
  - Adaptation
- GNN+LLM based Models
  - Backbone Architecutures
  - Pre-training
  - Adaptation
- Summary and outlook

LLM-based Models

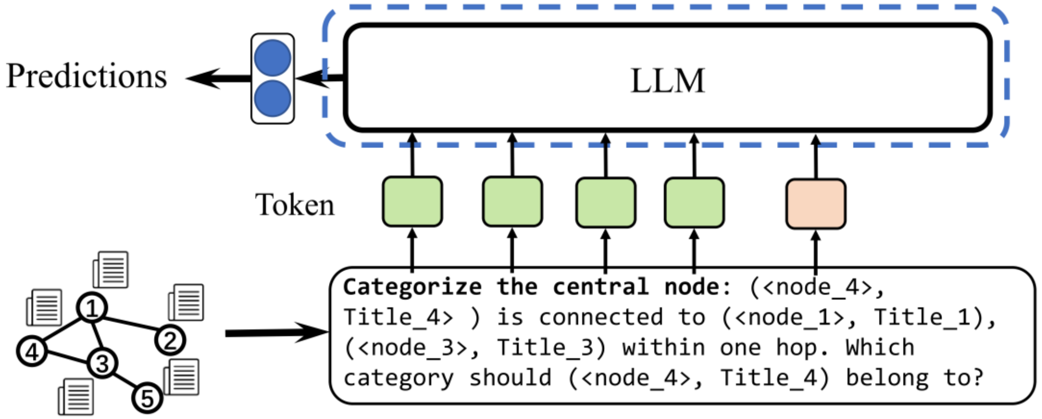
- ❑ **Backbone Architecutures**
- ❑ Pre-training
- ❑ Adaptation

Model	Backbone Architecture			Pre-training	Adaptation
InstructGLM[157]	Graph-to-token	+	Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text	+	GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text	+	GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text	+	BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

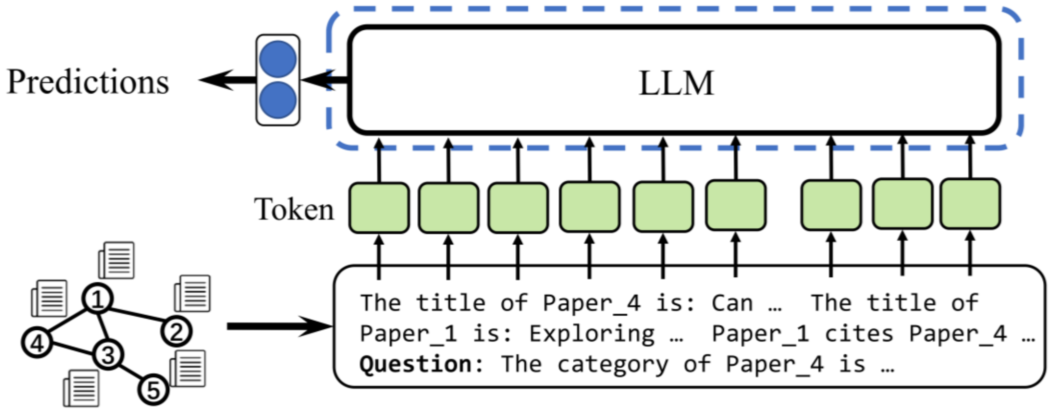
Table 3. Details of approaches involved as LLM based models

Backbone Architectures

- Graph-to-Token
  - Tokenize graph information to align it with LLM
- Graph-to-text
  - Describe graph information using natural language



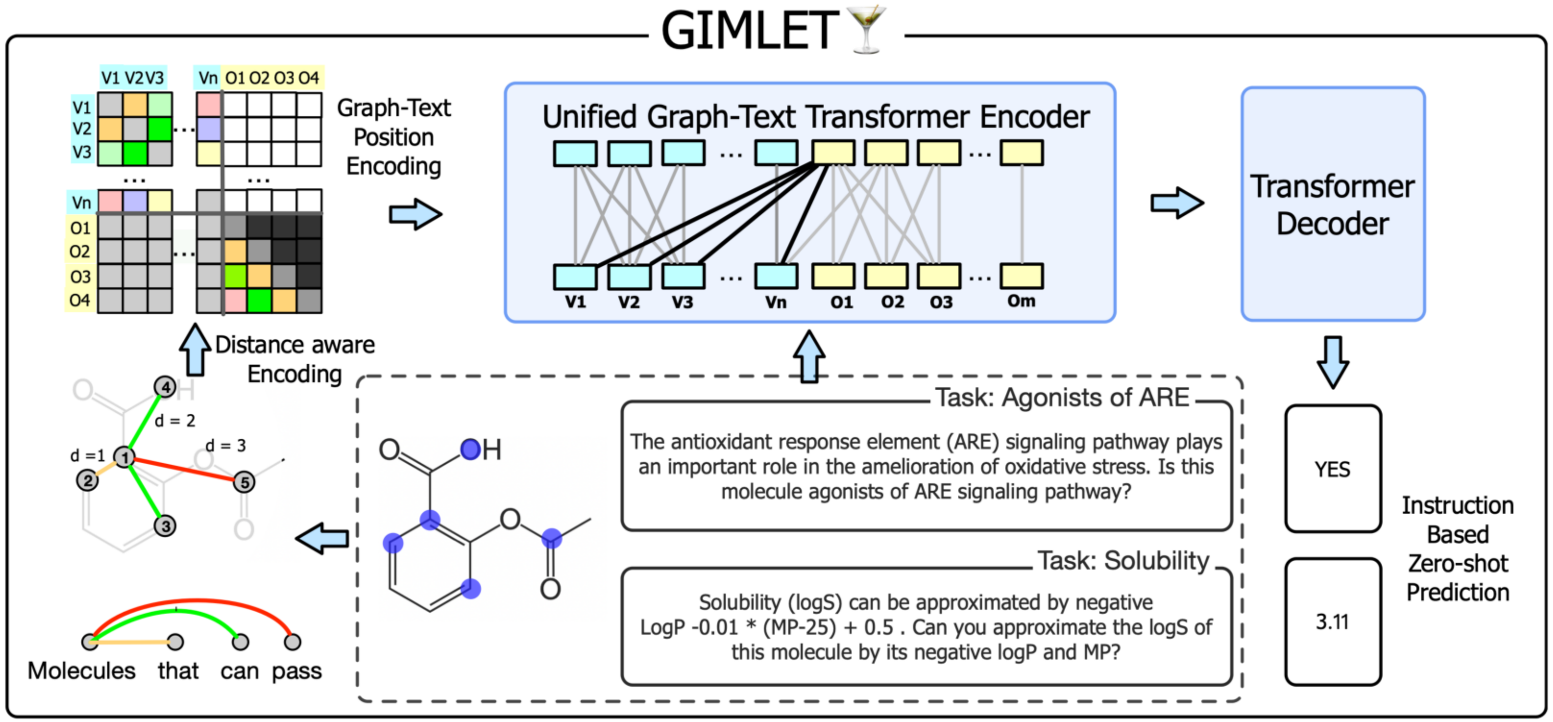
(a) Graph-to-token.



(b) Graph-to-text.

Graph-to-Token: GIMLET

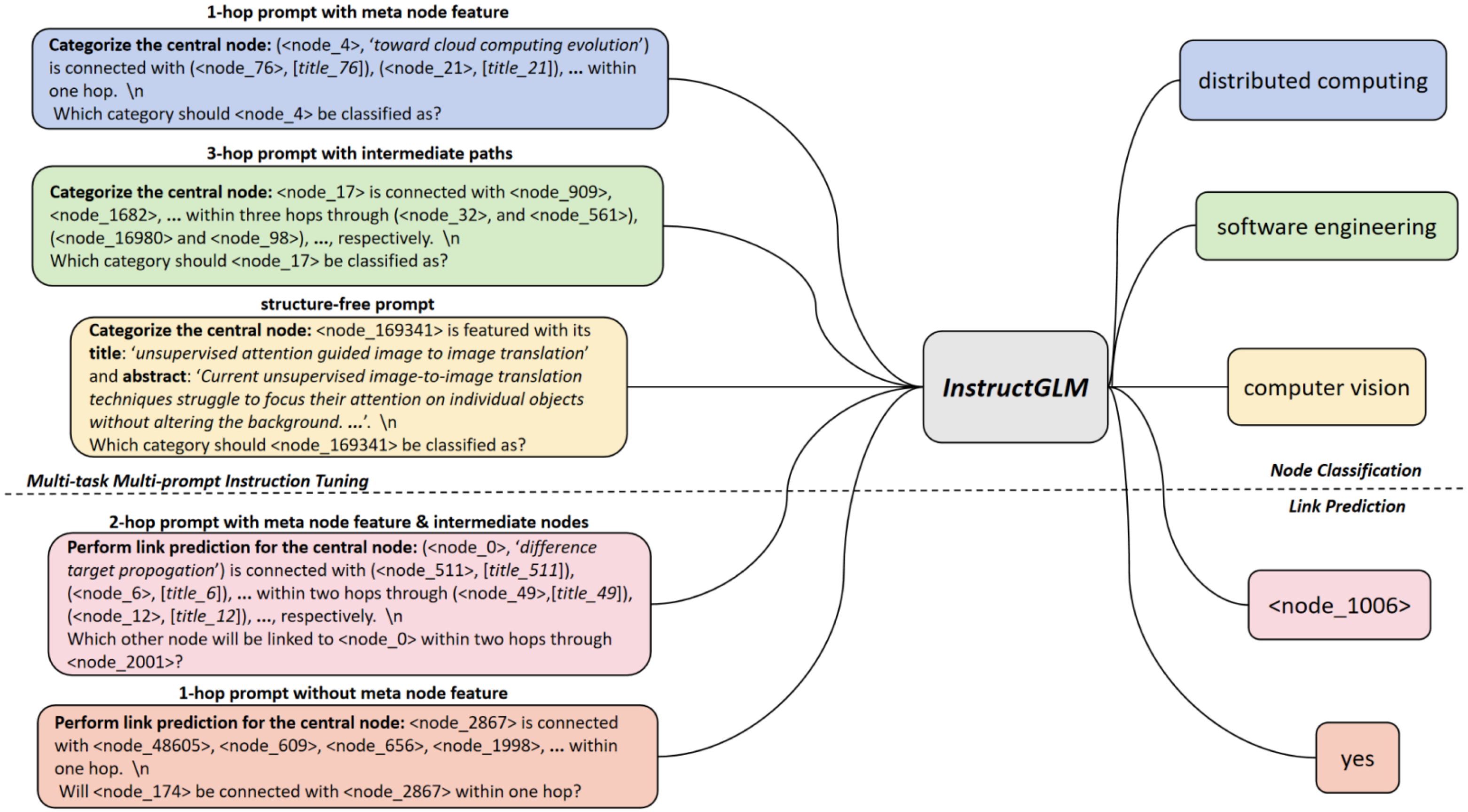
- ❑ Integrating graph data with textual data
- ❑ Encoding the graph's structural information



Zhao, et al. "GIMLET: A unified graph-text model for instruction-based molecule zero-shot learning." *NeurIPS'23.*

Graph-to-Token: InstructGLM

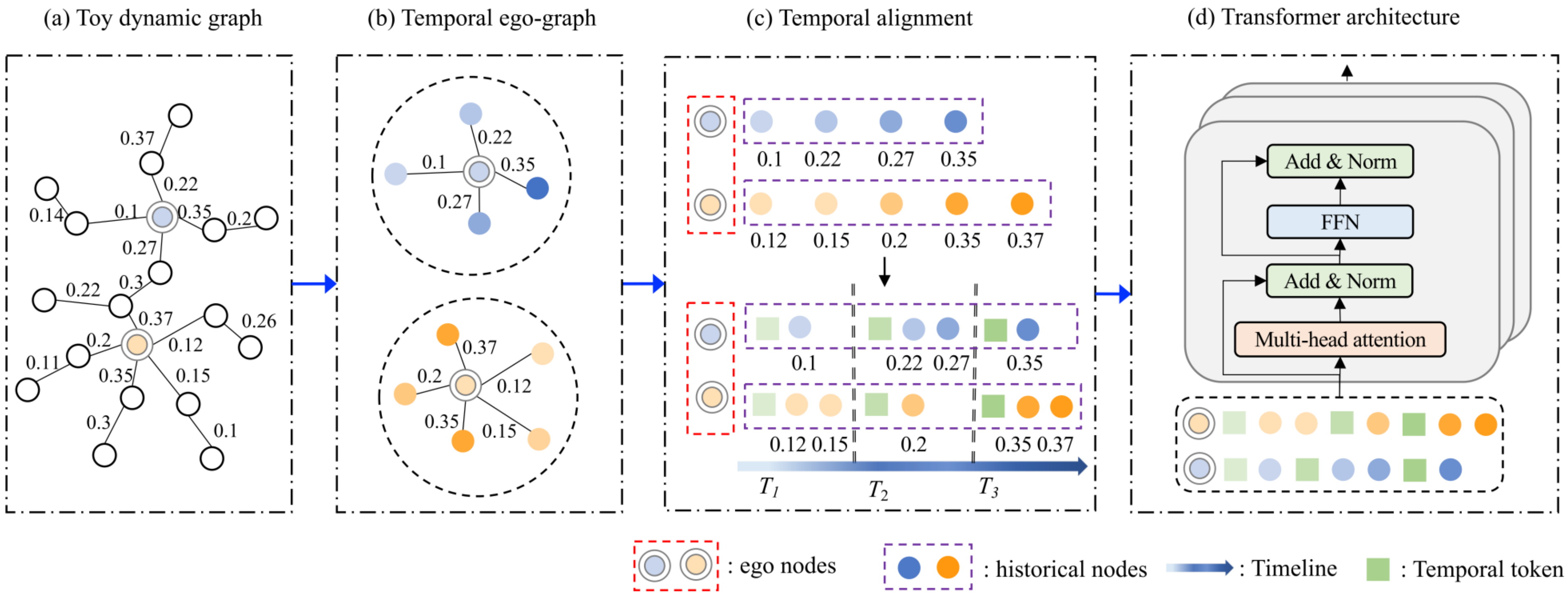
- Expand the vocabulary of the LLM by graph node features



Ye, et al. "Language is all a graph needs." *EACL 2024*.

Graph-to-Token: SimpleDyG

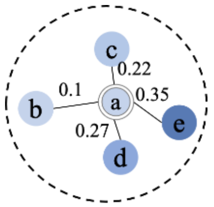
- Transformer-based approach for dynamic graphs
- Map a dynamic graph into a set of sequences



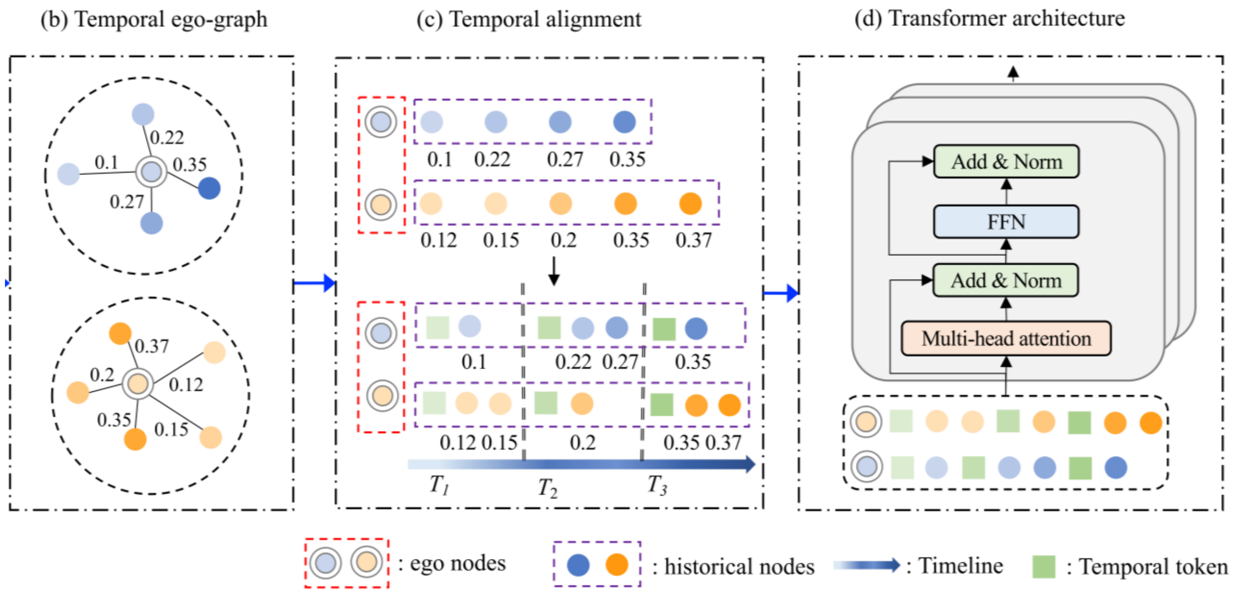
Wu, et al. "On the Feasibility of Simple Transformer for Dynamic Graph Modeling." *WWW'24*.

Graph-to-Token: SimpleDyG

- Temporal ego-graph



$$w_i = \langle b, c, d, e \rangle$$



- Temporal alignment:
  - Segment the time domain:

$$S_i^1 = \langle b \rangle \quad S_i^2 = \langle c, d \rangle \quad S_i^3 = \langle e \rangle$$

  - Sequence for Transformer:

$$x_i' = \langle |hist| \rangle, a, \langle |time1| \rangle, b, \langle |time2| \rangle, c, d, \langle |time3| \rangle, e, \langle |endofhist| \rangle$$

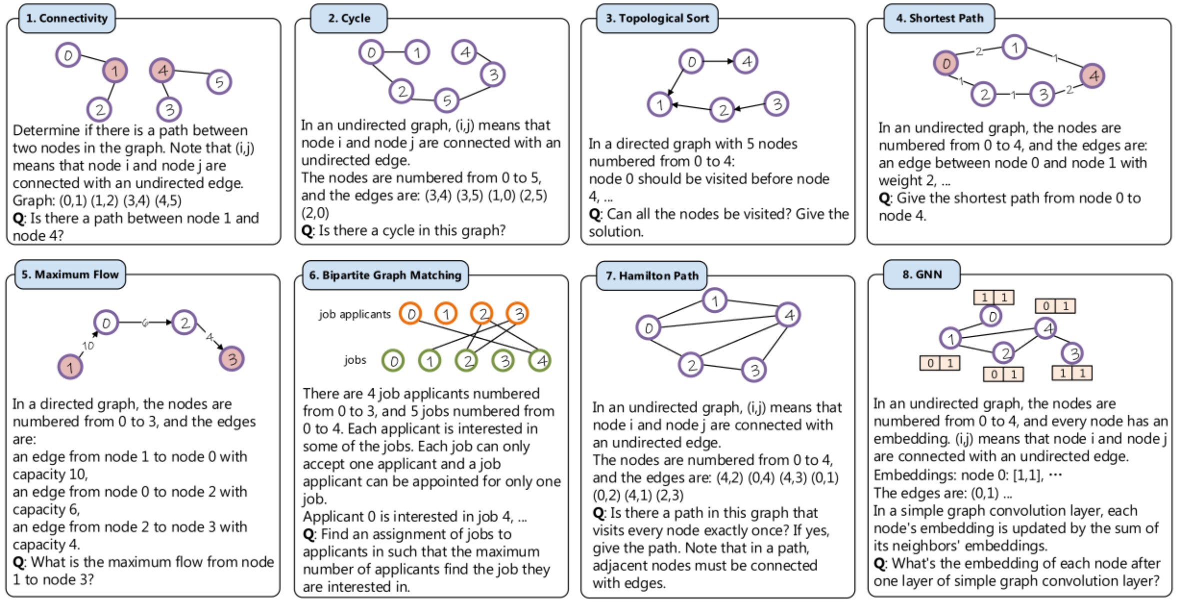
$$y_i' = \langle |pred| \rangle \langle |time4| \rangle S_i^4 \langle |endofpred| \rangle$$

Wu, et al. "On the Feasibility of Simple Transformer for Dynamic Graph Modeling." *WWW'24*.

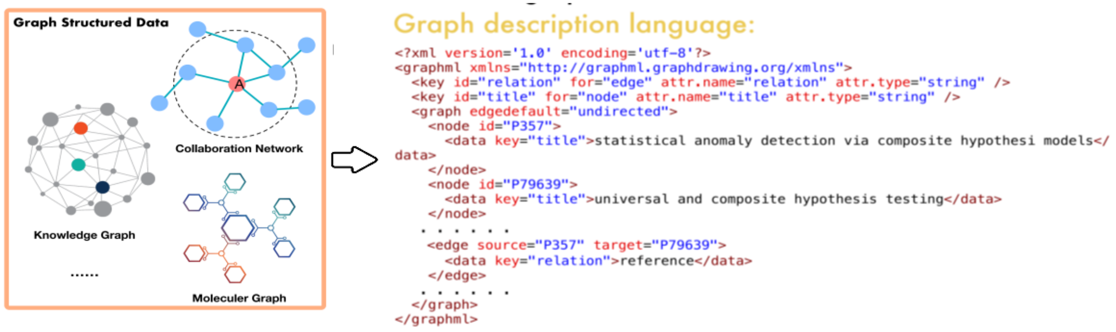
Graph-to-text

- ❑ Describe graph information for variours graphs and tasks

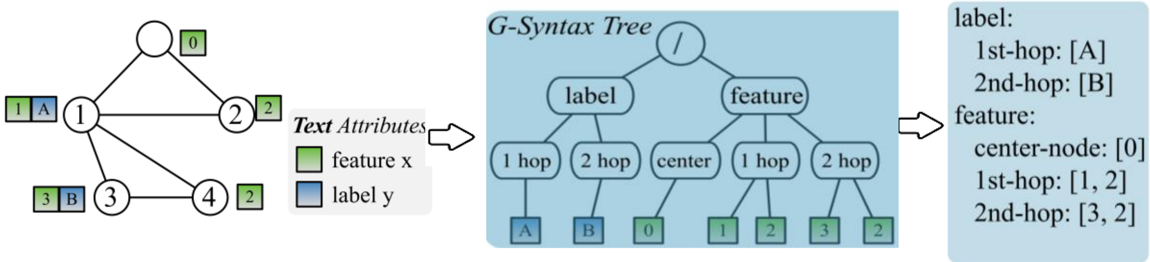
➢Node/edge list, graph properties



➢Graph description language



➢Graph-Syntax Tree



Wang, et al. "Can language models solve graph problems in natural language?." *NeurIPS'23.*
Guo, et al. "GPT4Graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking." *CoRR'23.*
Zhao, et al. "GraphText: Graph reasoning in text space." *CoRR'23.*

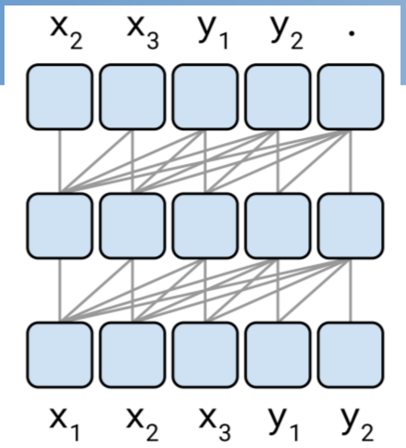
LLM-based Models

- Backbone Architecutures
- **Pre-training**
- Adaptation

Model	Backbone Architecture			Pre-training	Adaptation
InstructGLM[157]	Graph-to-token	+	Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text	+	GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text	+	GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text	+	BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

Table 3. Details of approaches involved as LLM based models

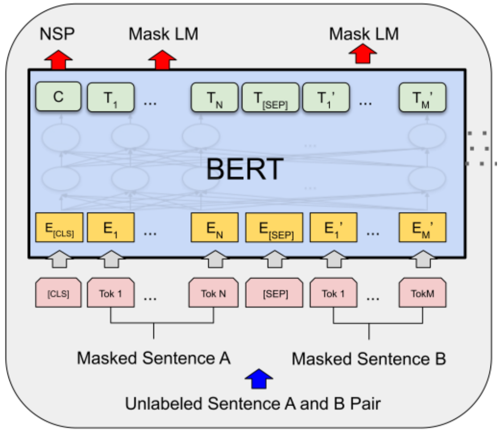
Pre-training



- ❑ Language Modeling (LM)
  - ➢ LLaMA, GPT-3...

- ❑ Masked Language Modeling (MLM)
  - ➢ BERT, T5...
  - ➢ Replace the word with the [MASK] token

    e.g., `my dog is hairy` → `my dog is [MASK]`



Touvron, et al. "Llama: Open and efficient foundation language models." *CoRR'23*.

Ouyang, et al. "Training language models to follow instructions with human feedback." *NeurIPS'22*.

Devlin, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." *CoRR'18*.

Raffel, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *JMLR'20*.

LLM-based Models

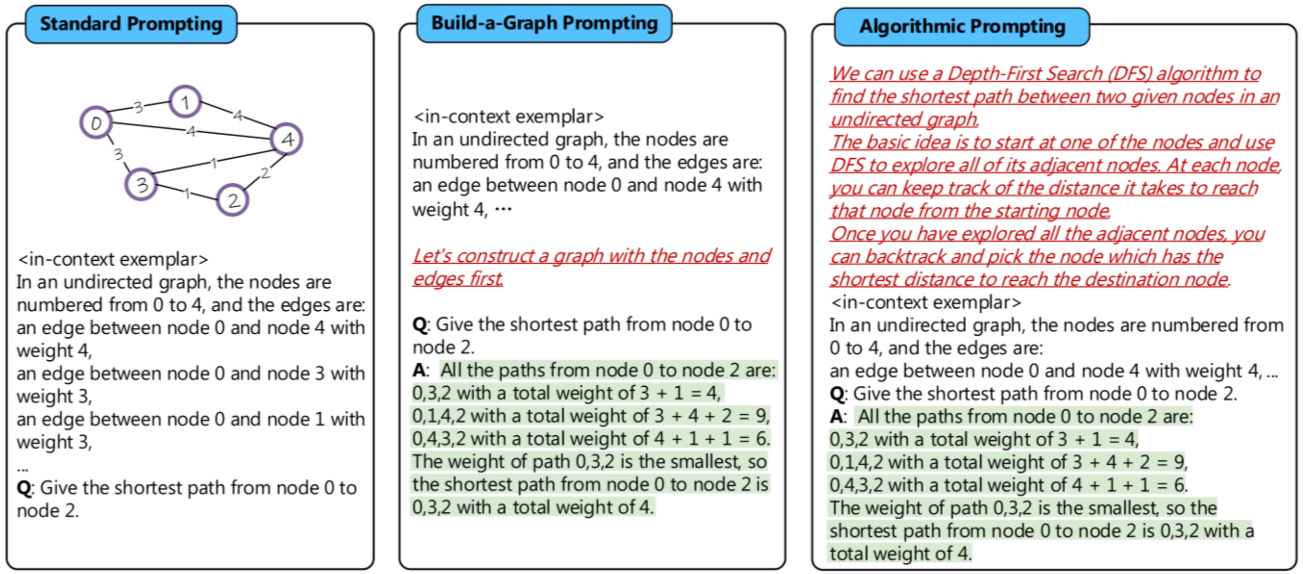
- ❑ Backbone Architecutures
- ❑ Pre-training
- ❑ **Adaptation**

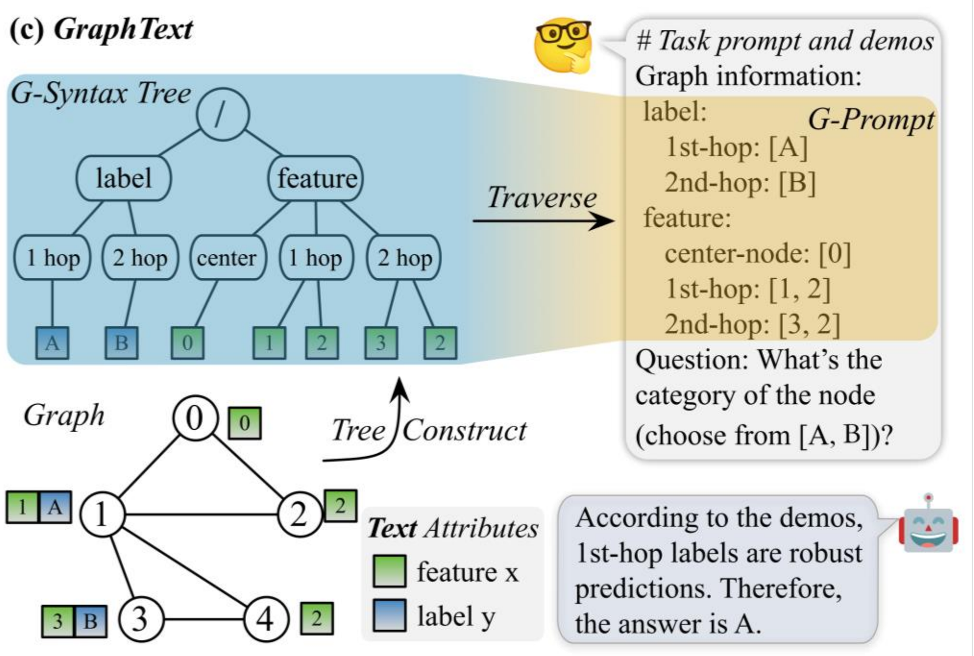
Model	Backbone Architecture			Pre-training	Adaptation
InstructGLM[157]	Graph-to-token	+	Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text	+	GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text	+	GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text	+	GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text	+	BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

Table 3. Details of approaches involved as LLM based models

Adaptation

- ❑ Manual Prompting: Graph information, task descriptions
- ❑ Automatic Prompting: LLMs--> generate the context

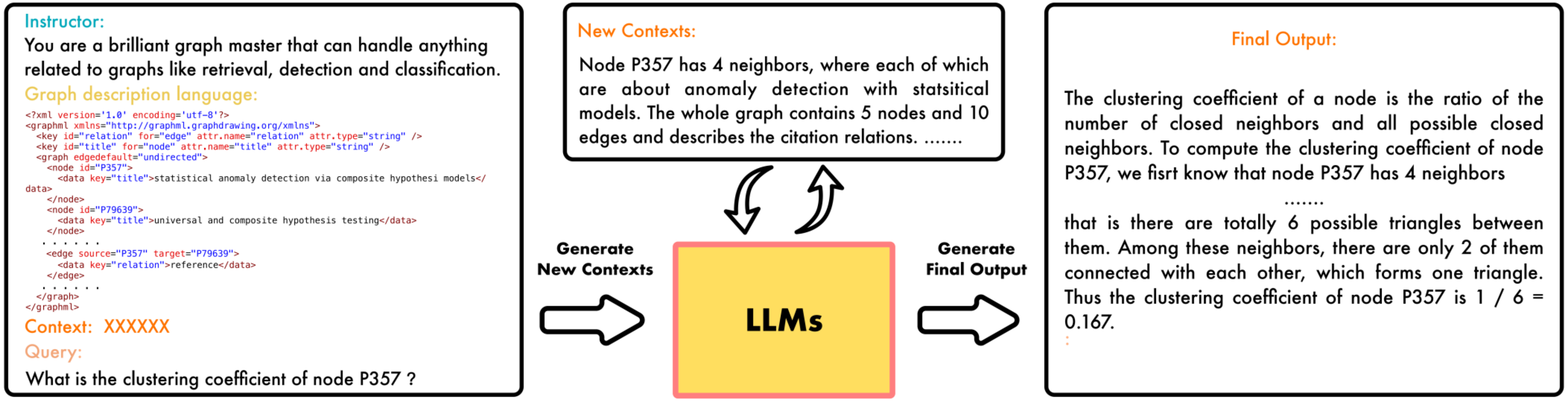




Wang, et al. "Can language models solve graph problems in natural language?." *NeurIPS'23*
Zhao, et al. "GraphText: Graph reasoning in text space." *CoRR'23*

Adaptation

- ❑ Manual Prompting: Graph information, task descriptions
- ❑ Automatic Prompting: LLMs → generate the context
  - ➢ Ask LLM generate graph/neighbor summarization



Guo, et al. "Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking." *CoRR'23*
Chen, et al. "Exploring the potential of large language models (llms) in learning on graphs." *ACM SIGKDD Explorations Newsletter 2024*

Outline

- ❑ LLM based Models
  - ➢ Backbone Architecutures
  - ➢ Pre-training
  - ➢ Adaptation
- ❑ **GNN+LLM based Models**
  - ➢ Backbone Architecutures
  - ➢ Pre-training
  - ➢ Adaptation
- ❑ Summary and outlook

GNN+LLM based Models

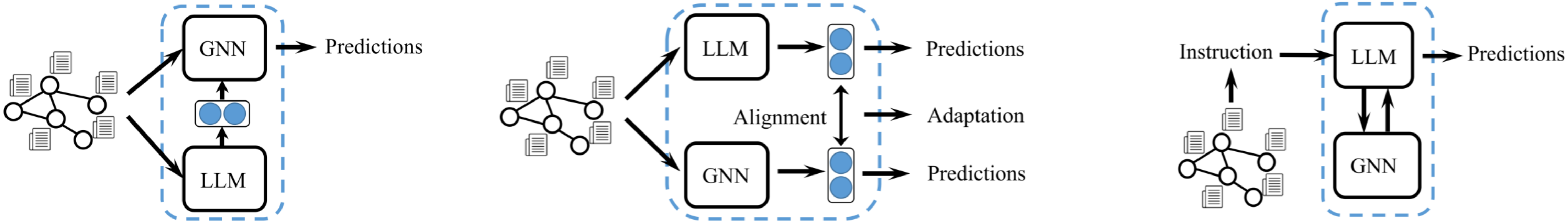
- ❑ **Backbone Architecutures**
- ❑ Pre-training
- ❑ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

Table 4. Details of approaches involved as GNN+LLM based models

Backbone Architectures

- GNN-centric Methods
  - LLMs extract node features from raw data; GNNs make predictions
- Symmetric Methods
  - Align the embeddings of GNN and LLM
- LLM-centric Methods
  - Utilize GNNs to enhance the performance of LLM

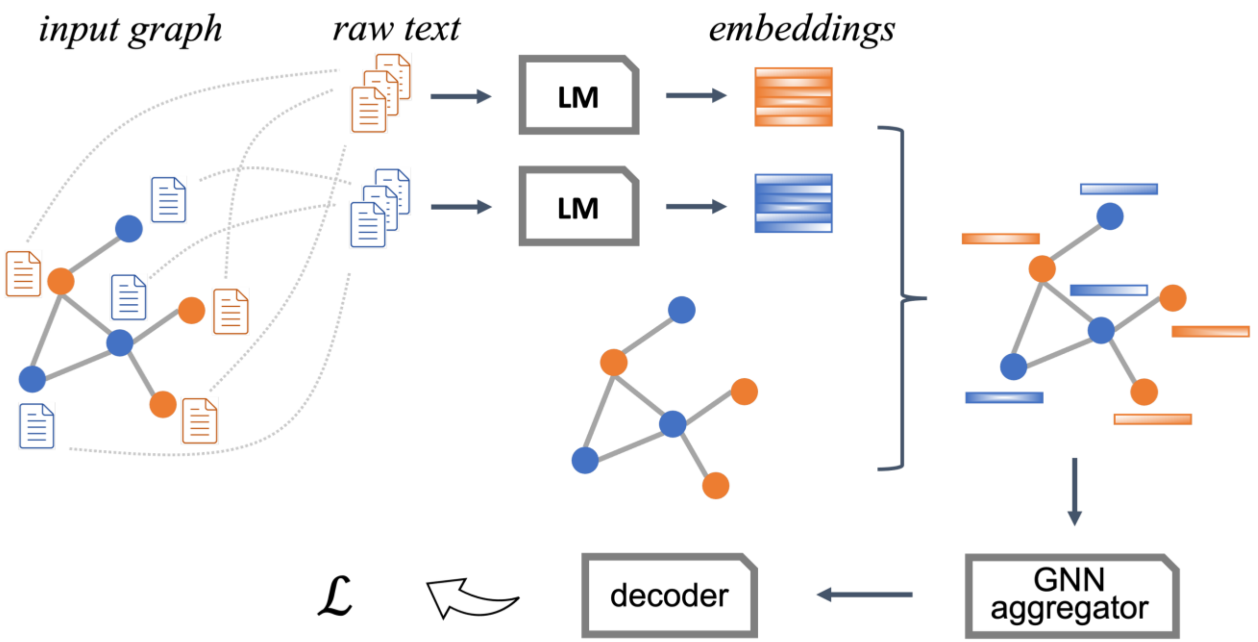


(a) GNN-centric methods. (b) Symmetric methods. (c) LLM-centric methods.

GNN-centric Methods: GaLM

- The backbone model:

  Raw text → LMs → GNN aggregator → decoder



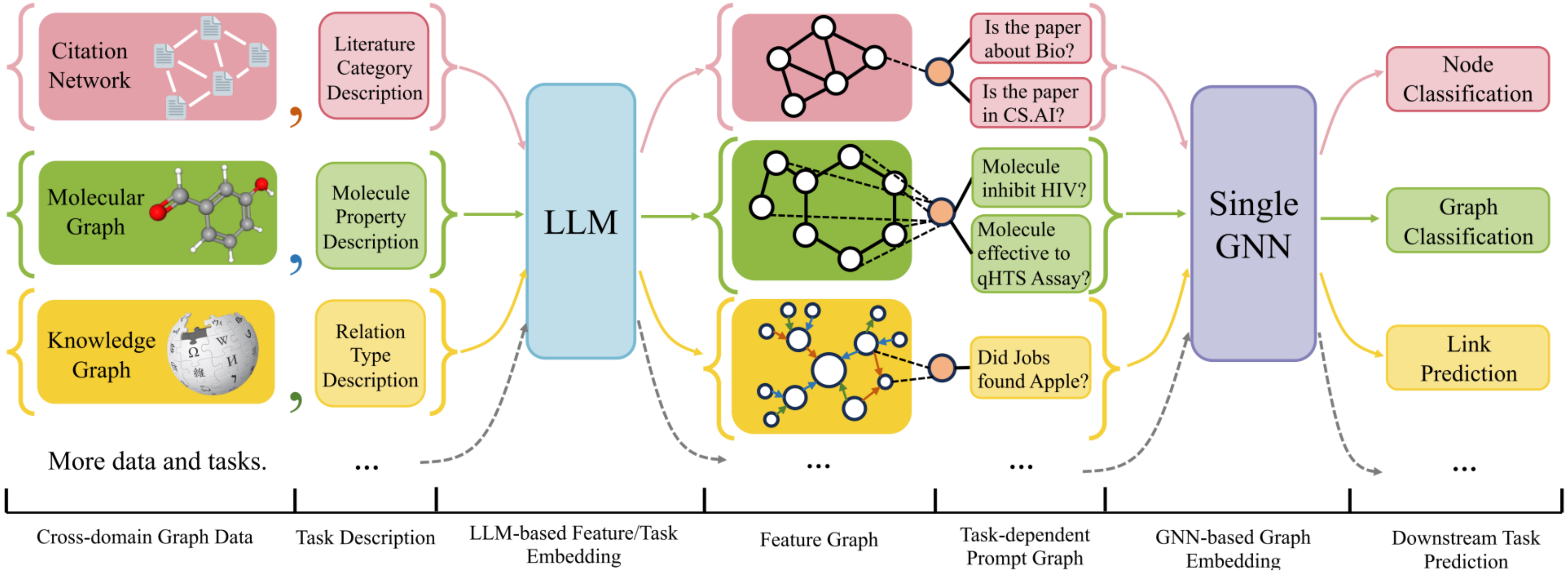
Xie, et al. "Graph-aware language model pre-training on a large graph corpus can help multiple graph applications." *KDD'23*.

GNN-centric Methods: One for all

❑ The backbone model:

Text-attributed graph
Task description

LLMs → Prompted graph → GNN → Downstream tasks

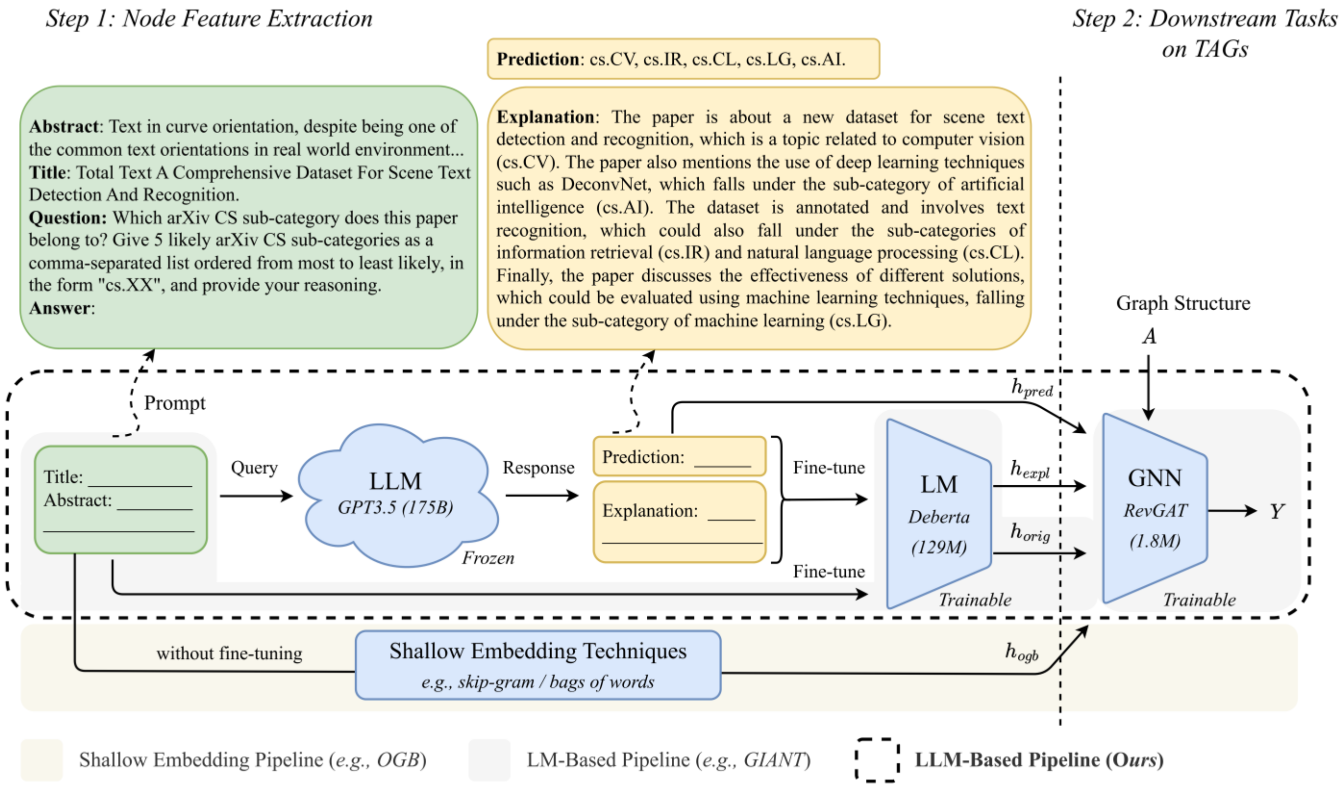


Liu, et al. "One for all: Towards training one graph model for all classification tasks." *ICLR'24*

GNN-centric Methods: TAPE

- The backbone model:

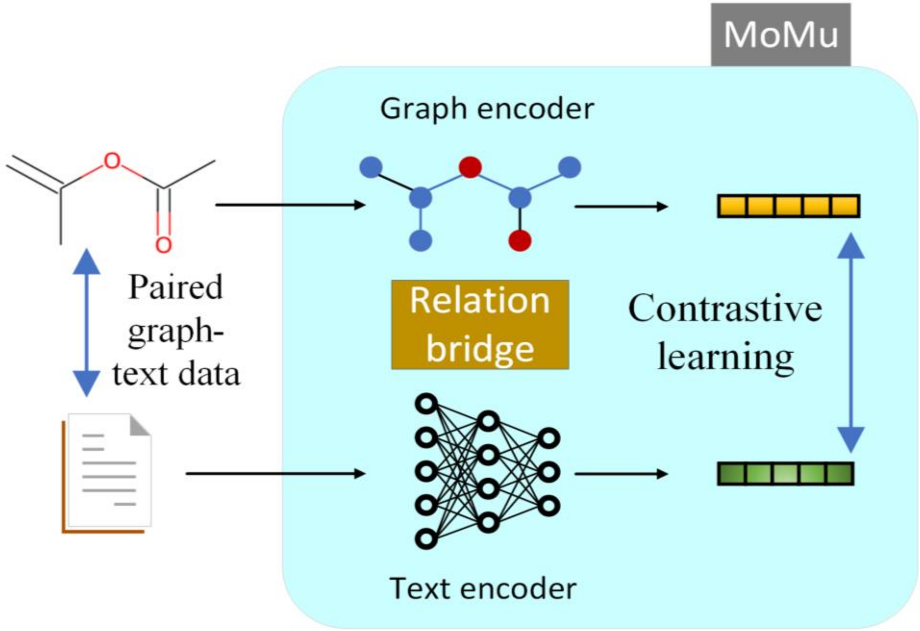
Textual attributes → LLM → Prediction & Explanation → Fine-tune LM → Node features → GNN

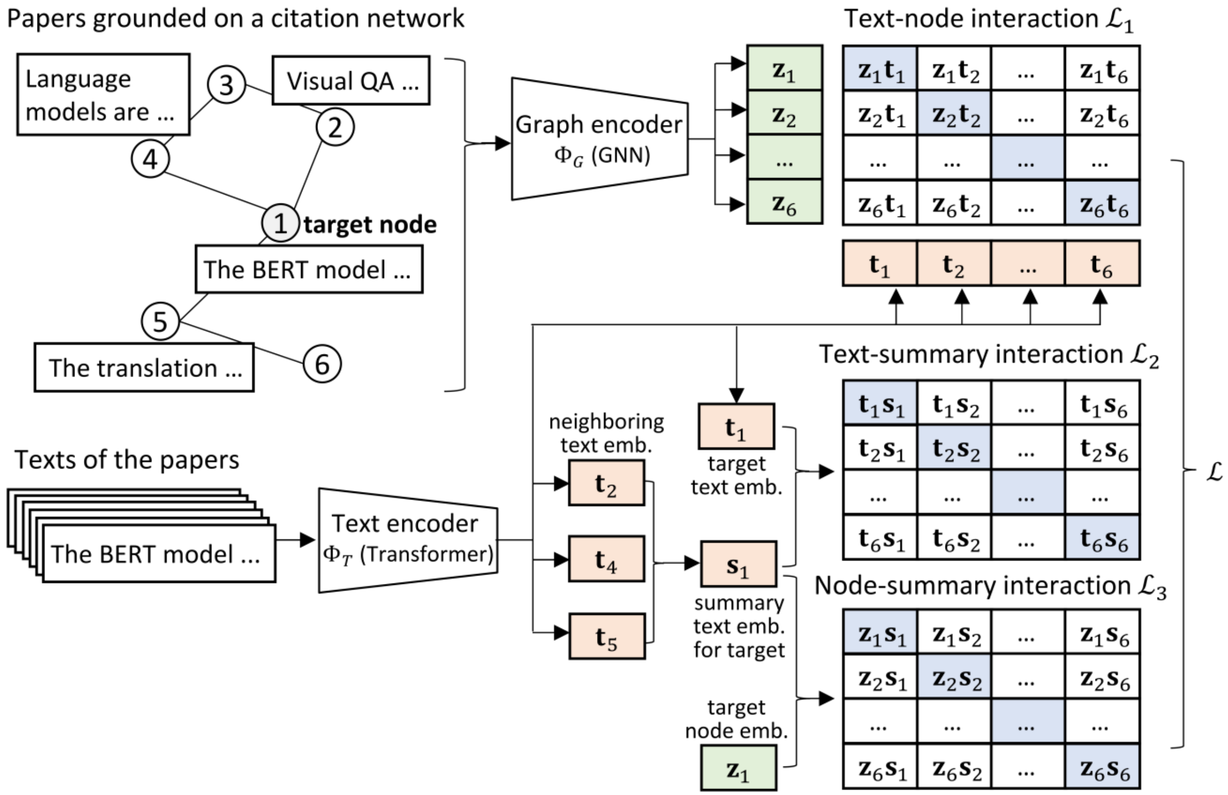


He, et al. "Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning." *ICLR'24*

Symmetric Methods: MoMu, G2P2

- The backbone model:
  - Dual encoders: Graph & Text encoder
  - Contrastive Learning





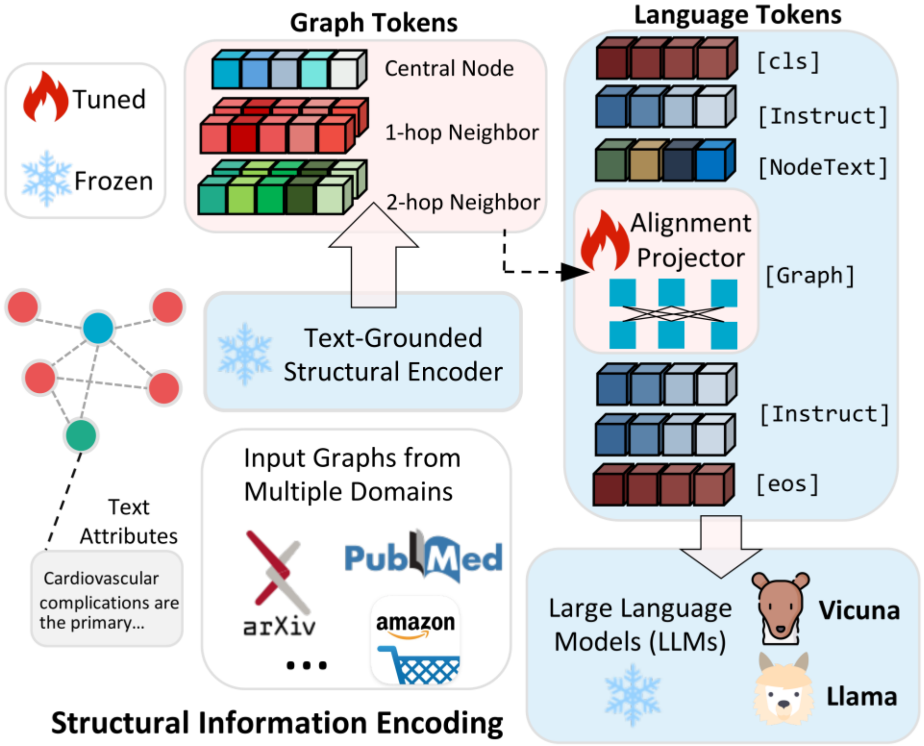
Su, et al. "A molecular multimodal foundation model associating molecule graphs with natural language." *CoRR'22*.

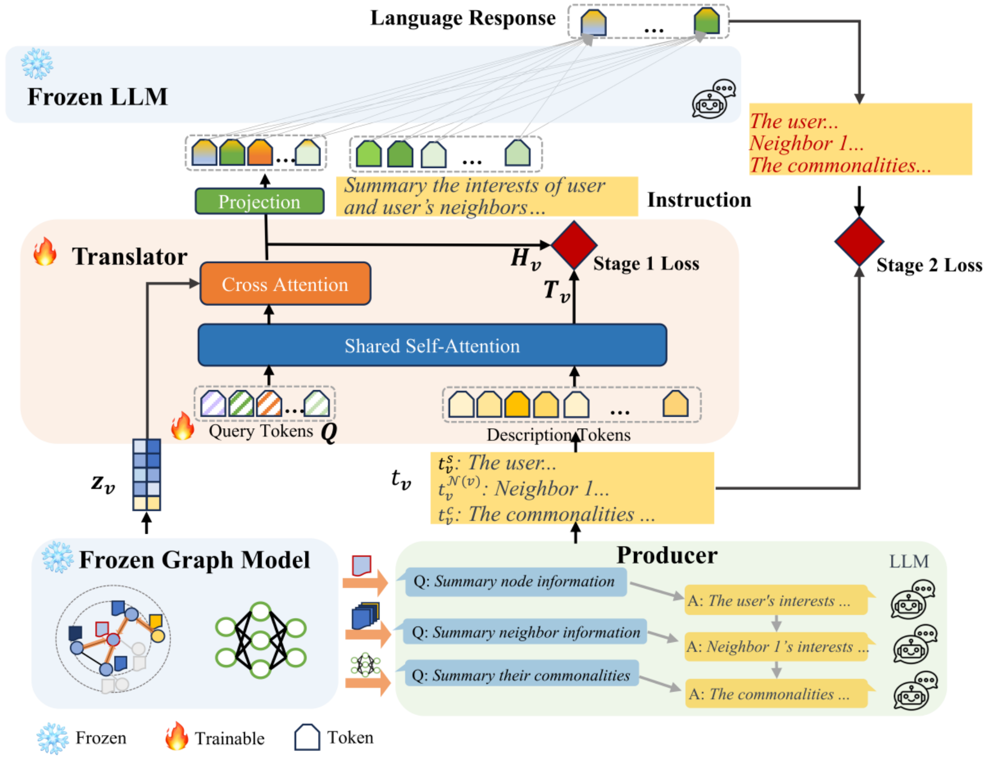
Wen, et al. "Augmenting low-resource text classification with graph-grounded pre-training and prompting." *SIGIR'23*.

LLM-centric Methods: GraphGPT, GraphTranslator

- The backbone model:

  Graph → GNN → Projection → LLM





Tang, et al. "GraphGPT: Graph instruction tuning for large language models." *SIGIR'24*

Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

GNN+LLM based Models

- ❑ Backbone Architecutures
- ❑ **Pre-training**
- ❑ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

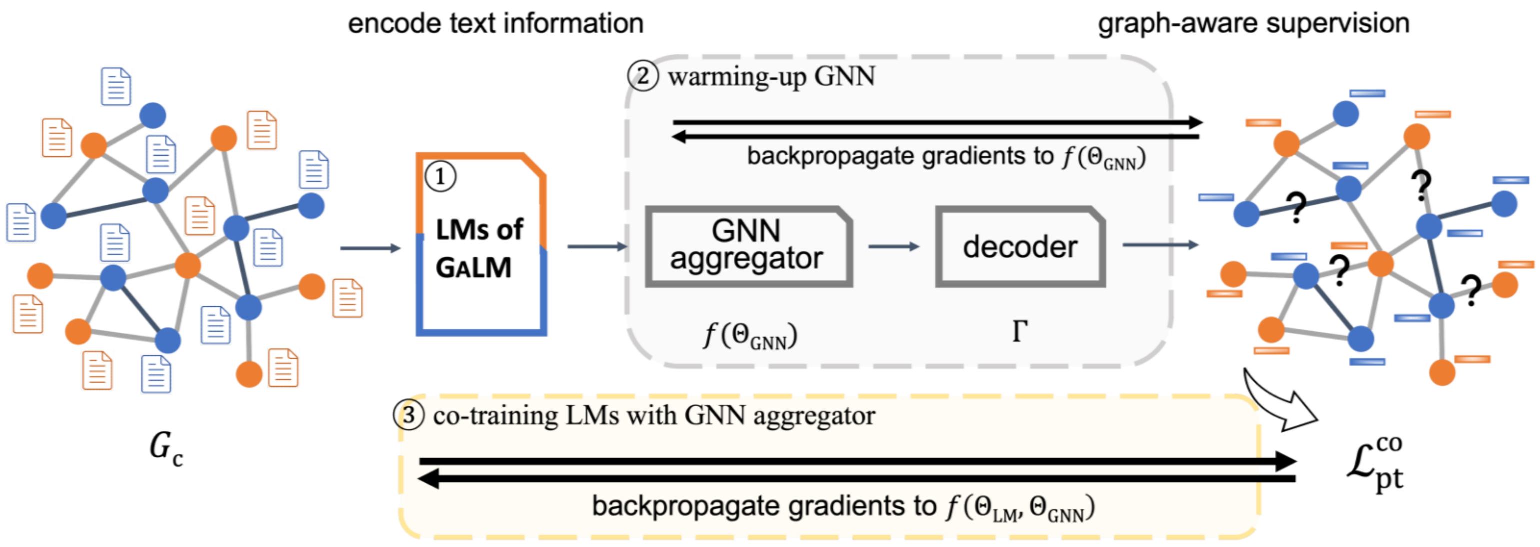
Table 4. Details of approaches involved as GNN+LLM based models

Pre-training

- GNN or LLM-based
  - Masked Language Modeling
  - Language Modeling
  - Text-Text Contrastive Learning
  - Graph reconstruction
- Alignment-based
  - Graph-Text Contrastive Learning

GNN or LLM-based: GaLM

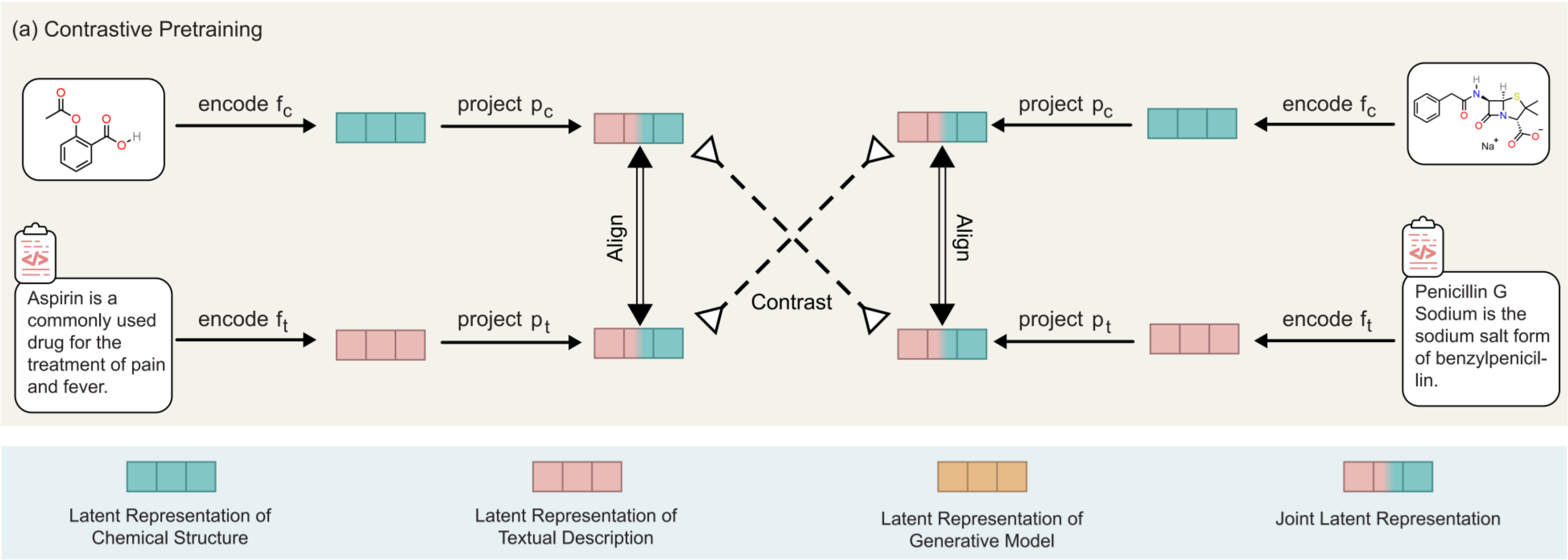
- GaLM (Graph-aware Language Model pre-training):
  - Fine-tuning existing general LMs by graph-aware supervision
  - Warming up the GNN aggregator by fixing the pre-trained LMs
  - Co-training GNN+LMs



Xie, et al. "Graph-aware language model pre-training on a large graph corpus can help multiple graph applications." *KDD'23*.

Alignment-based: MoleculeSTM

- Graph-Text Contrastive Learning (GTCL)
  - Map the graph and text representations extracted to a joint space using two projectors (p_c and p_t) via contrastive learning

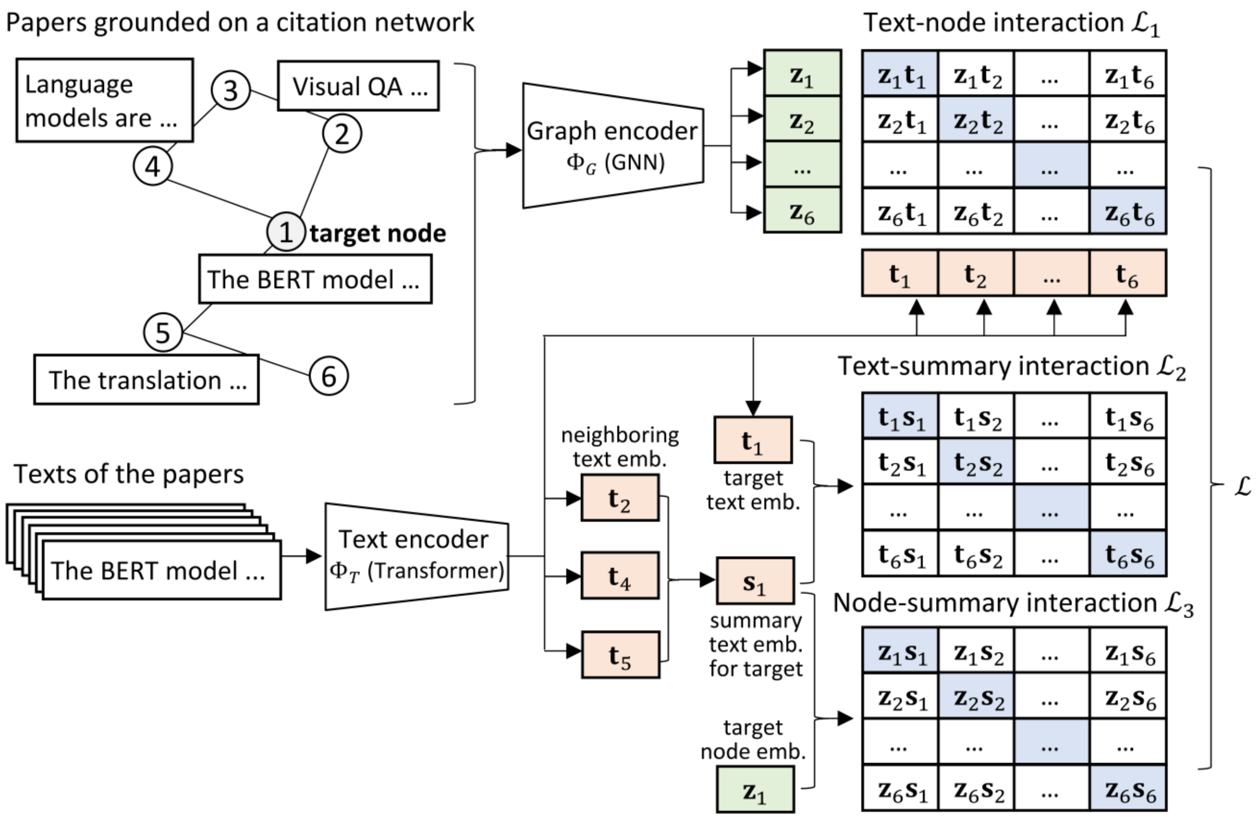


Liu, et al. "Multi-modal molecule structure–text model for text-based retrieval and editing." *Nature Machine Intelligence* 2023

Alignment-based: G2P2

- Dual encoders
- Three kinds of alignments
  - Text-Node: L1
  - Text summary-Text: L2
  - Text summary-Node: L3
    - Text-summary: text of neighbors

$$\mathbf{s}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{t}_j$$



Wen, et al. "Augmenting low-resource text classification with graph-grounded pre-training and prompting." *SIGIR'23*.

GNN+LLM based Models

- ❑ Backbone Architecutures
- ❑ Pre-training
- ❑ **Adaptation**

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

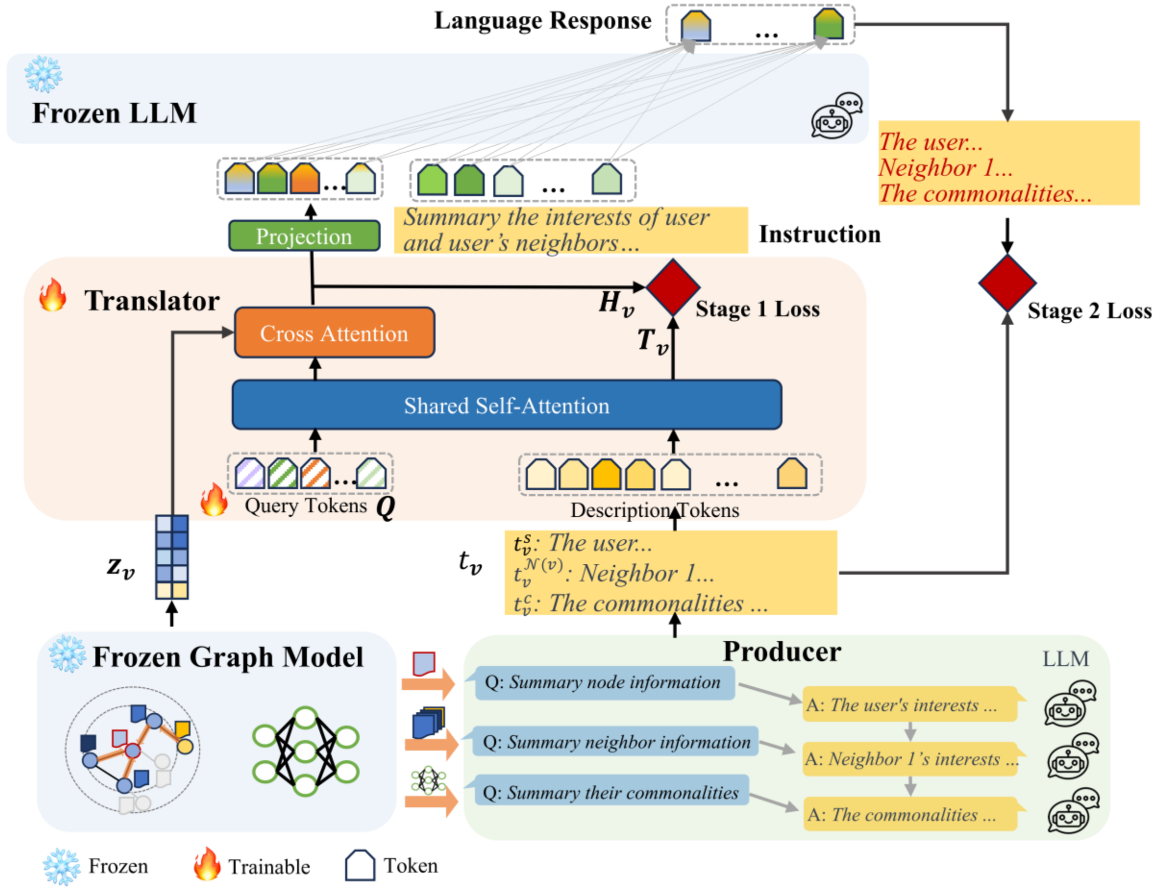
Table 4. Details of approaches involved as GNN+LLM based models

Adaptation

- Fine-tuning
  - Vanilla tuning: tune all the parameters
    - computationally intensive, resource-demanding
  - Parameter-efficient fine-tuning (PEFT): tune a subset of parameters
    - more efficient, resource-friendly
- Prompt-Tuning: design and tune external prompts

PEFT: GraphTranslator

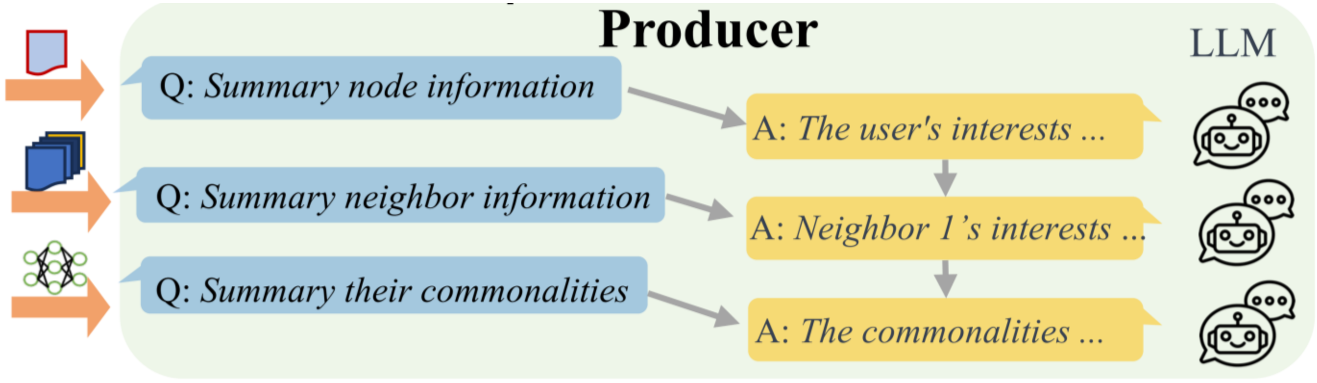
- Frozen:
  - Graph Model
  - Large Language Model
- Tunable:
  - Producer Module
    - Construct alignment data
  - Translator Module
    - Convert node representations into tokens for LLM prediction



Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

PEFT: GraphTranslator

- Producer:
  - ➢"Chain of Thought" (CoT) ->LLM->high-quality description
    - node information
    - neighbor information
    - commonalities



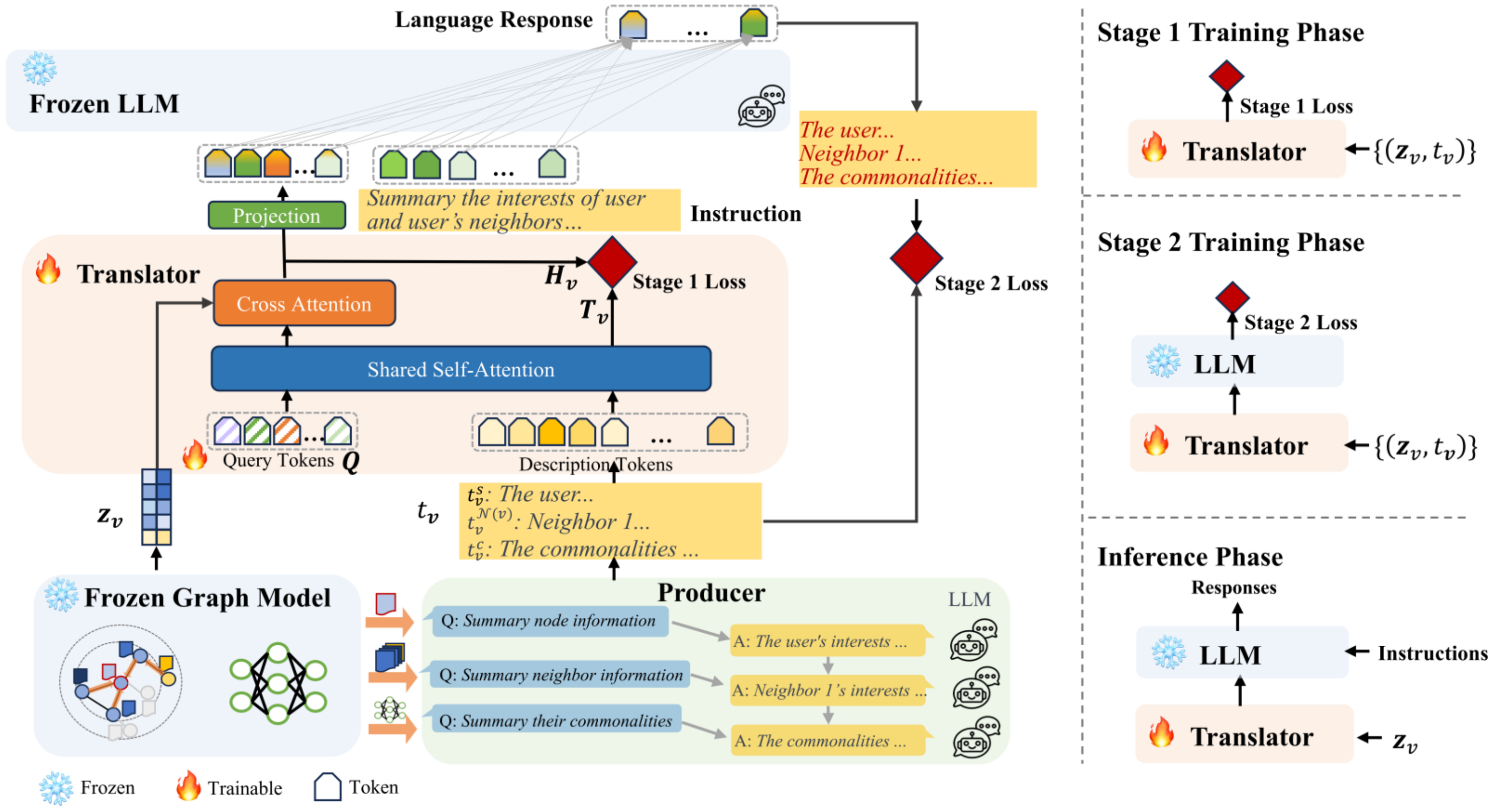
- Prompt template:

Dataset	Step	Prompt
Taobao	User behavior summary	User Behavior Description: <User Behavior Description>. Please summarize the characteristics of this user according to the product behavior information. The answer format is: What kind of characteristics does the user have in terms of interests, hobbies, personality traits, and life needs
	Neighbor behavior summary	Neighbor Behavior Description: <Neighbor Behavior Description>. Please summarize most of the similarities that this user's friends have based on the product behavior information. The answer format is: What do several friends of this user have in common in interests, hobbies, personality traits, and life needs?

Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

PEFT: GraphTranslator

- Training: Only fine-tune Translator and Projection

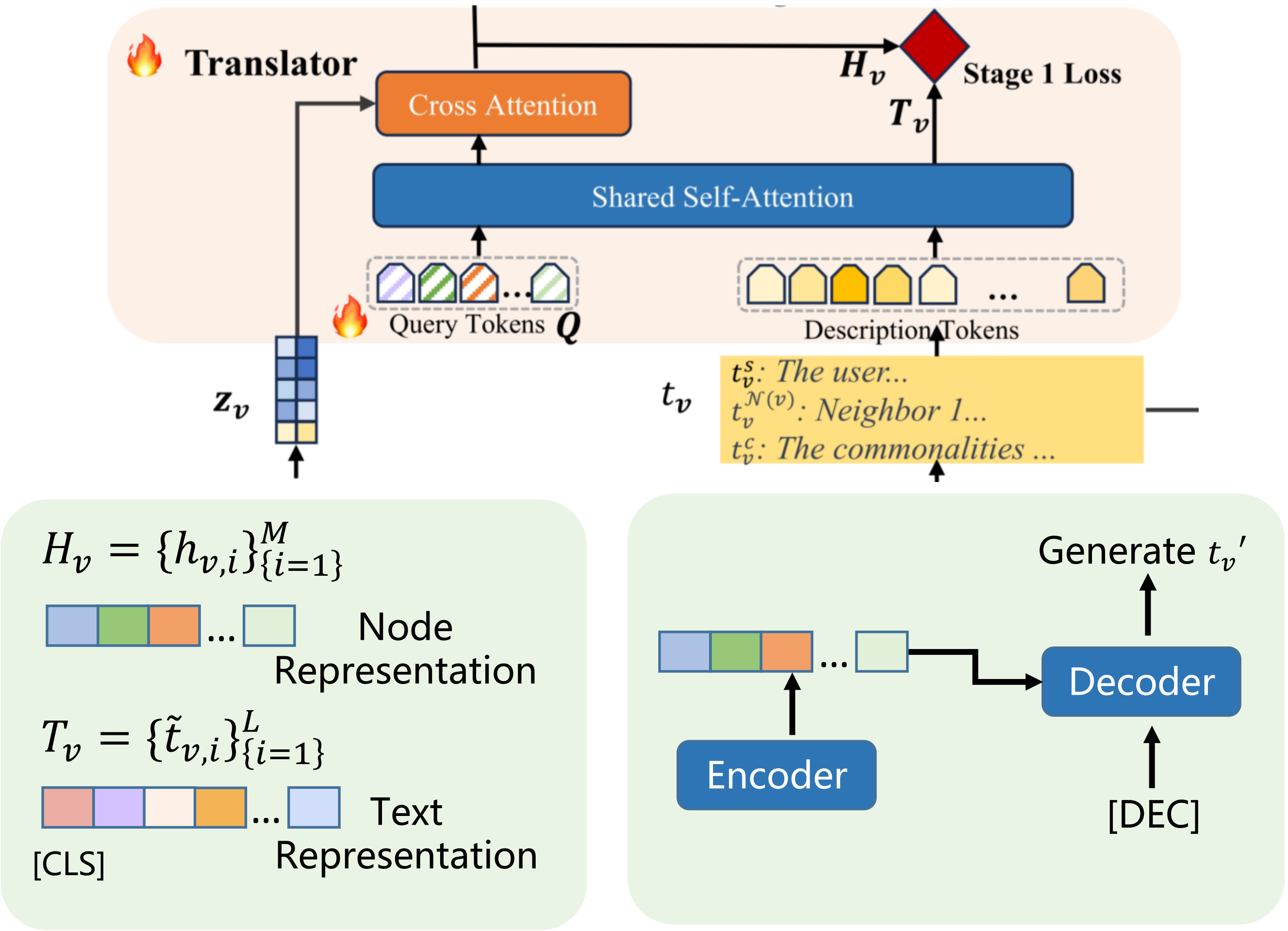


- Stage1: Align graph-text
- Stage2: Align graph-LLM

Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

PEFT: GraphTranslator

❑ Training: Stage 1

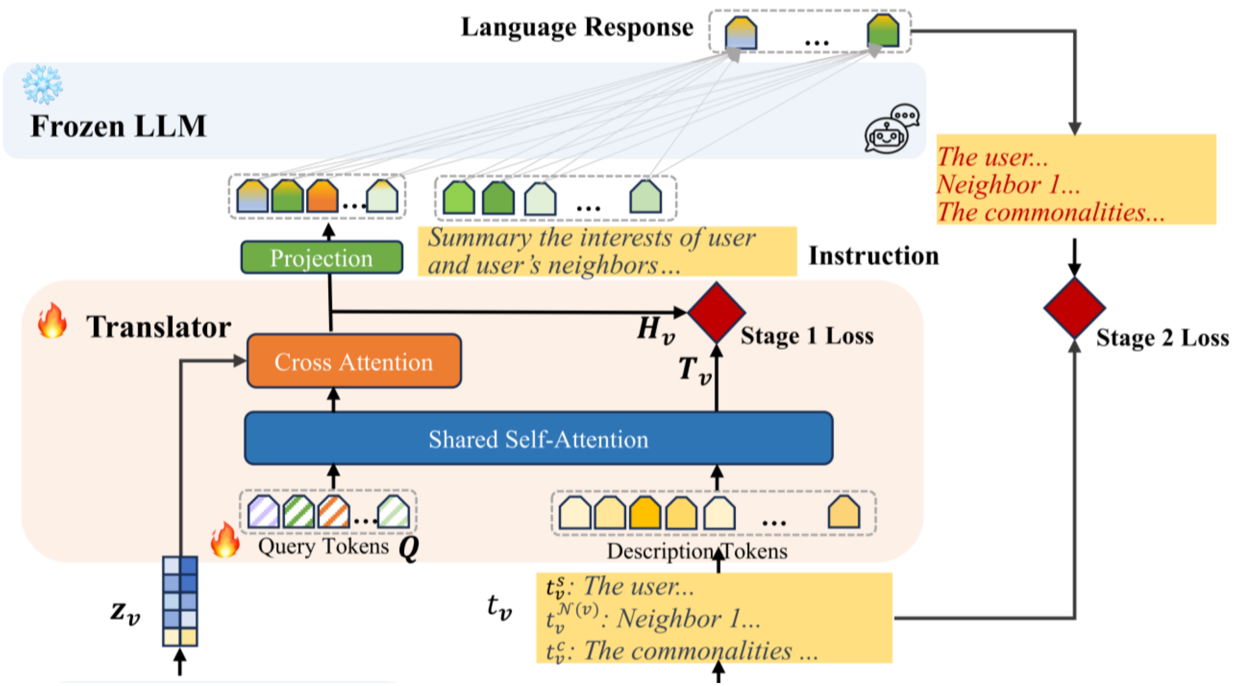


- Contrastive Objective
  - Node ↔ Text
  - High-level alignment
- Matching Objective
  - Node ↔ Text
  - Fine-grained alignment
- Generation Objective
  - Node → Text
  - Replace the [CLS] token with a new [DEC] token as the first text token to signal the decoding task

Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

PEFT: GraphTranslator

❑ Training: Stage 2

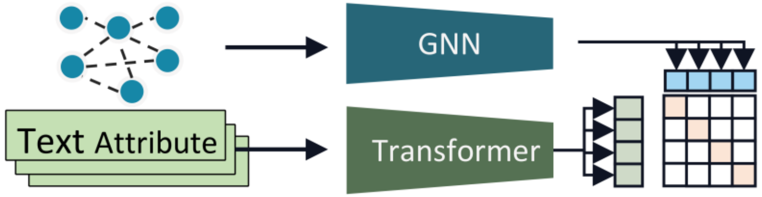


- Projection:
  - A linear layer: project H_v to token representation space of LLM
- Concatenate:
  - Connect the projected representation with the human instruction and feed into LLM
- Fine-tune Translator
  - Alignn the response text of LLM with the actual descriptive text

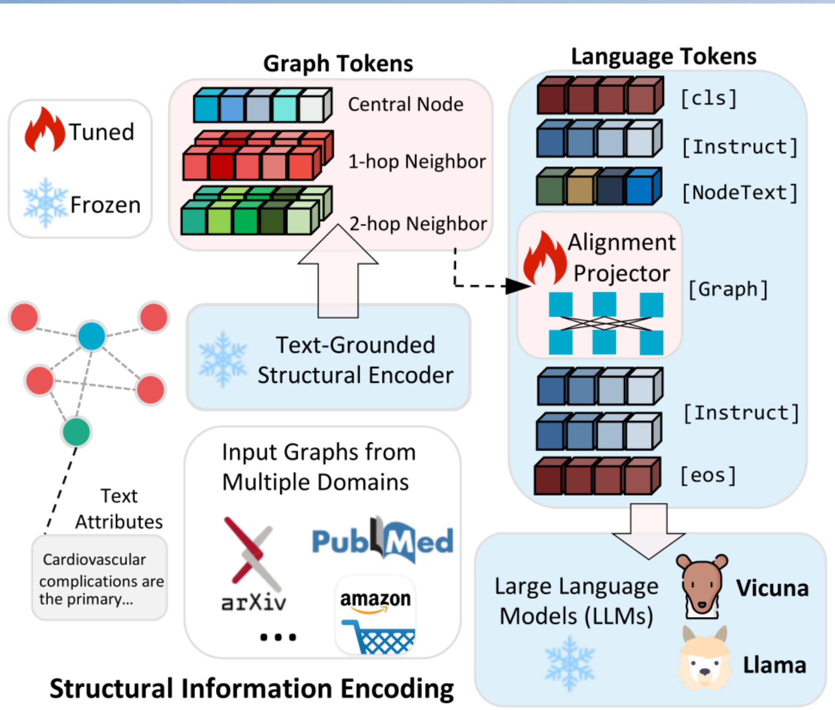
Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." WWW'24

PEFT: GraphGPT

- Graph: Text-Grounded Structural Encoder



- Projector: Map graph representation to LLM
- Instruction Tuning: Only fine-tune projector



Graph Information: <graph>: **Central Node:** 68442, **Edge index:** [[…src node…],[…dst node…]], **Node list:** […] **Graph Matching**
Human Question: Given a sequence of graph tokens <graph> that constitute a subgraph of a citation graph, …. Here is a list of paper titles: 1. … 2. …, please reorder the list of papers according to the order of graph tokens.
GraphGPT Response: Based on the given graph tokens and the list of paper titles, we obtain the matching of graph tokens and papers: Graph token 1 corresponds to smt based induction methods for timed systems. Graph token 2 corresponds to …

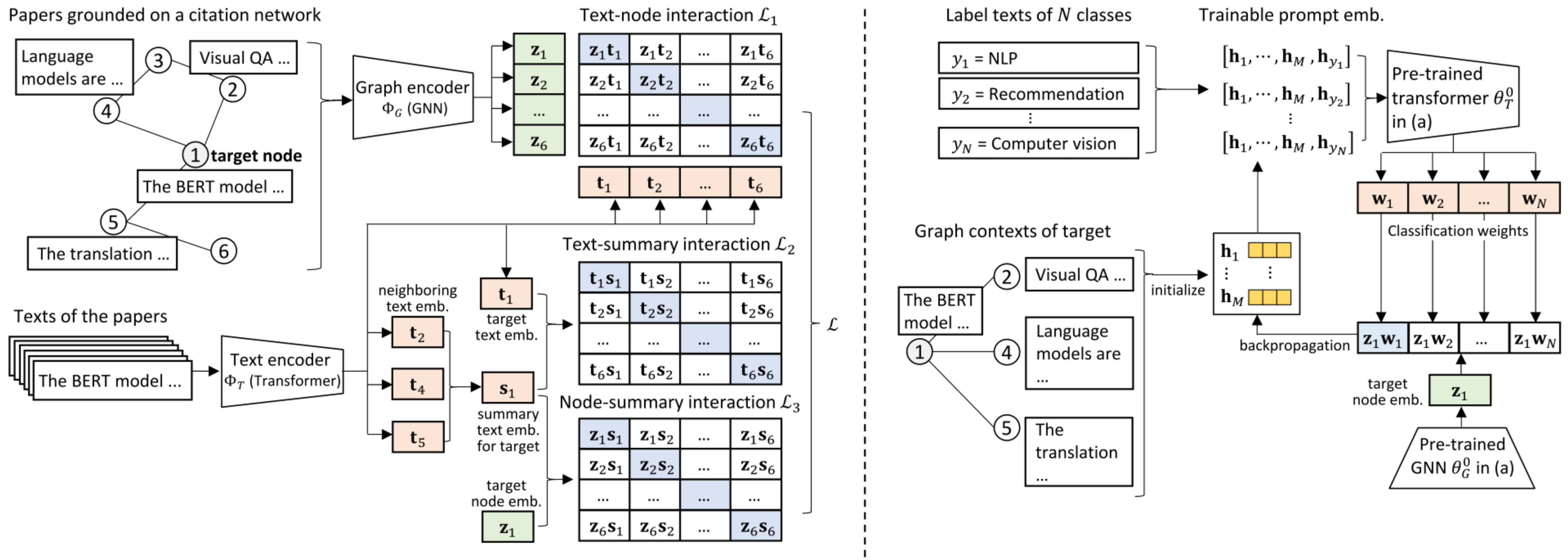
Graph Information: <graph>: **Central Node:** 2, **Edge index:** [[…src node…],[…dst node…]], **Node list:** […] **Node Classification**
Human Question: Given a citation graph: <graph> where the 0th node is the target paper, with the following information: Abstract: … Title: … Question: Which arXiv CS sub-category does this paper belong to? …
GraphGPT Response: cs.IT, cs.LG, cs.SP, cs.CV, cs.NA. The paper discusses the Restricted Isometry …. So, it is likely to belong to cs.IT…

Graph Information: <graph>: **Central Node 1:** 8471, **Edge index 1:** [[…src node…],[…dst node…]], **Node list 1:** […] **Link Prediction**
<graph>: **Central Node 2:** 19368, **Edge index 2:** [[…src node…],[…dst node…]], **Node list 2:** […]
Human Question: Given a sequence of graph tokens: <graph> that constitute a subgraph of a citation graph, …. Abstract: … Titile: … and the other sequence of graph tokens: <graph>, … Abstract: … Title: …, are these two central nodes connected? Give me an answer of "yes" or "no".
GraphGPT Response: Yes, they are connected. Based on the first paper, …. And the second paper proposes ….

Tang, et al. "GraphGPT: Graph instruction tuning for large language models." *SIGIR'24*

Prompt-Tuning: G2P2

- ❑ Learnable prompts: $[h_1, \cdots h_M, h_{CLASS}]$
- ❑ Tuning prompts with limited labeled data for efficient adaptation



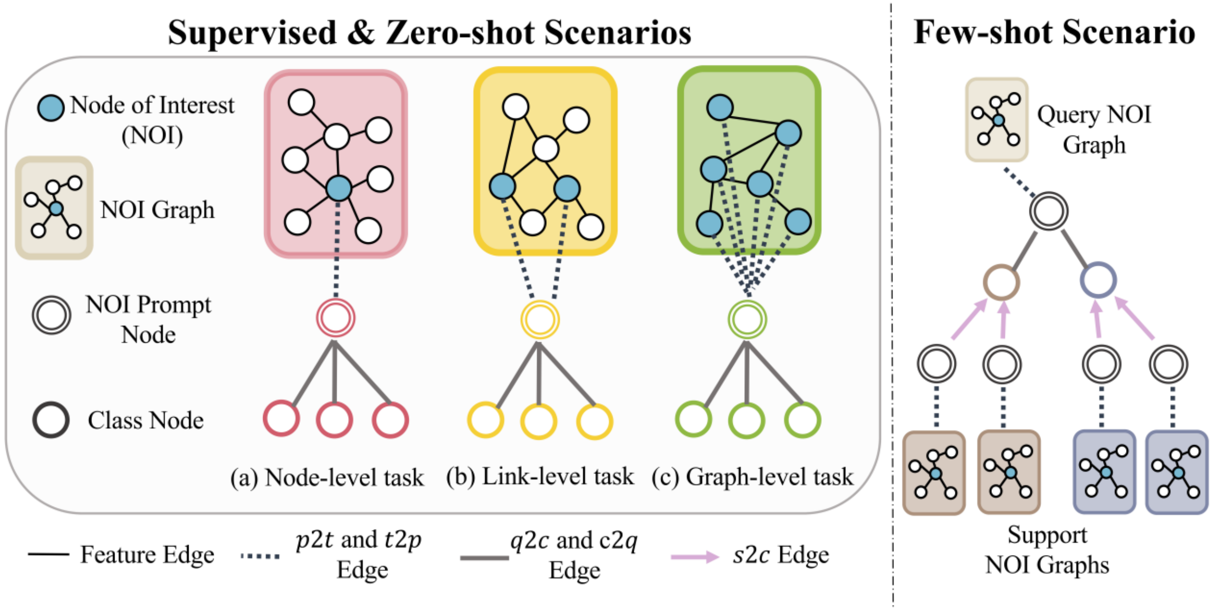
(a) Graph-grounded contrastive pre-training

(b) Graph-grounded prompt tuning (few-shot classification)

Wen, et al. "Augmenting low-resource text classification with graph-grounded pre-training and prompting." *SIGIR'23*.

Prompt-Tuning: One for all

- ❑ NOI (Node of Interest):
  - ➢ Node-level: node
  - ➢ Link-level: node pair
  - ➢ Graph-level: subgraph



- ❑ NOI Prompt Node

> **Text feature of the NOI prompt node:** Prompt node. <*task description*>.
> **Example:** Prompt node. Graph classification on molecule properties.
> **Example:** Prompt node. Node classification on the literature category of the paper.

- ❑ Class Node

> **Text feature of class node:** Prompt node. <*class description*>.
> **Example:** Prompt node. Molecule property. The molecule is effective in: ...
> **Example:** Prompt node. Literature Category. cs.AI (Artificial Intelligence). Covers all areas of AI except Vision ...

Liu, et al. "One for all: Towards training one graph model for all classification tasks." *ICLR'24*.

Outline

- LLM based Models
  - Backbone Architecutures
  - Pre-training
  - Adaptation
- GNN+LLM based Models
  - Backbone Architecutures
  - Pre-training
  - Adaptation
- **Summary and outlook**

Summary and outlook

❑ Summary

- ➢ Leveraging LLMs facilitates a unified approach to various graph tasks by describing them in natural language.
- ➢ Merging graph data, text, and other modalities into LLMs creates a promising path for graph foundation models.
- ➢ Combining GNNs and LLMs leads to improved performance in graph-related tasks.

Summary and outlook

❑ Outlook

- Focus on resolving LLMs' limitations: multi-hop reasoning, graph topology, and diverse graph data.
- Explore efficient training methods to manage the high computational costs and data requirements.
- Explore applications of GNN+LLM models in multimodal and cross-modal tasks.