Beyond Small GNNs: Graph Foundation Models

Jiaxuan You

Assistant Professor at UIUC CDS

CS598: Deep Learning with Graphs, 2024 Fall

https://ulab-uiuc.github.io/CS598/

Today's Lecture

- Retrospectives of Foundation Models
- Towards Graph Foundation Models
- Discussions of Recent Work
  - GNN-based methods
  - LLM-based methods
- Open questions for GFM

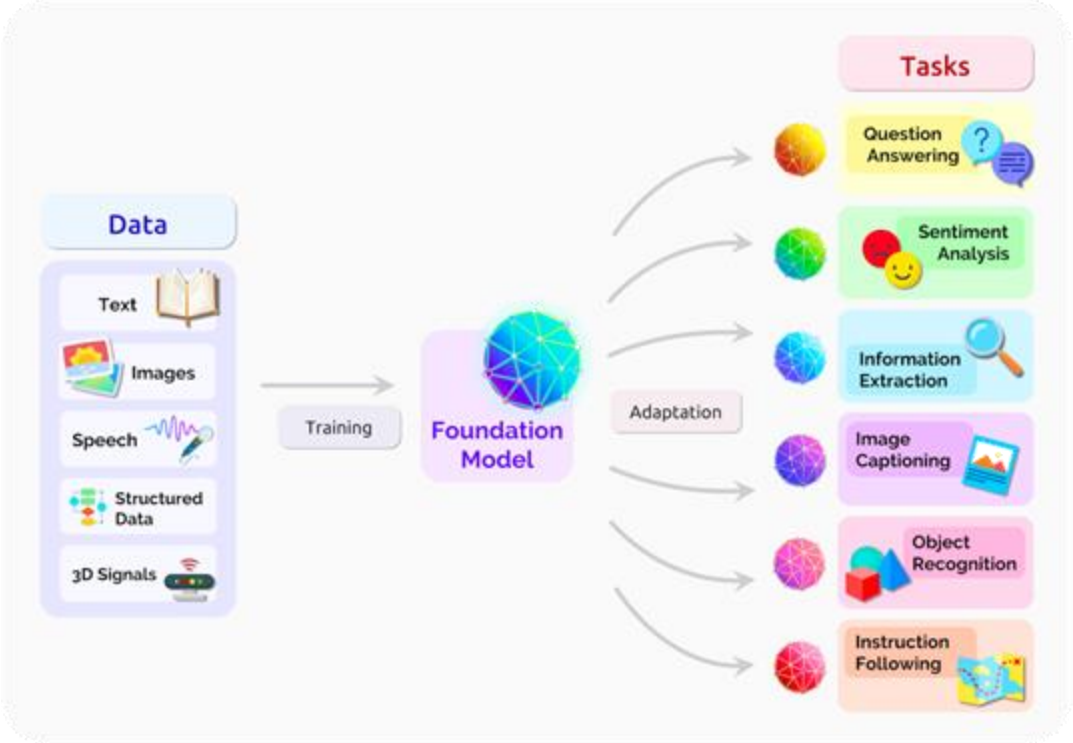
Beyond Small GNNs: Graph Foundation Models

Background: Foundation Models

Bommasani et al. On the Opportunities and Risks of Foundation Models. 2021.

Foundation Models

- A foundation model is any model that **is trained on broad data** (generally using self-supervision at scale) that **can be adapted** (e.g., fine-tuned) **to a wide range of downstream tasks**.



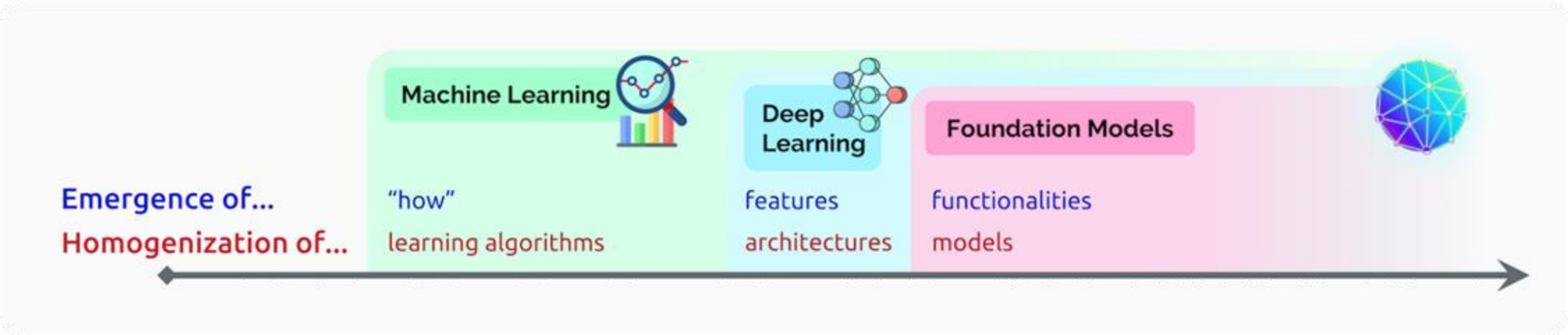
Foundation Models

- From a technological point of view, foundation models are **not new**.
  - They are based on **deep neural networks** and **self-supervised learning**, both of which have existed for decades.
- However, the sheer **scale** and **scope** of foundation models from the last few years have stretched our imagination of what is possible.
- **Scaling law:**
  - x = **N** (parameter count), **D** (dataset size), **C** (compute cost), **L** (loss)

$$L(x) = L_\infty + \left(\frac{x_0}{x}\right)^{\alpha_x}$$

Characteristics of Foundation Models

- The significance of foundation models can be summarized by two words: **emergence** and **homogenization**.
  - **Emergence** means that the behavior of a system is implicitly induced rather than explicitly constructed.
  - **Homogenization** indicates the consolidation of methodologies for building machine learning systems across a wide range of applications.



Driving Factors of Foundation Models

- On a technical level, foundation models are enabled by **transfer learning** and **scale**.
- **Transfer/multi-task learning** aims to take the "knowledge" learned from one task (e.g., object recognition in images) and apply it to another task (e.g., activity recognition in videos).
  - **Pretraining** is the dominant approach to transfer learning: a model is trained on a surrogate task (often just as a means to an end) and then adapted to the downstream task of interest via **fine-tuning**.

Bommasani et al. On the Opportunities and Risks of Foundation Models. 2021.

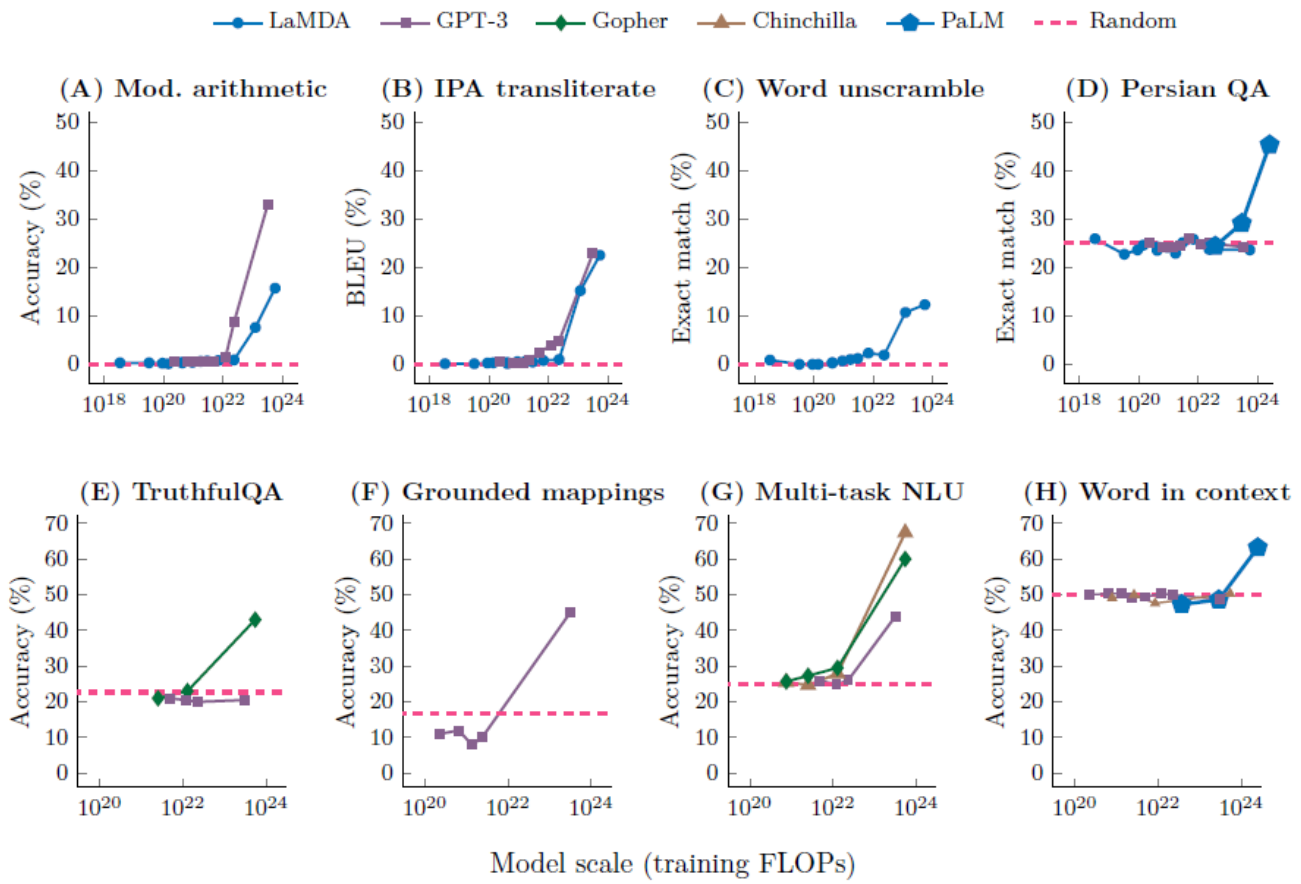
Driving Factors of Foundation Models

- **Scale** required three ingredients:
  - Improvements in computer **hardware**
  - Development of the Transformer **model architecture**
  - Availability of much more **training data**
  - Exploiting raw unlabeled data with **self-supervised** learning

Wei et al. Emergent Abilities of Large Language Models. TMLR, 2022.

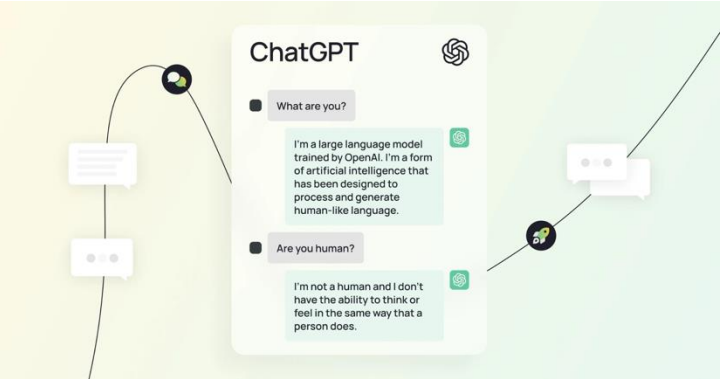
Progresses of Foundation Models

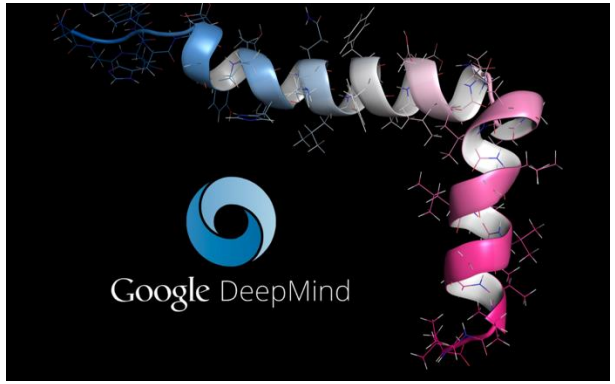
- Surprising emergence which results from scale



Progresses of Foundation Models

- Multimodal models from different research communities



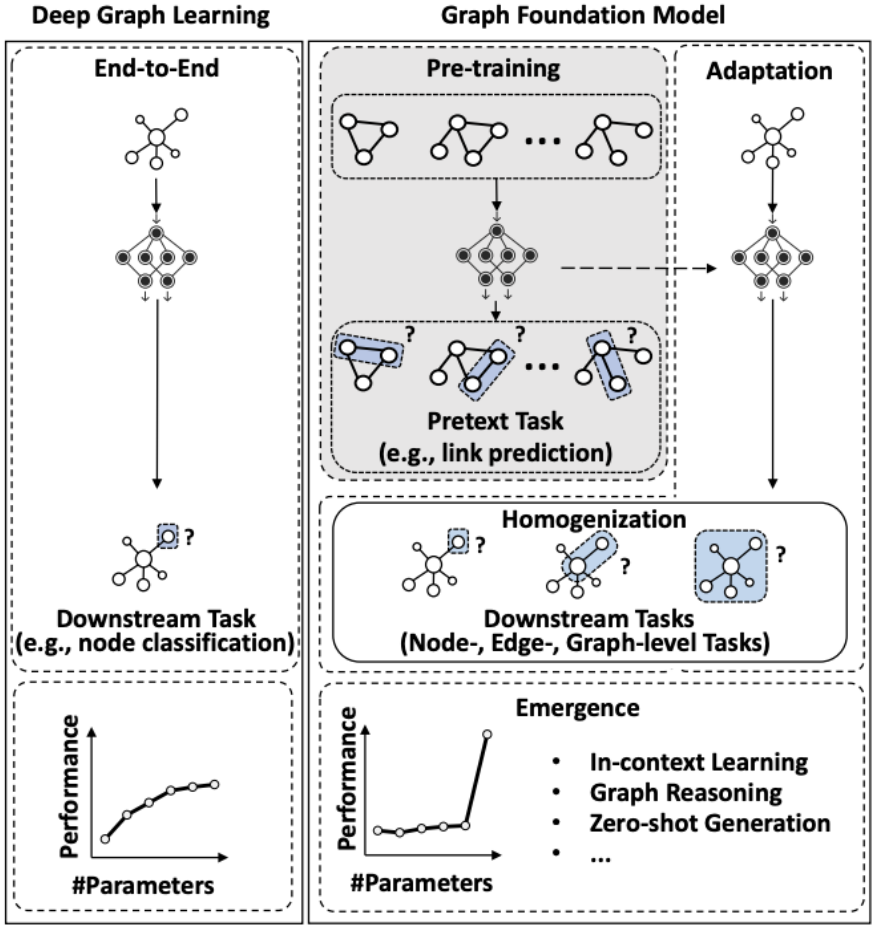


Beyond Small GNNs: Graph Foundation Models

Graph Foundation Models

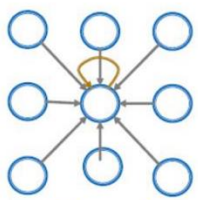
Graph Foundation Models

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



Graph Foundation Models

- **Extensive graph data**
  - Graph data: non-Euclidean data
  - Various domains
    - Social networks
    - Molecules
    - E-commerce
  - Various types
    - Homogeneous graph
    - Heterogeneous graph
    - Hypergraph


Graph

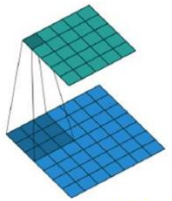
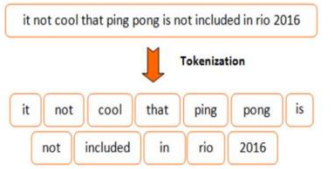
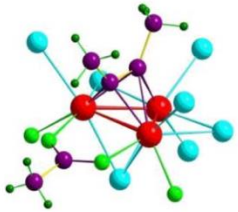


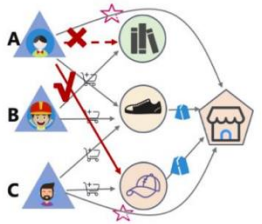
Image (Grid)



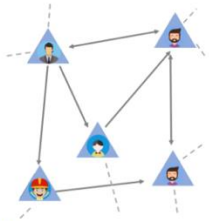
Language (Seq.)

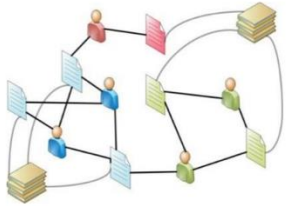

Social Networks

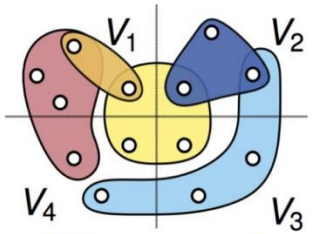

Molecules



E-commerce


Homogeneous

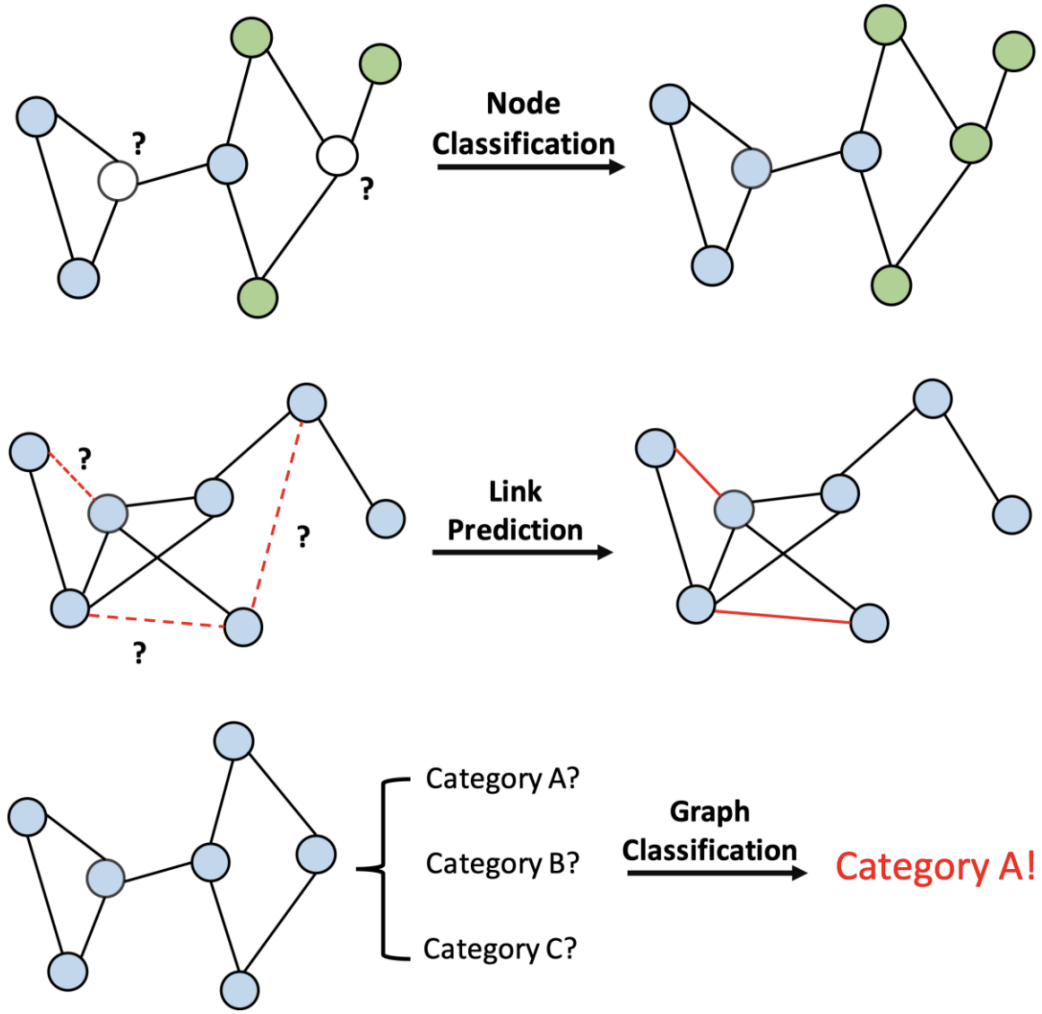

Heterogeneous



Hypergraph

Graph Foundation Models

- **Diverse downstream graph tasks**
  - **Node-level tasks**
    - Node classification
    - Node regression
    - Node clustering
  - **Edge-level tasks**
    - Link prediction
  - **Graph-level tasks**
    - Graph classification
    - Graph generation



Key Techniques

- Graph foundation models require two key techniques: **pre-training** and **adaptation**.
  - **Pre-training**: neural networks are trained on a large graph dataset in a self-supervised manner (e.g., attribute masking, contrastive learning)
  - **Adaptation**: adapt pre-trained models to specific downstream tasks or domains to enhance their performance (e.g., finetuning, prompt-tuning)

GFMs vs. 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

Beyond Small GNNs: Graph Foundation Models

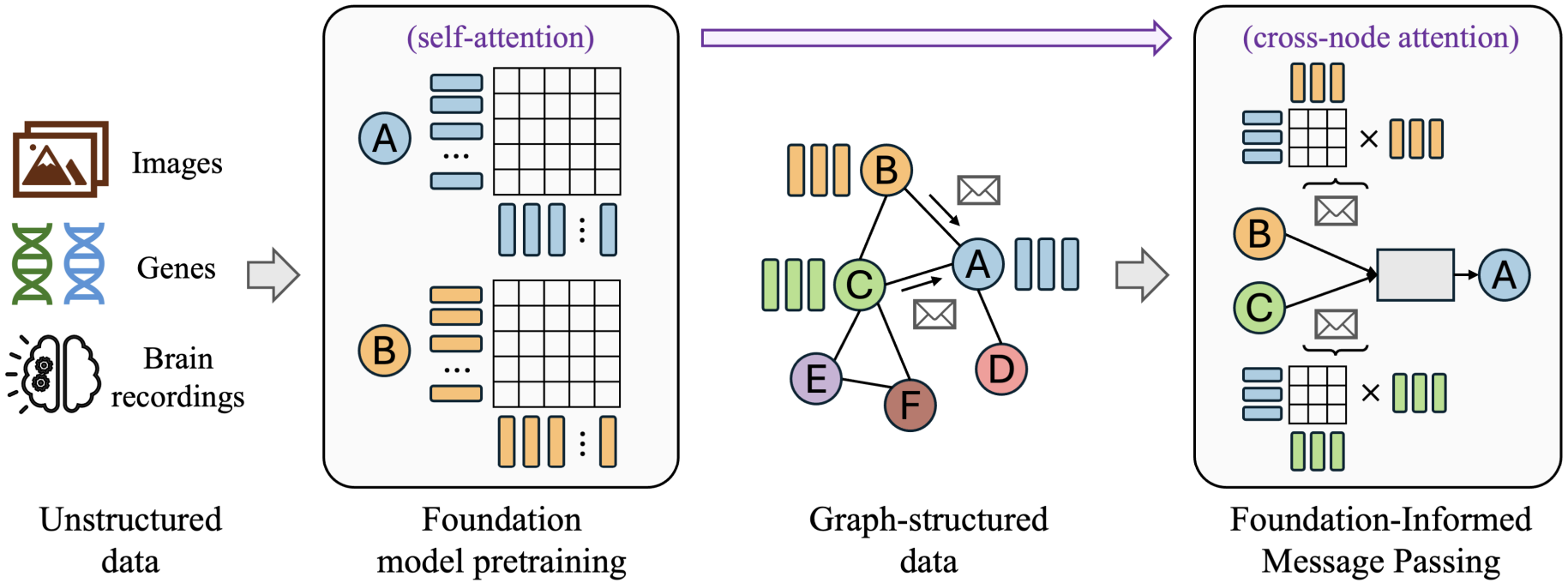
GNN-Based Methods

GNN-Based Methods

- No GFMs until now, but a lot of explorations is on the way.
- **GNN-based methods** seek to enhance current graph learning through innovative approaches in graph neural network architectures.
- We discuss two papers from this perspective.
  - FIMP: Foundation Model-Informed Message Passing for Graph Neural Networks
  - GraphProp: Training the Graph Foundation Models using Graph Properties

Foundation-Informed Message Passing for Graph Neural Networks

- **Key Idea**: leveraging pre-trained non-textual foundation models to generate messages between neighboring nodes on graph-based tasks.
  - Unify node entity tokenization between foundation models and GNNs.
  - Develop a cross-node attention-based message creation module.



Cross Attention Based Message Passing

- In FIMP, nodes are tokenized into sequences of feature vectors H_i.
  - Cross node attention between the feature sequences of neighboring nodes is used to create messages. A transformation function τ, position encoding P

$$H_i = \tau(X_i) = \text{CONCAT}(X_i \boldsymbol{W}, P) \in \mathbb{R}^{f \times d}$$

- We define a **cross-node attention-based** message creation module:

$$Q = H_i \boldsymbol{W}_Q$$
$$K = H_j \boldsymbol{W}_K$$
$$V = H_j \boldsymbol{W}_V$$
$$H_{ji} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

- Message H_{ji} can be aggregated and used to complete the regular message passing aggregation and update steps

Message Creation

- Objective: to formulate message creation between two nodes
  - Pretrained foundation models can be leveraged to create the messages
  - Fitting into the rest of the message passing framework
- Observation: transformer-based foundation models operate using self-attention over sequences of feature tokens and contain learned attention weights per layer which are trained to highlight important interactions between feature tokens.

Rizvi et al. FIMP: Foundation Model-Informed Message Passing for Graph Neural Networks. arXiv, 2022.

Leveraging Non-Textual Foundation Models

- In the previous formulation, cross-attention message passing can be done with a simple cross-attention mechanism which is **learned from scratch during training**.
- Pretrained foundation models can be repurposed to do the message creation to leverage their pretraining over vast amounts of unstructured data.
- We can adapt the self-attention mechanism in each layer to do cross attention between node feature sequences from neighboring nodes.

Experiments and Findings

- **Findings**: FIMP demonstrates improved performance over baselines across multiple tasks in image networks, spatial transcriptomics data, and fMRI brain activity recordings.

Method	Mouse Hippocampus		Human Heart	
	MSE (↓)	R^2 (↑)	MSE (↓)	R^2 (↑)
GCN	0.0211 ± 0.0018	0.0236 ± 0.0457	0.0045 ± 0.00019	0.3368 ± 0.04453
GraphSAGE	0.0181 ± 0.0012	0.1853 ± 0.0306	0.0054 ± 0.00033	0.2080 ± 0.01973
GAT	0.0201 ± 0.0008	0.0905 ± 0.0233	0.0043 ± 0.00023	0.3468 ± 0.02313
GIN	0.0175 ± 0.0009	0.1707 ± 0.0424	0.0025 ± 0.00029	0.6625 ± 0.01269
GraphMAE	0.0178 ± 0.0006	0.1538 ± 0.0254	0.0024 ± 0.00016	0.6589 ± 0.01715
GPS	0.0149 ± 0.0012	0.2977 ± 0.0308	0.0024 ± 0.00031	0.6538 ± 0.01043
FIMP-base (ours)	*0.0134 ± 0.0009*	*0.3815 ± 0.0226*	*0.0021 ± 0.00003*	*0.6955 ± 0.02048*
FIMP + ViT (ours)	0.0128 ± 0.0010	0.3506 ± 0.0452	0.0042 ± 0.00089	0.4026 ± 0.08102
FIMP + GenePT (ours)	0.0129 ± 0.0005	0.4058 ± 0.0302	0.0013 ± 0.00023	0.7952 ± 0.01430
FIMP + scGPT (ours)	**0.0119 ± 0.0008**	**0.4612 ± 0.0029**	**0.0011 ± 0.00008**	**0.8119 ± 0.01428**

Experiments and Findings

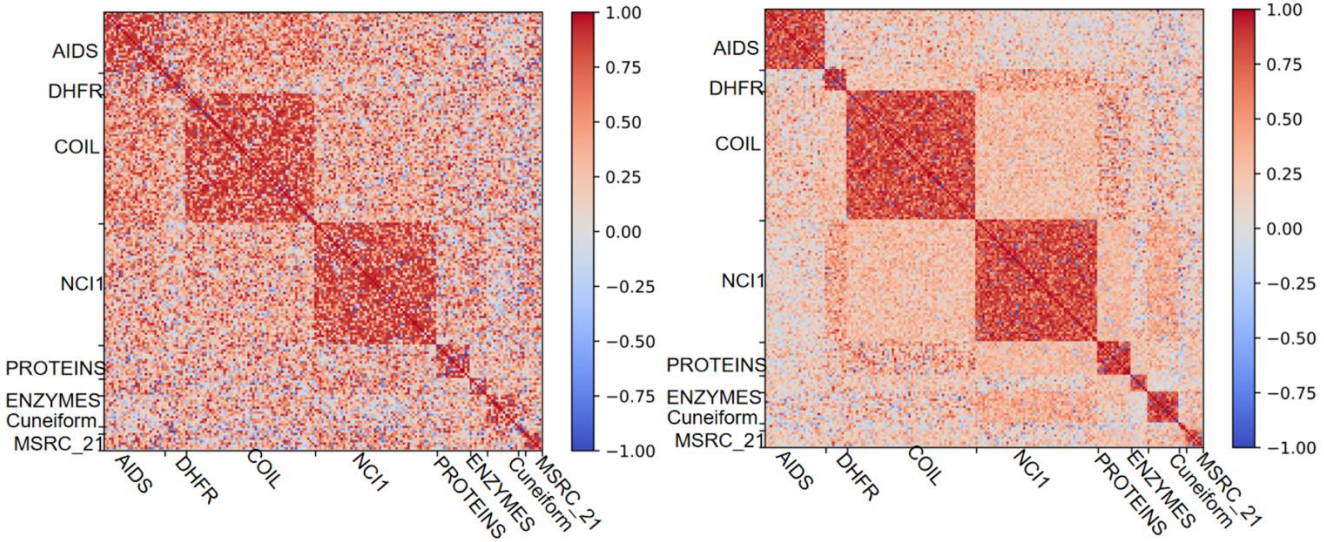
- **Findings**: FIMP demonstrates zero-shot embedding capabilities on image networks that are on par with trained GNNs.
  - This enables zero-shot applications on graphs, previously impossible with non-textual foundation models.

Training the Graph Foundation Models using Graph Properties

- **Key Idea**: graph properties like fractional chromatic number and Lovász number can generalize across different graphs
- **Goal**: to train GFMs that effectively learn across different domains by capturing consistent information shared by different domains.

Training the Graph Foundation Models using Graph Properties

- Question: **how much cross-domain consistent information do graph structures and node features contain, respectively**?
  - Example: both molecular data and social networks share common graph properties like the Lovász number.

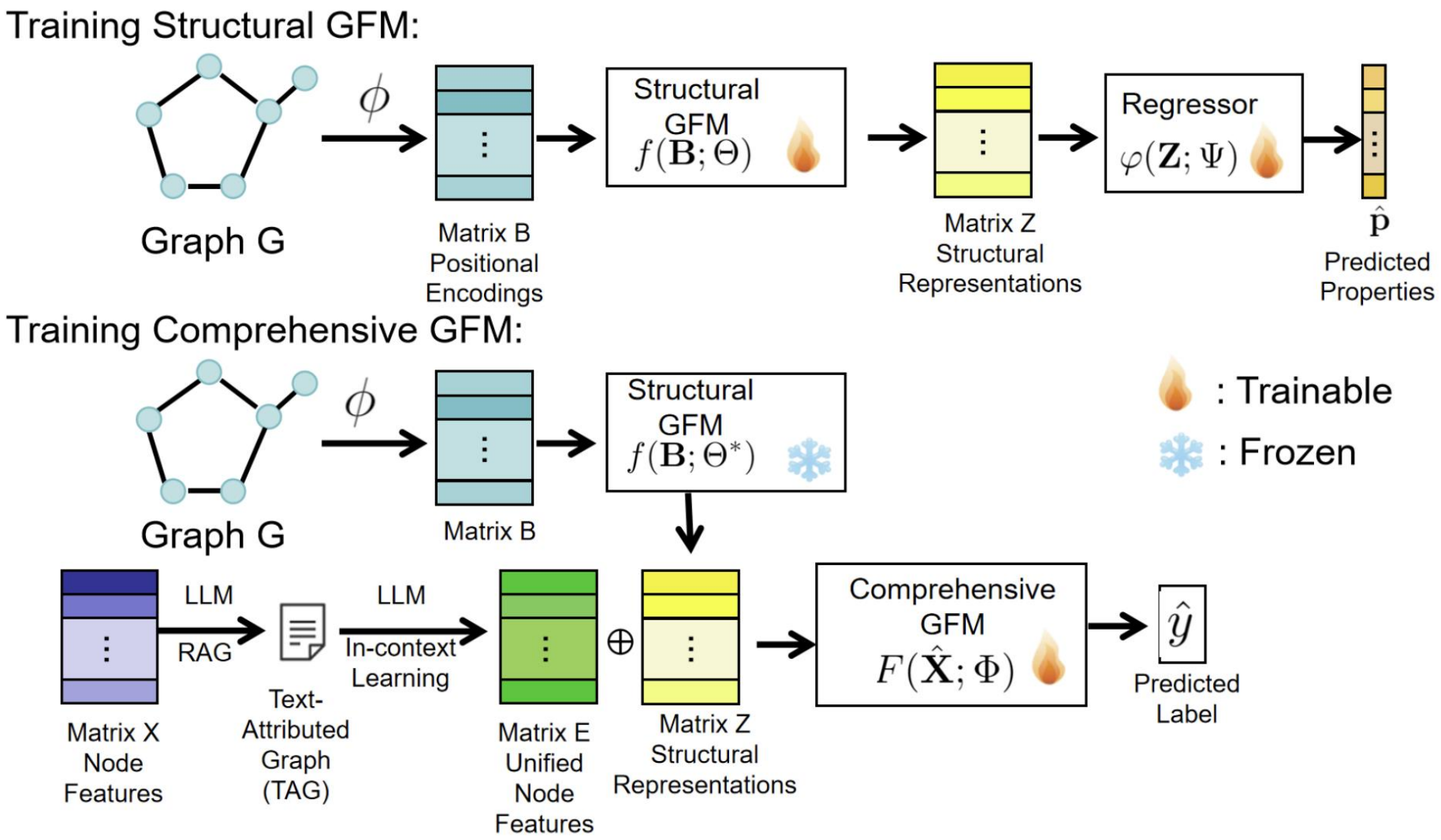


(a) Correlation coefficient matrix cor(C)

(b) Correlation coefficient matrix cor($\bar{\mathbf{E}}$)

Findings: Cross-domain correlation of graph structure representation matrix $\boldsymbol{C}$ is higher than that of node feature representation matrix $\overline{\boldsymbol{E}}$.

Training a Comprehensive GFM



First pretrain on graph properties

Then adopt structure augmented transformer

Training a Structural GFM – More Details

- We begin by calculating a ground truth graph properties vector $\boldsymbol{p}$. The goal is to train a structural GFM $f(\cdot;\Theta)$ implemented using graph transformers with parameters Θ to predict this vector.
- We do not use node features $\boldsymbol{X}$ during training. We feed the positional encoding matrix $\boldsymbol{B}$ directly into $f(\cdot;\Theta)$, which generates a structural representation $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$.

$$\hat{\boldsymbol{p}}_{\Theta,\Psi} = \varphi(\boldsymbol{Z}_\Theta;\Psi), \boldsymbol{Z}_\Theta = f(\boldsymbol{B};\Theta)$$

- During the training process, we optimize the parameters Θ and Ψ by solving the minimization problem:

$$\Theta^*, \Psi^* = \arg\min_{\Theta,\Psi} \ell_{prop}(\hat{\boldsymbol{p}}_{\Theta,\Psi}, \boldsymbol{p})$$

Training a Comprehensive GFM – More Details

- Given the trained structural GFM f with parameters Θ^*, we compute the positional encoding $\boldsymbol{B}$ for each graph G to obtain the structural representation $\boldsymbol{Z}$

$$\boldsymbol{Z} = f(\boldsymbol{B}; \Theta^*)$$

- Let $\boldsymbol{E} = [e_1, \ldots, e_n]^T$ be the unified node features. We can create an augmented feature matrix $\widehat{\boldsymbol{X}}$ by combining the unified node features $\boldsymbol{e}_i$ with the corresponding structural representation $\boldsymbol{z}_i$

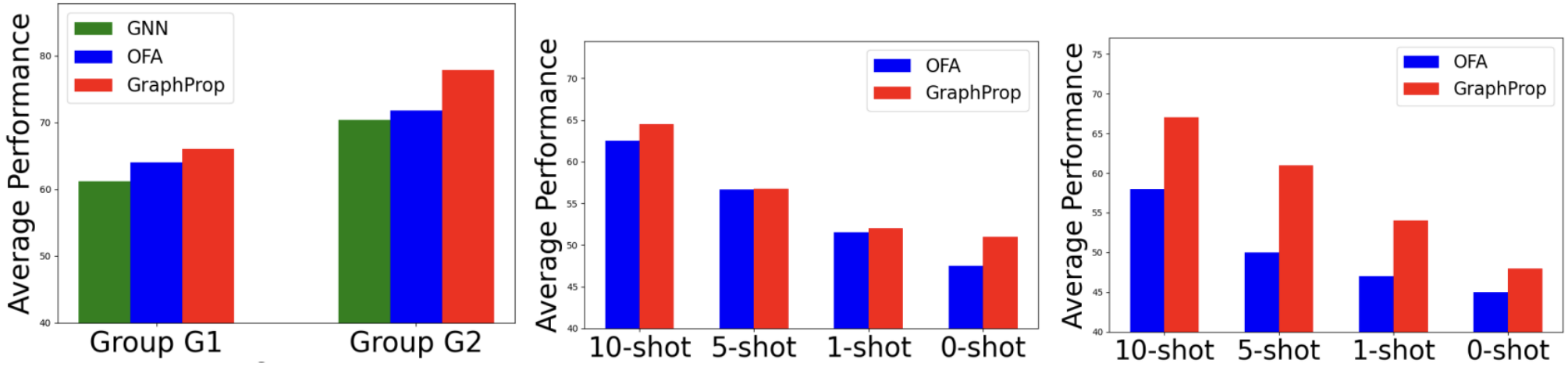
$$\hat{\boldsymbol{x}}_i = \boldsymbol{e}_i \oplus \boldsymbol{z}_i, \forall i \in [n]$$

- Next, we train the comprehensive GFM $F(\cdot; \Phi)$ with trainable parameters Φ by minimizing the cross-entropy loss ℓ_{ce} for classification.

$$\Phi^* = \text{argmin}_\Phi \ell_{ce}(\hat{y}_\Phi, y), \hat{y}_\Phi = F(\widehat{\boldsymbol{X}}; \Phi)$$

Experiments and Findings

- **Highlight**: GraphProp achieves both node feature and structural cross-domain generalization, while previous in-context learning methods primarily focus on node feature generalization and may struggle with datasets lacking node features.



(a) Supervised Learning (b) Few-shot Learning on $\mathbb{G}_1$ (c) Few-shot Learning on $\mathbb{G}_2$

Beyond Small GNNs: Graph Foundation Models

LLM-Based Methods

LLM-Based Methods

- **LLM could play 2 roles** to facilitate general graph models.
  - ***LLM as predictors***: flatten graphs into text representations and feed them into LLM -> LLMs output predictions
  - ***LLM as enhancers***: Convert the text features of graphs to unified representations -> feed to downstream GNN models
  - A LLM-based GFM could leverage both roles

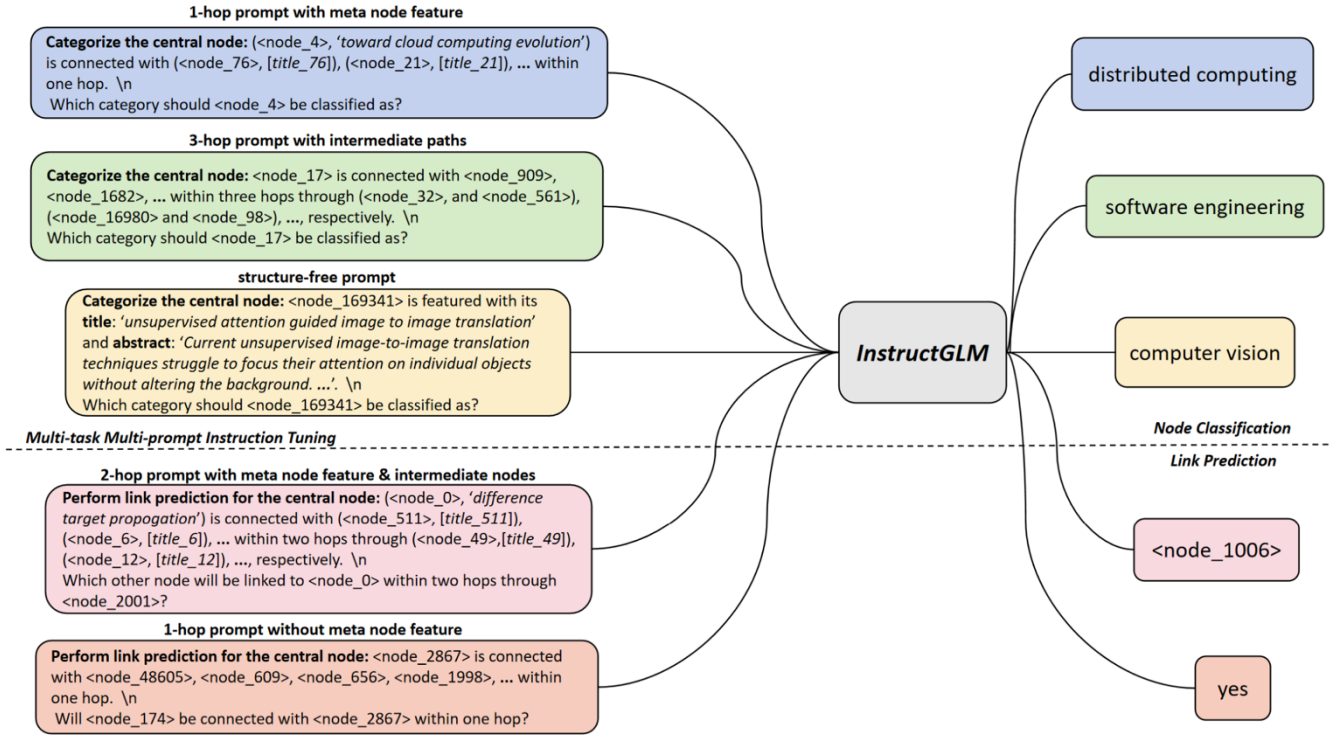
LLM-Based Methods

- We discuss three new papers from this perspective.
  - Language is All a Graph Needs **[LLM predictor]**
  - LLaGA: Large Language and Graph Assistant **[LLM predictor]**
  - One for All: Towards Training One Graph Model for All Classification Tasks **[LLM enhancer]**
  - GOFA: A Generative One-For-All Model for Joint Graph Language Modeling.

  [LLM enhancer+predictor]

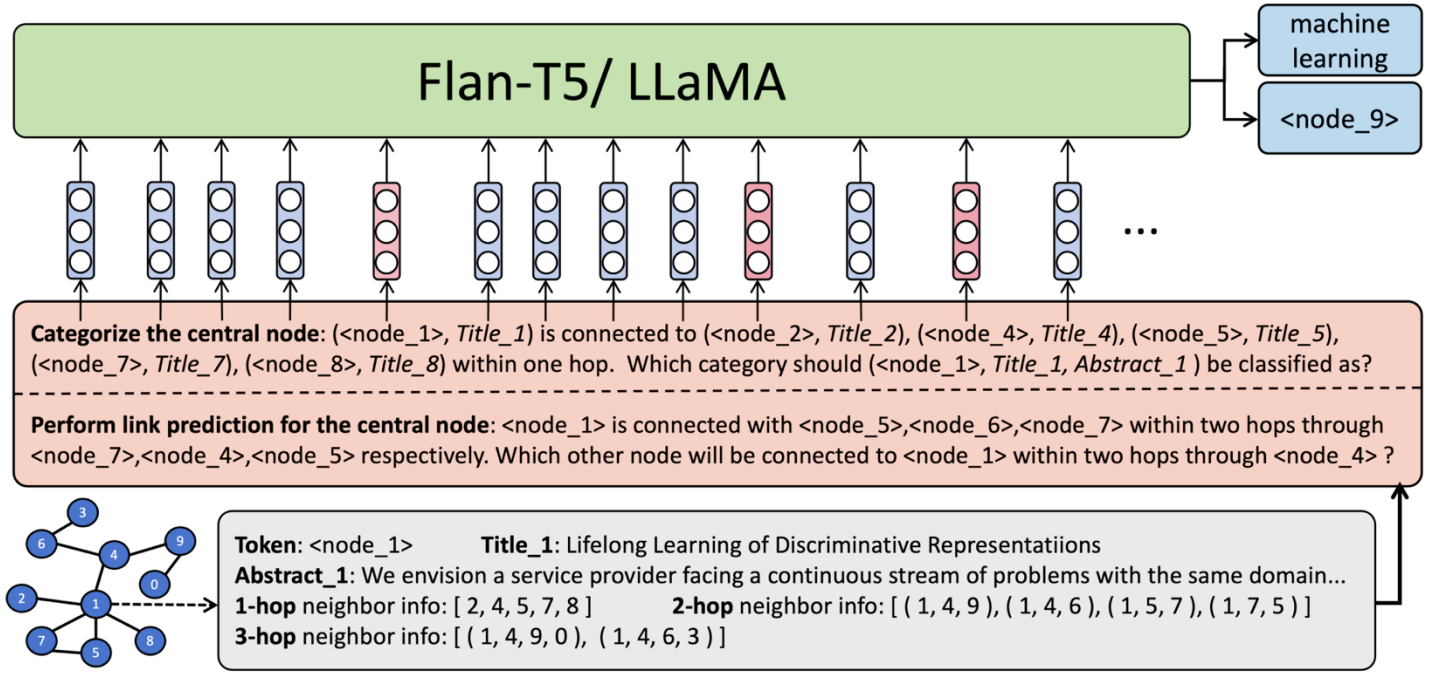
Language is All a Graph Needs – LLM Predictor

- **Key Idea**: Use natural language to describe multi-scale geometric structure of graphs and then instruction finetune LLMs to perform graph tasks.



Instruction Finetuning

- For node classification, we might design:
  - $\mathcal{P}$ = ‘Classify the central node into one of the following categories: [<All category>]. Pay attention to the multi-hop link relationships between the nodes.’
  - $\mathcal{Q}$ = ‘Which category should $\{v\}$ be classified as?’



Instruction Prompt Design

- Denote an instruction prompt as $\mathcal{T}(\cdot)$ such that $\mathcal{I} = \mathcal{T}(v, \mathcal{A}, \{\mathcal{N}_v\}_{v \in V}, \{\mathcal{E}_e\}_{e \in E})$ is the input to LLM and v is the central node of this prompt.
- Considerations:
  - (i) what is the largest hop level of neighbor information about the central node in the prompt?
  - (ii) does the prompt include meta node features or edge features?
  - (iii) for prompts with large hop level neighbors about the central node, does the prompt encompass information about the intermediate nodes or paths along the corresponding connecting route?

Instruction Finetuning

- Given $\mathcal{G} = (V, \mathcal{A}, E, \{\mathcal{N}_v\}_{v \in V}, \{\mathcal{E}_e\}_{e \in \mathrm{E}})$ and a specific instruction prompt $\mathcal{T}$, we denote $\boldsymbol{x}$ and $\boldsymbol{y}$ as the LLM's input and output, respectively. The instruction finetuning pipeline can be formed as:

$$P_\theta\left(\boldsymbol{y}_j \middle| \boldsymbol{x}, \boldsymbol{y}_{<j}\right) = LLM_\theta(\boldsymbol{x}, \boldsymbol{y}_{<j})$$

$$\boldsymbol{x} = \mathrm{CONCAT}(\mathcal{P}; \mathcal{I}; \mathcal{Q})$$

$$\mathcal{L}_\theta = -\sum_{j=1}^{|\boldsymbol{y}|} \log P_\theta\left(\boldsymbol{y}_j \middle| \boldsymbol{x}, \boldsymbol{y}_{<j}\right)$$

- $\mathcal{L}$ denotes the NLL loss, $\mathcal{P}$ and $\mathcal{Q}$ are the task-specific instruction prefix and query.

Experiments and Findings

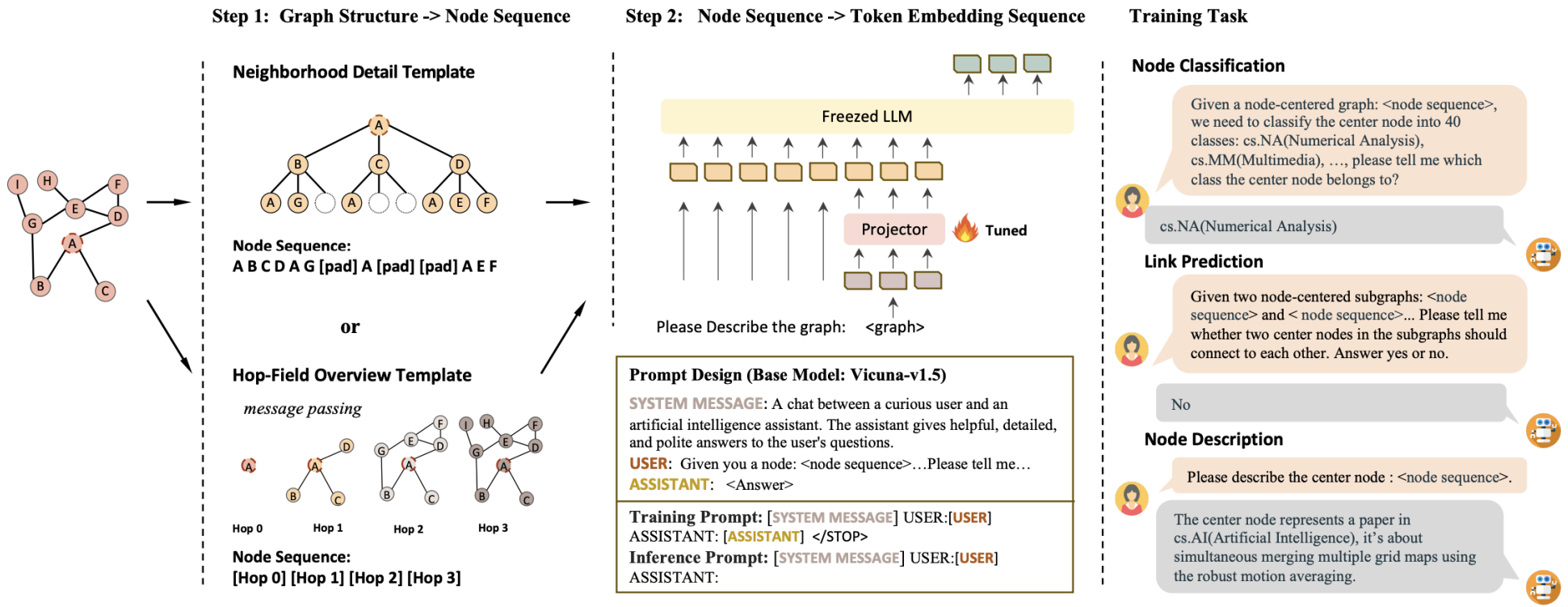
- **Findings**: The proposed method achieves single-model state-of-the-art performance, surpassing all single graph learners across all three datasets, including both representative GNN models and graph Transformer models.

Method	OGB	GIANT
MLP	55.50 ± 0.23	73.06 ± 0.11
GAMLP	56.53 ± 0.16	73.35 ± 0.08
GraphSAGE	71.19 ± 0.21	74.35 ± 0.14
GCN	71.74 ± 0.29	73.29 ± 0.01
DeeperGCN	71.92 ± 0.16	–
ALT-OPT	72.76 ± 0.00	–
UniMP	73.11 ± 0.20	–
LEGNN	73.37 ± 0.07	–
GAT	73.66 ± 0.11	74.15 ± 0.05
AGDN	73.75 ± 0.21	76.02 ± 0.16
RvGAT	74.02 ± 0.18	75.90 ± 0.19
DRGAT	74.16 ± 0.07	76.11 ± 0.09
CoarFormer	71.66 ± 0.24	–
SGFormer	72.63 ± 0.13	–
Graphormer	72.81 ± 0.23	–
E2EG	73.62 ± 0.14	–
Flan-T5-base	73.51 ± 0.16	74.45 ± 0.11
Flan-T5-large	74.67 ± 0.08	74.80 ± 0.18
Llama-7b	**75.70 ± 0.12**	**76.42 ± 0.09**

Method	Cora	PubMed
MixHop	75.65 ± 1.31	90.04 ± 1.41
GAT	76.70 ± 0.42	83.28 ± 0.12
Geom-GCN	85.27 ± 1.48	90.05 ± 0.14
SGC-v2	85.48 ± 1.48	85.36 ± 0.52
GraphSAGE	86.58 ± 0.26	86.85 ± 0.11
GCN	87.78 ± 0.96	88.90 ± 0.32
BernNet	88.52 ± 0.95	88.48 ± 0.41
FAGCN	88.85 ± 1.36	89.98 ± 0.54
GCNII	88.93 ± 1.37	89.80 ± 0.30
RevGAT	89.11 ± 0.00	88.50 ± 0.05
Snowball-V3	89.59 ± 1.58	91.44 ± 0.59
ACM-GCN+	89.75 ± 1.16	90.96 ± 0.62
Graphormer	80.41 ± 0.30	88.24 ± 1.50
GT	86.42 ± 0.82	88.75 ± 0.16
CoarFormer	88.69 ± 0.82	89.75 ± 0.31
Llama-7b	87.08 ± 0.32	93.84 ± 0.25
Flan-T5-base	**90.77 ± 0.52**	94.45 ± 0.12
Flan-T5-large	88.93 ± 1.06	**94.62 ± 0.13**

LLaGA – LLM Predictor

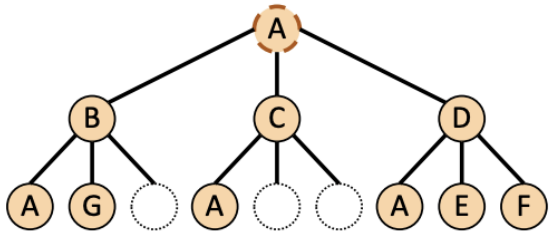
- **Key Idea**: LLaGA reorganizes graph nodes to structure-aware sequences and then mapping these into the token embedding space for LLMs.



Structure-Aware Graph Translation

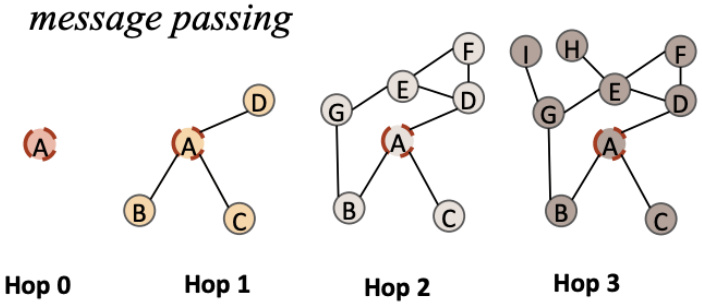
- The key step of LLaGA is to translate **graph inputs** into a **token embedding space** that is comprehensible to LLMs, enabling the utilization of LLMs' inherent reasoning capabilities for graph-related tasks.
- They developed two **node-level templates** for analysis on graphs.

Neighborhood Detail Template



Node Sequence:
A B C D A G [pad] A [pad] [pad] A E F

Hop-Field Overview Template



Node Sequence:
[Hop 0] [Hop 1] [Hop 2] [Hop 3]

Structure-Aware Graph Translation

- Alignment of the **node embedding space** with the **input token space** is realized by mapping each node embedding into the token embedding space, utilizing a specifically calibrated projector.

$$e_i = f_\theta(h_i)$$

Token Embeddings

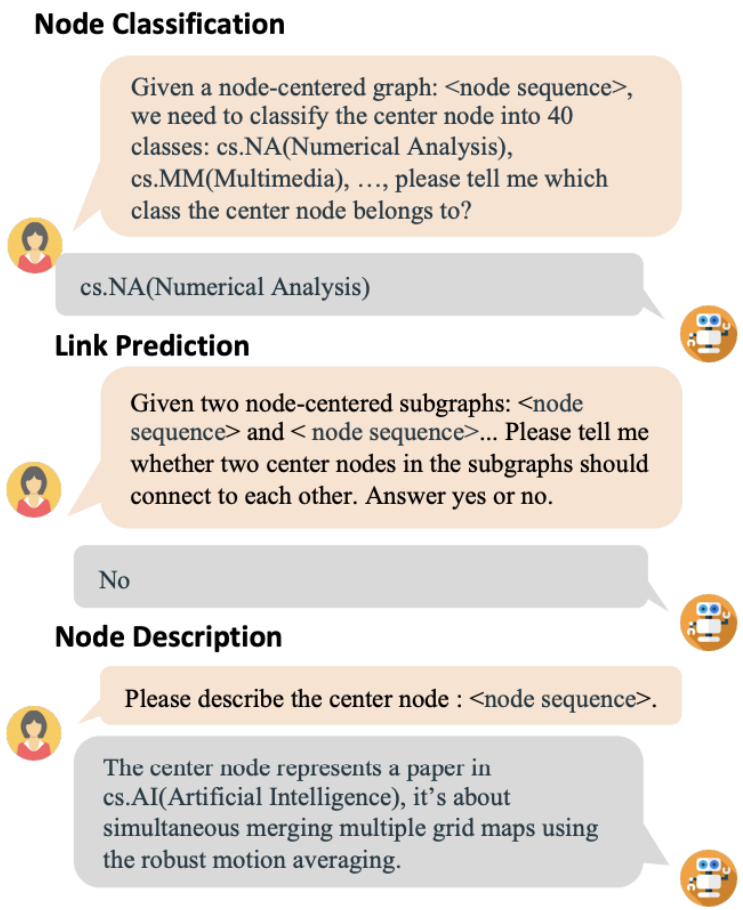
Calibrated Projector

Node Embeddings

Alignment Tuning

- LLaGA considers three tasks on graphs – node classification, link prediction, and node description – to meticulously tune the projector.
- LLaGA trains all tasks in a uniform **Question-Answer** format, eschewing the need for task-specific loss functions or heads. The training objective is formulated as:

$$\max_{\theta} p(X_{answer} | X_{graph}, X_{question}, X_{system})$$



Experiments and Findings

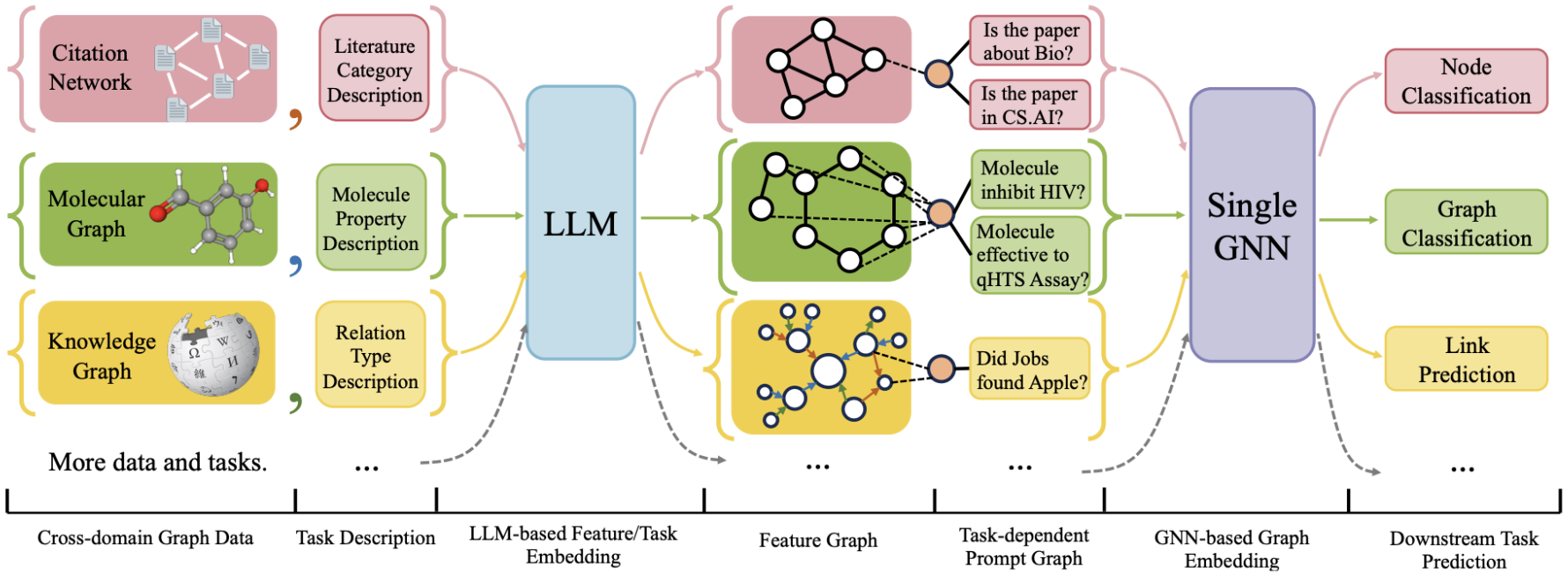
- **Findings**: The performance of LLaGA can exceed that of specialized task-focused graph models. LLaGA also demonstrates robust generalization to previously unseen datasets and tasks without additional tuning.

Model Type	Model	Node Classification Accuracy(%)				Link Prediction Accuracy(%)			
		Arxiv	Products	Pubmed	Cora	Arxiv	Products	Pubmed	Cora
Single Focus	GCN	73.72	80.75	92.96	88.93	91.43	93.95	90.91	81.59
	GraphSAGE	76.29	82.87	94.87	88.89	91.64	94.96	90.64	79.15
	GAT	74.06	83.06	92.33	88.97	85.99	93.85	83.96	80.06
	SGC	71.77	75.47	87.35	87.97	87.99	88.51	83.60	80.94
	SAGN	75.70	82.58	**95.17**	89.19	90.62	94.85	90.48	79.88
	NodeFormer	74.85	83.72	94.90	88.23	91.84	90.93	77.69	77.26
	LLaGA-ND-7B	75.98	84.60	95.03	88.86	91.24	**97.36**	**91.41**	83.79
	LLaGA-HO-7B	**76.66**	**84.67**	95.03	**89.22**	**94.15**	95.56	89.18	**86.82**
Task Expert	GCN	71.45	80.88	89.25	81.62	88.51	93.54	81.01	78.88
	GraphSAGE	72.56	82.50	94.15	81.99	87.76	93.49	76.14	80.74
	GAT	72.19	82.61	87.97	83.58	82.58	92.03	76.85	79.76
	NodeFormer	72.35	82.99	94.41	83.27	84.11	93.42	80.40	81.03
	LLaGA-ND-7B	**76.41**	**84.60**	94.78	88.19	91.20	**97.38**	**93.27**	**89.41**
	LLaGA-HO-7B	76.40	84.18	**95.06**	**89.85**	**94.36**	95.85	88.88	87.50
Classification Expert	GCN	70.95	80.02	89.00	82.77	89.67	93.02	78.79	79.82
	GraphSAGE	71.91	81.62	91.81	82.44	89.23	92.22	75.36	82.09
	GAT	70.90	81.83	87.72	82.07	85.18	92.11	75.00	80.35
	NodeFormer	63.20	75.55	89.50	69.19	82.33	75.42	78.22	81.47
	LLaGA-ND-7B	75.85	**83.58**	**95.06**	87.64	90.81	**96.56**	**92.36**	87.35
	LLaGA-HO-7B	**75.99**	83.32	94.80	**89.30**	**94.30**	96.05	88.64	**88.53**
General Model	GPT3.5-Turbo	55.00	75.25	88.00	71.75	63.80	60.30	68.70	65.74
	LLaGA-ND-7B	74.29	**82.21**	92.42	**87.82**	90.53	**96.82**	86.31	81.91
	LLaGA-HO-7B	**75.01**	82.07	94.45	**87.82**	**92.04**	86.80	**89.81**	**84.41**

Liu et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR, 2024.

One for All – LLM Enhancer

- **Key Idea**: use text-attributed graphs to unify different graph data by describing nodes and edges with natural language and uses language models to encode the diverse and possibly cross-domain text attributes to feature vectors in the same embedding space.



Liu et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR, 2024.

Unifying Graph Data with Tags

- Cross-domain graph data are usually generated by entirely different procedures and have attributes embedded in different spaces.
- However, despite the distinct attributes across datasets, almost all can be described by human-interpretable language. Then we can apply an LLM to encode different graph attributes into the same space!

Text feature of nodes: Feature node. <*feature description*>: <*feature content*>; <*feature description*>: <*feature content*>; ...
Example: Feature node. Atom: Carbon, Atomic number 6, helix chirality, is not in a ring, ...
Example: Feature node. Paper title and abstract: Attention is all you need. The dominant sequence transduction models are ...

Text feature of edges: Feature edge. <*feature description*>: <*feature content*>; <*feature description*>: <*feature content*>; ...
Example: Feature edge. Chemical Bond: ionic bonding, is conjugated, ...
Example: Feature edge. Citation from one paper to another.

Liu et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR, 2024.

Unifying Graph Tasks with Nodes-of-Interest

- Drawing on ideas from downstream tasks in language, can we unify different graph tasks into a single task to facilitate the training and knowledge transferring in the graph domain?
- We propose Nodes-of-Interest (NOI) subgraph and NOI prompt node to achieve the goal.
  - NOI refers to the set of target nodes in a task and is represented as $\mathcal{T}$.
  - An NOI subgraph is defined as the subgraph around the NOI.

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.

Liu et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR, 2024.

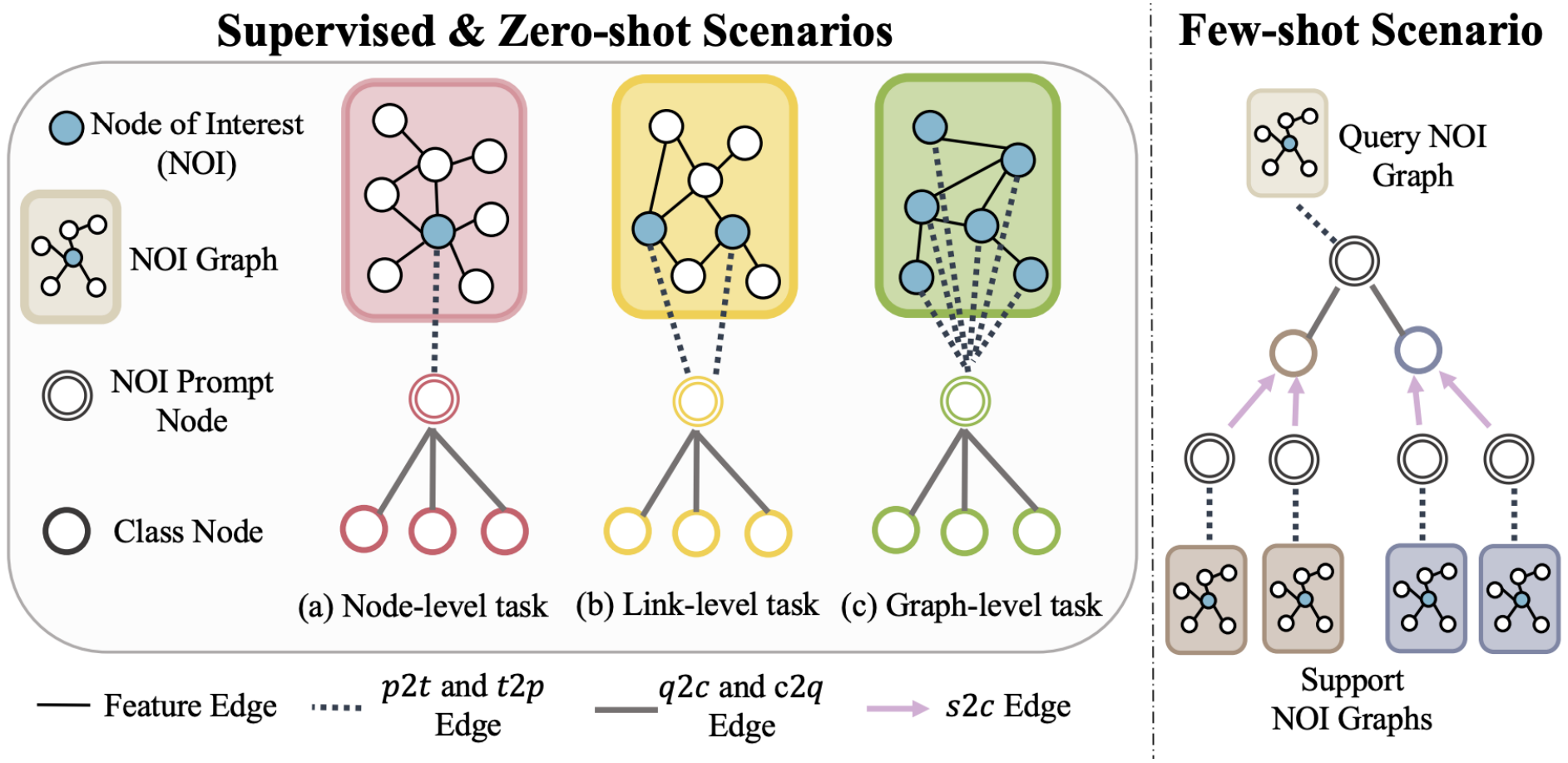
Graph Prompting Paradigm

- The core principle of in-context learning involves manipulating the input data to align it with downstream tasks.
- We propose the Graph Prompting Paradigm (GPP) to manipulate the input graph so that the graph model can acquire task-relevant information from the input itself.

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, 2024.

Graph Prompting Paradigm



Experiments and Findings

- **Findings**: OFA successfully enabled a single graph model to be effective on all graph datasets across different domains as the joint version with all different LLMs performs well on all datasets.

Task type Metric	Cora Link AUC ↑	Cora[1] Node Acc ↑	PubMed Link AUC ↑	PubMed[1] Node Acc ↑	ogbn-arxiv[1] Node Acc ↑	Wiki-CS Node Acc ↑	HIV Graph AUC ↑
GCN	90.40±0.20	78.86±1.48	91.10±0.50	74.49±0.99	74.09±0.17	79.07±0.10	75.49±1.63
GAT	93.70±0.10	**82.76±0.79**	91.20±0.10	75.24±0.44	74.07±0.10	**79.63±0.10**	74.45±1.53
OFA-ind-st	91.87±1.03	75.61±0.87	98.50±0.06	73.87±0.88	75.79±0.11	77.72±0.65	73.42±1.14
OFA-st	94.04±0.49	75.90±1.26	98.21±0.02	75.54±0.05	75.54±0.11	78.34±0.35	78.02±0.17
OFA-e5	92.83±0.38	72.20±3.24	98.45±0.05	77.91±1.44	75.88±0.17	73.02±1.06	**78.29±1.48**
OFA-llama2-7b	94.22±0.48	73.21±0.73	**98.69±0.10**	77.80±2.60	77.48±0.17	77.75±0.74	74.45±3.55
OFA-llama2-13b	**94.53±0.51**	74.76±1.22	98.59±0.10	**78.25±0.71**	**77.51±0.17**	77.65±0.22	76.71±1.19

Task type Metric	WN18RR Link Acc ↑	FB15K237 Link Acc ↑	PCBA Graph APR ↑
GCN	67.40±2.40	74.20±1.10	20.20±0.24
GIN	57.30±3.40	70.70±1.80	22.66±0.28
OFA-ind-st	97.22±0.18	**95.77±0.01**	22.73±0.32
OFA-st	96.91±0.11	95.54±0.06	24.83±0.10
OFA-e5	97.84±0.35	95.27±0.28	**25.19±0.33**
OFA-llama2-7b	98.08±0.16	95.56±0.05	21.35±0.94
OFA-llama2-13b	**98.14±0.25**	95.69±0.07	21.54±1.25

Results on supervised learning

Experiments and Findings

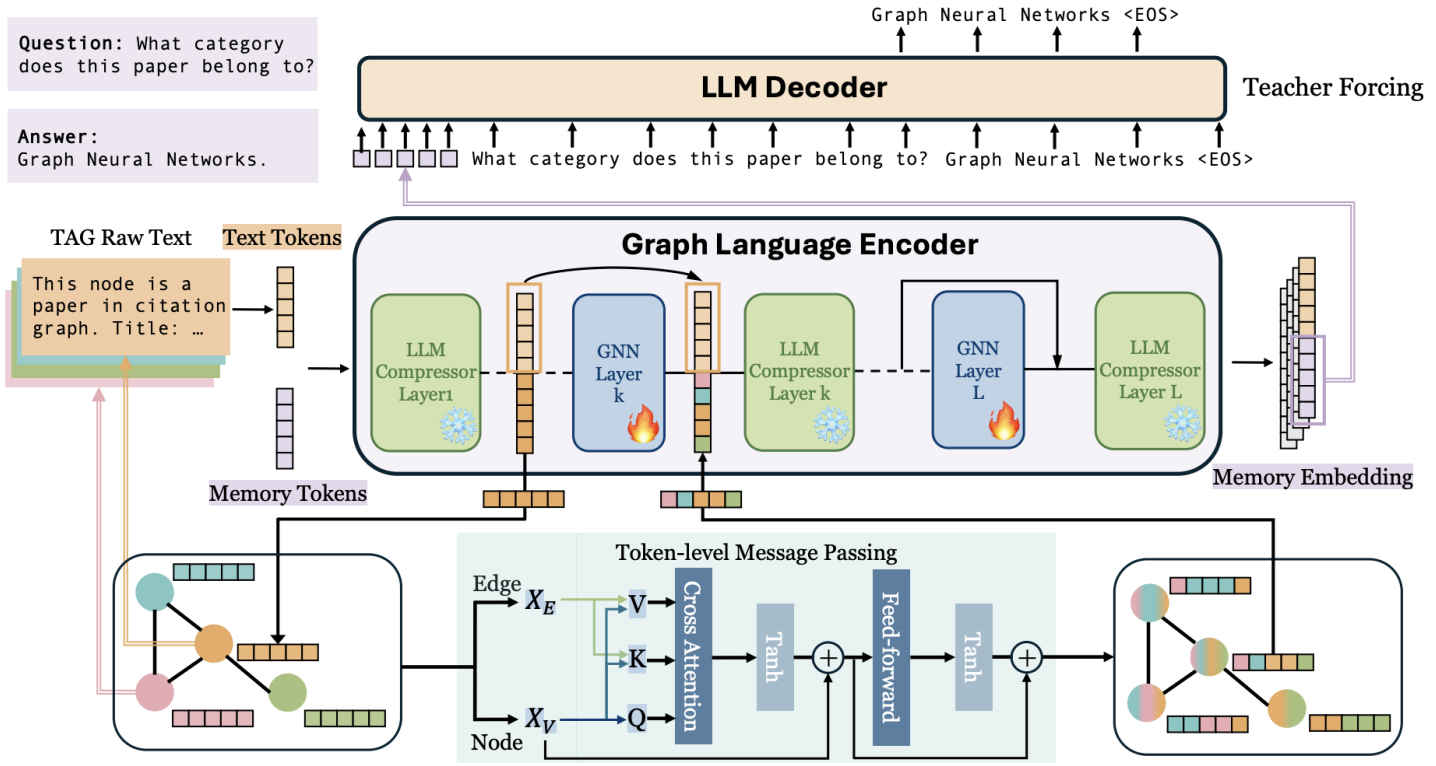
- **Findings**: OFA successfully enabled a single graph model to be effective on all graph datasets across different domains as the joint version with all different LLMs performs well on all datasets.

# Way	ogbn-arxiv-5-way (Transductive)				Cora-2-way (Transfer)		
Task	5-shot	3-shot	1-shot	0-shot	5-shot	1-shot	0-shot
GPN	50.53±3.07	48.32±3.80	38.58±1.61	-	63.83±2.86	56.09±2.08	-
TENT	60.83±7.45	56.03±8.90	45.62±10.70	-	58.97±2.40	54.33±2.10	-
GLITTER	56.00±4.40	57.44±4.90	47.12±2.73	-	-	-	-
TLP-BGRL	50.13±8.78	46.21±7.92	35.81±8.58	-	81.31±1.89	59.16±2.48	-
TLP-SURGL	77.89±6.46	74.19±7.55	61.75±10.07	-	92.49±1.02	81.52±2.09	-
Prodigy	61.09±5.85	58.64±5.84	48.23±6.18	-	-	-	-
OFA-joint-lr	61.45±2.56	59.78±2.51	50.20±4.27	46.19±3.83	76.10±4.41	67.44±4.47	56.92±3.09

Few-shot and Zero-shot results (part)

Generative One-For-All Model for Joint Graph Language Modeling

- **Key Idea**: **integrating GNN layers with LLM layers** to combine the generative capabilities of LLMs for free-form output with the structural learning strengths of GNNs for understanding complex connections.



Generative Modeling for Graph

- Defining a unified input and output format
  - For unified input, any node and edge features can be represented by texts. We can represent non-textual features like pure numbers as texts too.
  - Formally, a TAG is a graph $G = \{V, E, X_V, X_E\}$ where V and E are the sets of nodes and edges. Each node $v \in V$ (edge $e \in E$) corresponds to a text description $x(v) \in X_V$ ($x(e) \in X_E$).
  - For unified output, natural language is still the most tangible format for users.

Generative Modeling for Graph

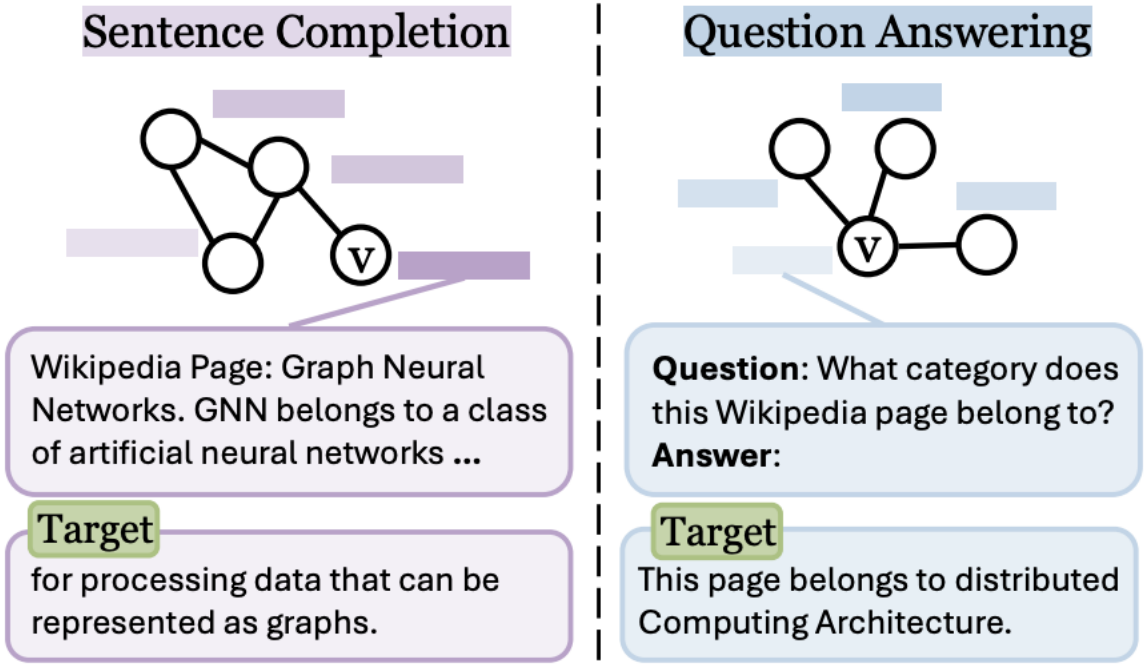
- Defining the generative modeling framework
  - To generalize next token prediction to graphs, we propose to specify Nodes of Generation on graphs as starting points for output generation.
  - Formally, we can define graph generative modeling as the likelihood of the text y associated with the NOG v:

$$p(y|v,G) = \prod_{l=1}^{L} p(y_l|y_{<l}, v, G)$$

  - where y_l is the l-th token of y, and $y_{<l}$ is its preceding tokens.

Generative Modeling for Graph

- Defining the generative modeling framework
  - The NOG v is a learning target with initial corresponding text $x(v)$, and $x(v)$ can be empty. G contains structural and textual information of neighbor nodes to help the model generate y.



Generative One-For-All Model

- Designing the GOFA architecture
  - **Graph language encoder** interleaves *token-level GNN* layers with frozen *LLM compressor* layers to get the final *multi-token node representations* containing joint structural and semantic information.
  - **LLM decoder** generates texts from the NOG representation.

GOFA: Generative One-For-All Model

- Designing the GOFA architecture
  - *LLM compressor:* A sentence compressor that preserves as much information as possible from the original sentence in fixed-size multi-token embeddings, which has the same architecture as a decoder-only LLM.

$$\{Q_x^{t+1}, Q_{m,x}^{t+1}\} = LLM^t(\{Q_x^t, H_x^t\})$$

  - *Token-level GNN:* A simple extension of conventional GNNs to token level.

$$H_{x(v)}^t[k] = GNN(Q_{m,x(v)}^t[k], \{(Q_{m,x(u)}^t[k], Q_{m,x(e_{uv})}^t[k]) | u \in \mathcal{N}(v)\})$$

GOFA: Generative One-For-All Model

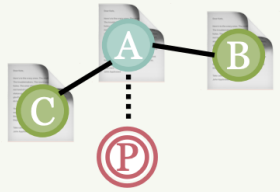
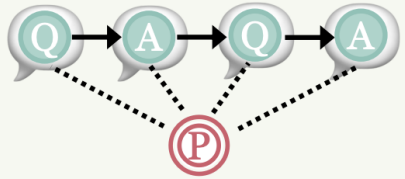
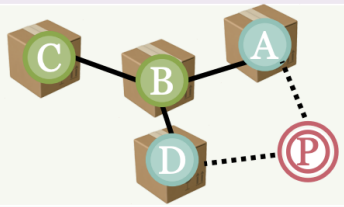
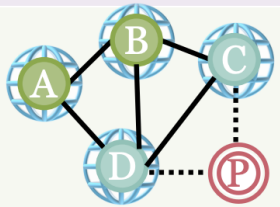
- Designing the GOFA architecture
  - *LLM decoder:* for the NOG v and its corresponding target text y, we insert the memory tokens $Q_{m,x}$ at the front of the token embeddings of the target text to generate and user teacher-forcing to maximize the standard log-likelihood for the next-token-prediction objective.

$$\mathcal{L} = \max_{Q_{m,x(v)}} P(y_1 \dots y_l | Q_{m,x(v)}; \Theta_{\mathrm{DEC}})$$

- Unifying Task Representations
  - We convert all tasks into subgraph tasks and connect all target nodes in graph to a virtual prompt node as the NOG for generation.

Experiments and Findings

- We pre-train the model with large-scale real-world graph data. After pre-training, we further fine-tune the model on downstream tasks.

Task	Sentence Completion	QA-Chain Task	Shortest Path	Common Neighbors
TAG				
TAG Raw Text	A: This is [Node A]. Title: Graph Attention Networks. Abstract: We present graph attention ... B: This is [Node B]. Title: Attention is all you need. Abstract: The dominant sequence transduction models ... C: This is [Node C]. Title: Adam: A method for stochastic optimization. Abstract: We introduce Adam, an algorithm for ...	Q: Which type of Rock is commonly used for construction and why? A: Sedimentary rock. It is easy to extract, cut, and shape. Q: Are there any other types of rocks used for construction? A: Yes. Igneous rocks like granite are used for their durability.	A: This is [Node A]. Product: Wireless Controller for Switch or OLED... D: This is [Node D]. Product: Amazon Fire TV, 4-series 4K UHD smart TV... B: This is [Node B]. Product: Nintendo Switch with Blue and Red Joy-Con...	C: This is [Node C]. Wikipedia entry: system_7. Seventh major release of ... D: This is [Node D]. Wikipedia entry: quickdraw. A graphics software ... B: This is [Node B]. Wikipedia entry: unix. Unix is a family of multitasking...
Prompt	P: **Please complete the sentence** on [Node A].	P: **Do certain regions or cultures have preference of rocks?**	P: **Compute the shortest path** between [Node A] and [Node D] and **generate all shortest paths** from [Node A] to [Node D].	P: Is there any **common neighbors** between [Node C] and [Node D]? If exist, please **give the total number** and **list all common neighbors**.
Answer	**networks (GATs), novel neural network architectures that operate on graph-structured data.**	Yes, **limestone is commonly used in UK** because it can withstand high levels of rainfall and humidity.	The **shortest path distance is 2. Shortest path:** [Node A] -> [Node B] -> [Node D] .	There is **1 common neighbor** between these two nodes: [Node B].

Experiments and Findings

- GOFA when finetuned with a small number of data, achieves impressive zero-shot performance, highlighting its potential as a robust graph foundation model.

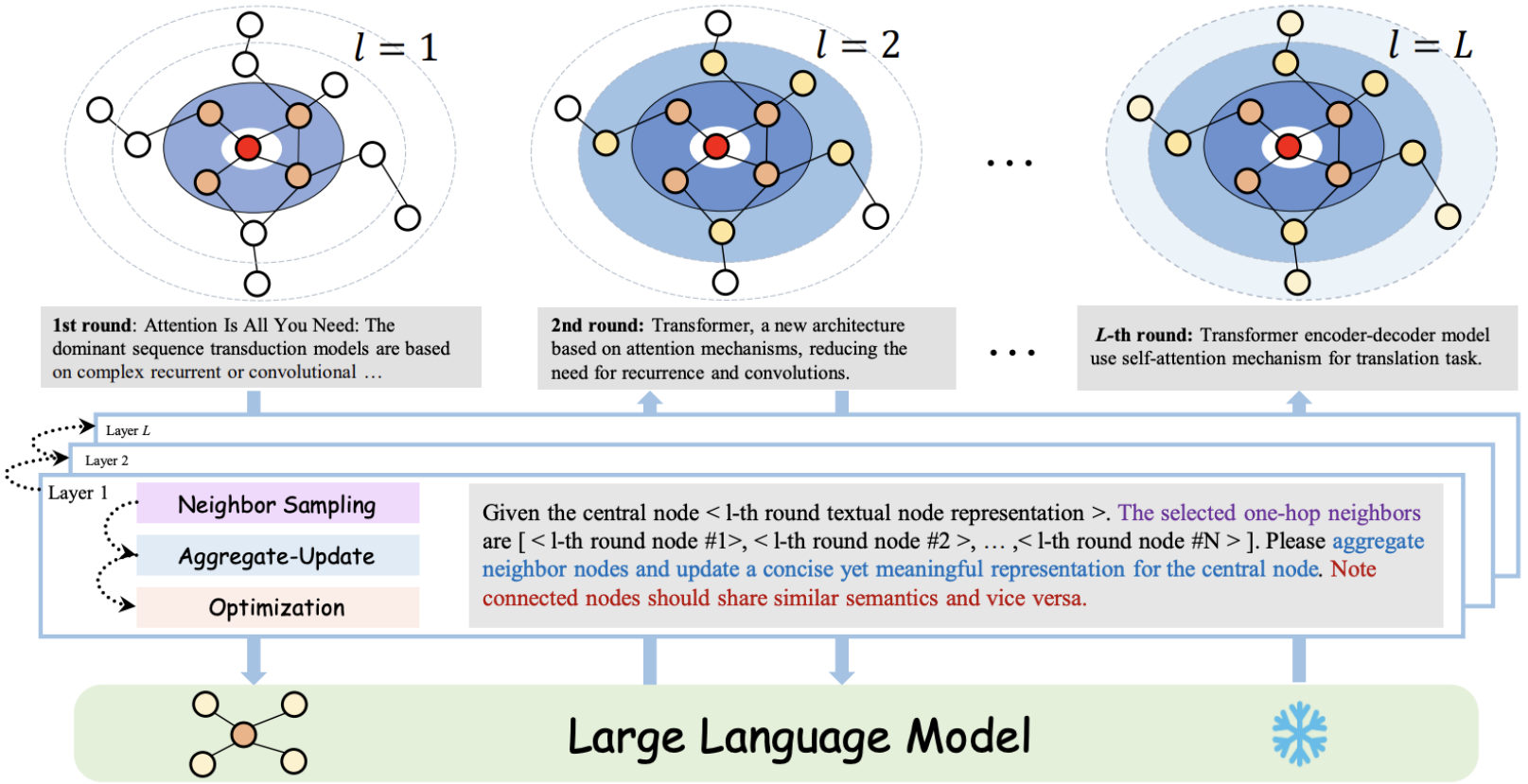
Task Way	Cora-Node 7	PubMed-Node 3	WikiCS 10	Products 10	ExplaGraphs 2	FB15K237 10	SceneGraphs QA
Llama2-7B	29.69	60.95	32.56	50.69	59.02	27.66	38.62
Mistral-7B	54.79	71.02	58.83	61.99	73.03	**63.85**	**45.95**
OFA-Llama2	27.70	56.42	18.5	-	-	-	-
GraphGPT	18.13	70.11	-	-	-	-	-
UniGraph	69.53	72.48	43.45	66.07	-	-	-
GOFA-L2-arxiv-10K	63.56	65.26	47.98	36.85	53.61	28.47	23.52
GOFA-M-arxiv-10K	65.15	64.37	68.19	72.60	78.21	45.81	32.44
GOFA-M-arxiv-40K	**71.20**	**73.11**	**70.49**	**75.83**	**79.56**	55.96	33.06

Graph Vocabulary Learning for Graph Foundation Model

- **Key Idea**: building a versatile GFM grounded in graph vocabulary learning.
  - (1) Expressiveness: the vocabulary encapsulates semantic and structural knowledge across various domain-specific graphs.
  - (2) Transferability: each node in any graph is represented by one or more fundamental units within this vocabulary.
  - (3) Scalability: the vocabulary is sufficiently inclusive to accommodate unseen nodes, even those from outside existing graphs. S

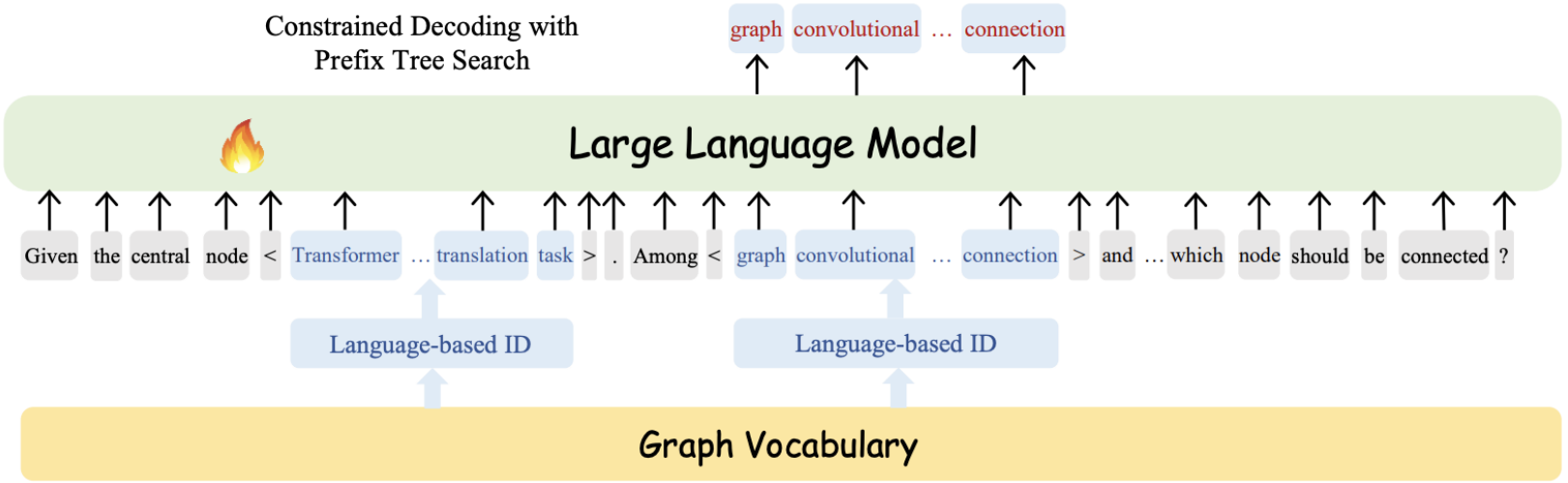
Graph Understanding Module

- To function LLMs as GNNs, we prompt LLMs to explicitly replicate the core principles of GNNs within the language space.



Graph Inference Module

- We propose decoupling these textual representations to establish a universal graph vocabulary, where each node is mapped to a finite sequence of tokens, essentially as language-based IDs.



Experiments and Findings

- Extensive experiments demonstrate the superiority of PromptGFM in node classification and link prediction, along with its strong transferability across different datasets and tasks.

Model	Cora	Citeseer	PubMed	ogbn-arxiv
MF	60.07±2.69	62.33±2.08	59.82±1.42	68.47±0.92
MLP	62.29±4.69	64.42±2.43	62.88±1.88	62.07±0.35
GCN	82.47±3.89	76.11±3.34	77.36±1.07	66.15±0.46
GAT	82.92±2.58	77.30±2.57	74.36±3.48	65.29±0.64
SAGE	83.69±2.43	73.17±3.83	83.22±1.86	68.78±0.77
RevGNN	86.90±1.72	77.34±2.59	82.16±2.27	70.43±0.38
AGNN	77.64±2.55	73.14±1.93	73.55±0.62	60.63±0.42
DNA	80.81±3.38	73.64±2.98	80.68±1.33	58.51±0.67
SGFormer	82.36±2.88	73.76±3.03	78.92±1.63	63.44±0.95
NodeFormer	81.55±3.01	72.98±2.10	76.49±1.91	73.21±0.41
OFA	79.41	81.35	N/A	73.75
LLaGA	81.25	68.80	N/A	76.05
ENGINE	91.48	78.46	N/A	76.02
GraphPrompter	80.26	73.61	**94.80**	79.54
PromptGFM	**91.72±1.06**	**84.49±1.37**	90.67±1.16	**80.58±0.54**

Node Classification Results

Model	Cora	Citeseer	PubMed	ogbn-arxiv
MF	66.43±1.13	70.12±1.99	59.34±1.03	60.26±0.92
MLP	70.11±2.50	74.76±1.80	62.88±1.88	72.64±3.37
GCN	77.15±2.20	78.72±2.11	77.36±1.07	80.89±1.66
GAT	70.44±3.80	77.17±3.30	74.36±3.48	76.25±1.90
SAGE	85.31±3.34	87.15±1.83	83.22±1.86	80.76±2.77
RevGNN	70.13±1.72	78.26±3.08	82.16±2.27	69.67±0.80
AGNN	71.52±2.62	74.23±1.90	73.55±0.62	75.56±0.67
DNA	62.76±1.85	63.96±1.07	58.56±3.36	69.13±2.51
GraphPrompter	90.10	91.67	86.49	73.21
PromptGFM	**90.57±1.26**	**92.03±2.74**	**87.64±1.98**	**89.82±2.54**

Link Prediction Results

Limitations of Existing GFMs

- **LLM as predictors** transforms graph data into representations that LLM can understand and use LLM to generate predictions.
  - However, such an approach falls short of understanding graph structures.
- **LLM as enhancers** adopts LLM to process and unify diverse graph data and feeds them to a GNN to train general graph models.
  - Because GNN outputs fixed-sized representations, they can only handle specific tasks such as classification and cannot generalize to a foundational level due to the lack of generation ability.

Beyond Small GNNs: Graph Foundation Models

Open Questions for Graph Foundation Models

Open-ended Discussion

- What are your expectations on GFM?
- Are current GFM Foundational enough? What things do they miss?
- GNN backbone or LLM backbone?