# Context-aware Graph Meta-learning

**Ningbo Huang, Gang Zhou*, Meng Zhang, Shunhang Li, Ling Wang, Shiyu Wang, Yi Xia**

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China
rylynn_ab@163.com

## Abstract

Developing a universal graph model capable of generalizing across diverse graph domains has consistently been a key objective in graph learning. Recently, many studies have focused on achieving in-context learning (ICL) on graphs, which can generalize to novel tasks without the need for fine-tuning, similar to large language models (LLMs) such as GPT-3. These researches can be primarily divided into graph-based methods and LLM-based methods. However, the generalization performance of the former is limited by the representation capability of GNNs, while the latter faces the challenge of LLMs understanding graph structures. Therefore, we propose **CAGML**, a context-aware graph meta-learning model, which learns to generalize to cross-domain and cross-granularity graph tasks using a meta-trained Transformer. Firstly, we formulate graph few-shot learning tasks as a structure-aware sequence modeling problem to unify cross-domain and cross-granularity tasks. Then, a structure-aware Transformer (SAT) is introduced as a graph in-context learner to make predictions with a few labels and the task-specific structural context. Finally, we pre-train SAT in a meta-optimization manner on large-scale citation network and knowledge graph. Experiments on 6 cross-domain graph datasets show that, without fine-tuning, **CAGML** can achieve state-of-the-art (SOTA) performance in terms of average performance across cross-granularity tasks on adopted datasets.

**Code** — https://github.com/hningbo/CAGML

## 1 Introduction

Graph is a powerful tool to describe and model the complex connections between the entities in the real world such as citation graphs (Yang, Cohen, and Salakhutdinov 2016), product graphs (McAuley, Pandey, and Leskovec 2015; Yan et al. 2023), and knowledge graphs (Mahdisoltani, Biega, and Suchanek 2015; Mernyei and Cangea 2020). To quickly adapt to diverse downstream graph tasks from a small number of samples, many researchers have focused on graph few-shot learning techniques, mainly including graph self-supervised learning and graph meta-learning techniques. The former primarily learns to represent the structural pattern in graph data by constructing various pretext tasks (Liu et al. 2022a), while the latter extracts cross-task structural meta-knowledge by constructing meta-training tasks similar to the target task (Mandal et al. 2022). Despite the significant progress made by these methods in the field of graph few-shot learning, they require fine-tuning for specific tasks, which limits their applicability in real-world scenarios.

In recent years, LLMs such as GPT-3 (Brown et al. 2020) have gained increasing popularity, with in-context learning (ICL) technology emerging as one of the most key and popular capabilities of LLM models. In contrast to the rapidly developing ICL techniques in the text and visual domains (Wu, Wang, and Yao 2025), research on graph ICL is still in its early stages, due to the intrinsic structural complexity of graph data and the significant divergence between graph features. Existing graph ICL methods can be divided into two categories of graph-based and LLM-based methods. The graph-based methods construct prompt graphs to describe the relationships between the structural context of inputs and labels, and make predictions through similarity comparison between data nodes and category nodes. For example, Prodigy (Huang et al. 2023) and OFA (Liu et al. 2024a) respectively use GNN and LLM to unify the feature space of different graphs and construct data nodes and label nodes for various graph tasks to achieve graph ICL. The LLM-based methods serialize graph structure with natural language or codes, and leverage the inherent ICL capabilities of the LLM to perform ICL inference. AskGNN (Hu et al. 2024) injects the textual information of nodes' neighbors into the prompt templates for LLM inference, and learns to retrieve informative structure from the LLM feedback.

However, GNN-based methods are mainly limited by expressive power. Due to the over-smoothing problem, it is difficult to scale to very deep networks, which makes it hard for GNN to efficiently represent cross-domain graph data. For LLM-based methods, the ability of the LLMs to understand graph structure, especially large graphs, remains limited. Additionally, LLMs face a trade-off between the ability to understand structure and the cost of fine-tuning. Fortunately, recent researches on the underlying principles of ICL capabilities in LLMs reveal that the forward propagation of the demonstration samples in Transformer is equivalent to an implicit fine-tuning on the demonstration set (Dai et al. 2023). This insight motivates us to leverage the powerful data representation and ICL capabilities of Transformer to

*Corresponding Author

develop a graph ICL model for cross-domain graph data.

Therefore, we propose a **C**ontext-**A**ware **G**raph **M**eta-**L**earning model (**CAGML**), which enables cross-domain and cross-granularity graph ICL with a meta-trained structure-aware Transformer (SAT). Firstly, we formulate cross-domain and cross-granularity graph few-shot learning tasks as unified structure-aware sequence modeling problems. Then, we design SAT as a graph in-context learner to encode the task-specific structural context of the tasks and make predictions without the need for fine-tuning. Finally, we pre-train the SAT on large-scale graph datasets with meta-optimization. In general, the contributions of this paper can be concluded as follows:

- We study the universal graph ICL problem, aiming to learn a model that can effectively adapt to cross-domain and cross-granularity graph tasks without fine-tuning.
- We propose **CAGML**, a context-aware graph meta-learning model, which learns to generalize to diverse graph few-shot learning tasks with SAT as a graph in-context learner.
- We conduct extensive experiments across 6 cross-domain graph datasets. The results demonstrate that **CAGML** achieves state-of-the-art performance in terms of average accuracy on the selected datasets comparing with the (graph) ICL models.

## 2 Related Work

### 2.1 Graph in-context learning

Graph ICL models are capable of generalizing to diverse graph tasks with a few demonstration samples without the need for fine-tuning, which can be categorized into graph-based and LLM-based methods. Graph-based methods construct prompt graphs consisting of data nodes and label nodes, and make predictions according to the metric comparison between query graph data and the label nodes. PRODIGY (Huang et al. 2023) constructs a prompt graph by connecting the labeled instances to corresponding label nodes. During the inference, the labels of the query set are predicted using the cosine similarity. OFA (Liu et al. 2024a) enables in-context learning with a similar framework and further incorporates LLM as a textual feature encoder to enhance and unify the feature space of cross-domain graphs. ARC (Liu et al. 2024b) concentrates on the graph anomaly detection task, which designs a smoothness-based method to align the features to an anomaly-aware space and infer the results with attentive scores. LLM-based methods serialize the structural context into natural language or code and leverage the inherent ICL ability of LLM to make predictions. AskGNN (Hu et al. 2024) designs a structure aggregator module to sample informative neighbors and serializes the structural information as text prompt for LLM inference.

Graph-based methods rely on GNN to encode data nodes, therefore limited by the representation capability of GNN. While LLM-based methods face a trade-off between the ability to understand structure and the cost of fine-tuning.

### 2.2 Graph meta-learning

Graph meta-learning is based on the task i.d.d. hypothesis, which extracts cross-task structural meta-knowledge from meta-training tasks to better adapt to the target task. *Optimization-based methods* learn to fast adapt to novel classes with several steps of optimization. For example, MAML (Finn, Abbeel, and Levine 2017) and Reptile (Alex, Joshua, and John 2018) respectively use 2-order optimization of meta-tasks or the first-order approximation of the former objectives to learn a model initialization which can fast adapt to the target task. Meta-GNN (Zhou et al. 2019) and G-Meta (Huang and Zitnik 2021) are based on MAML to learn cross-task knowledge from the GNN representations in a bi-optimization manner. AMM-GNN (Wang et al. 2020) and Meta-GPS (Liu et al. 2022b) further consider task divergence by learning task-adaptive transformation for node features and GNN parameters. *Metric-based models* leverage the simple inductive bias of metric learning models to reduce overfitting on very limited labeled data, such as Prototype network, Siamese network, and Relation network (Gharoun et al. 2024). GPN (Ding et al. 2020) extends the prototype network (Snell, Swersky, and Zemel 2017) and further considers the node importance during prototype calculation. Recent studies such as COSMIC (Wang et al. 2023) and CSG-Meta (Huang et al. 2025) leverage graph contrastive learning (GCL) to improve the representation ability of GML models.

*Memory-based methods* are the most closely related to our work, but are also underexplored in graph learning. These methods store the in-context information through a memory module and retrieve the most relevant information during inference without fine-tuning (Gharoun et al. 2024). MANN (Santoro et al. 2016) proposes to use NTM (Alex, Greg, and Ivo 2014) as a memory module to store the (instance, label) pairs information and design an addressing network to retrieve the information helpful for prediction. SANIL (Mishra et al. 2018) uses temporal convolution layers to encode the in-context information and first introduces self-attention to memorize and pinpoint the related information. Furthermore, GPICL (Kirsch et al. 2024) proposes to use Transformer, a pure self-attention architecture. Following GPICL, CAML (Fifty et al. 2024) designs a novel label encoder to achieve the invariance of label assignment.

Despite the effort of existing GML models, these methods require fine-tuning on new tasks and have limited cross-domain and cross-granularity representation capability.

In this paper, inspired by the memory-based meta-learning methods, we propose **CAGML** to address existing problem by designing SAT as a powerful graph in-context learner to achieve cross-domain graph ICL.

## 3 The Proposed Method: CAGML

In this section, we will introduce the details of **CAGML** as illustrated in Figure 1. We firstly design the general framework to formulate the graph few-shot learning tasks as structure-aware sequence prediction problems. And we design SAT as the graph in-context learner, which can generalize to cross-domain and cross-granularity graph few-shot

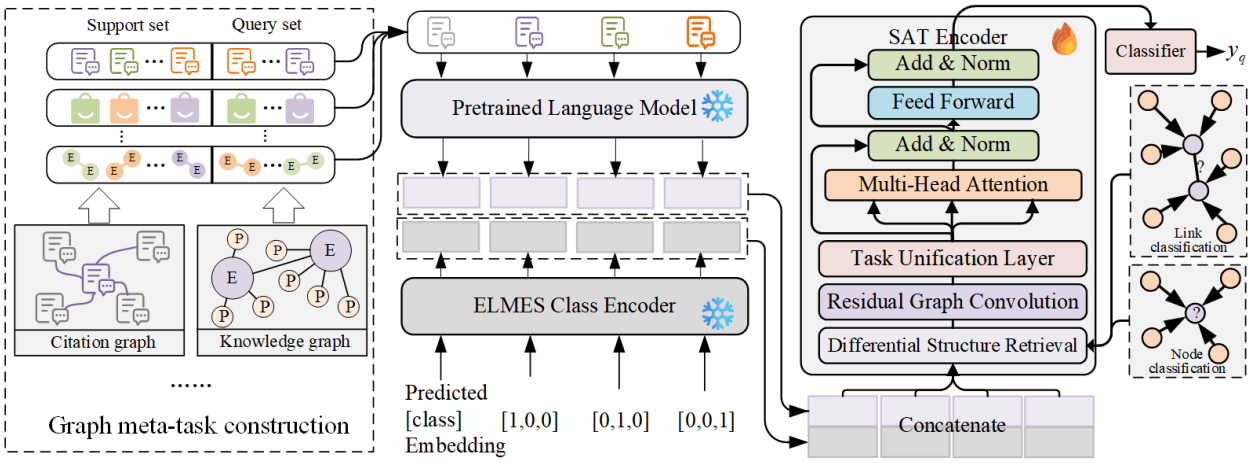


Figure 1: The over framework of our proposed **CAGML** model. We can construct cross-domain and cross-granularity meta-learning tasks from graph data such as citation networks and knowledge graphs. Here, we use a node classification task constructed from citation network as an example to illustrate the inference phase of **CAGML**.

learning tasks without the need for fine-tuning. Finally, the SAT is pre-trianed with large-scale meta-optimization.

### 3.1 Preliminary

**Graph few-shot learning**. Given an attributed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$ and the labels vector $y_{\text{n}}$ for node classification or $y_{\text{l}}$ for link classification, when the number of labeled nodes per class is very few (1/3/5), the task is a graph few-shot learning task. Generally, we use $N$-way $K$-shot to describe the problem setting of tasks, which means that the number of predicted classes is $N$ and the available labels of each class is $K$. The labeled data is denoted as the support set or demonstration set $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N\times K}$ where $x_i$ can be a node instance $v_i$ or link instance $(v_i^s, v_i^t)$.

**Graph in-context learning**. Given an attributed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$ and the support instances $\mathcal{S}$ in graph few-shot learning task, graph ICL aims to learn a model $f_{\text{ICL}}(\cdot)$ that can fast adopt to the novel patterns from support set $\mathcal{S}$ and the graph structure $\mathcal{G}$ without the need for fine-tuning. The graph ICL model can make predictions according to the structural context of the graph and the support set instances with $y_q = f_{\text{ICL}}(x_q|\mathcal{S}, \mathcal{G})$.

### 3.2 Graph task reformulation as structure-aware sequence modeling

To extract generalizable knowledge across cross-domain and cross-granularity tasks, we reformulate diverse graph few-shot learning tasks as unified sequence modeling problem, so that these tasks can later be represented in the unified task space. Specifically, we construct prompt graph to capture the structural context. Then the graph few-shot learning tasks, including node and link classification, are reformulated as structure-aware sequence prediction problems conditioned on the prompt graph.

**Prompt graph construction** For graph tasks, the structure information is critical for downstreaming tasks due to the homophily phenomenon (Zhu et al. 2020). To capture the structural dependencies of graph instances, we use the graph instances in the support set to construct a task-specific prompt graph, and then generate prompt nodes to describe the structural context of the graph instances. Specifically, for node classification task $S_{\text{nc}} = \{v_i, y_i\}_{i=1}^{N\times K}$, we construct prompt graph $\mathcal{G}_{\text{nc}} = \{\mathcal{V}_{\text{nc}}, \mathcal{E}_{\text{nc}}, \mathbf{X}_{\text{nc}}\}$ where $\mathcal{V}_{\text{nc}} = \mathcal{V}_S \cup \{v_q\} \cup \mathcal{N}(\mathcal{V}_S \cup \{v_q\})$, $\mathcal{E}_{\text{nc}} = (\mathcal{V}_{\text{nc}} \times \mathcal{V}_{\text{nc}}) \cap \mathcal{E}$, $\mathbf{X}_{\text{nc}} = \mathbf{X}[\mathcal{V}_{\text{nc}}]$, $\mathcal{V}_S$ indicates the nodes in support set, and $\mathcal{N}(v)$ is the neighbor set of node $v$.

Correspondingly, for link classification tasks, we construct $\mathcal{G}_{\text{lc}} = \{\mathcal{V}_{\text{lc}}, \mathcal{E}_{\text{lc}}, \mathbf{X}_{\text{lc}}\}$ where $\mathcal{V}_{\text{lc}} = \mathcal{V}_S^{s,t} \cup \{v_q^s\} \cup \{v_q^t\} \cup \mathcal{N}(\mathcal{V}_S^{s,t} \cup \{v_q^s\} \cup \{v_q^t\})$, $\mathcal{E}_{\text{lc}} = (\mathcal{V}_{\text{lc}} \times \mathcal{V}_{\text{lc}}) \cap \mathcal{E}$, $\mathbf{X}_{\text{lc}} = \mathbf{X}[\mathcal{V}_{\text{lc}}]$, and the $\mathcal{V}_S^{s,t}$ is the union set of source nodes and target nodes in support set.

**Structure-aware sequence prediction** Subsequently, we reformulate the graph few-shot learning tasks of different granularities into a unified sequence prediction problem with the context of graph structure. Specifically, we leverage the structural context of prompt graph $\mathcal{G}_{\text{nc}}/\mathcal{G}_{\text{lc}}$ to perform structure-aware sequence prediction.

For an $N$-way $K$-shot few-shot node classification task with a support set $\mathcal{S}_{\text{nc}} = \{(v_i, y_i)\}_{i=1}^{N\times K}$, we convert the support set into a sequence of (instance, label) pairs $[(v_i, y_i)]_{i=1}^{N\times K}$ and extract the corresponding prompt graph. Then the label of query node $v_q$ is predicted as follows:

$$y_q = f(v_q|[(v_i, y_i)]_{i=1}^{N\times K}, \mathcal{G}_{\text{nc}}), \tag{1}$$

where $f(\cdot)$ is the meta-learner which can both encode the structural context and the support set context.

For an $N$-way $K$-shot few-shot link classification task with support set $\mathcal{S}_{\text{lc}} = \{((v_i^s, v_i^t), y_i)\}_{i=1}^{N\times K}$, we convert the support set with the same way as node classification task, and extract the link task-specific prompt graph $\mathcal{G}_{\text{lc}}$ to predict the label $y_q$ of the link $(v_q^s, v_q^t)$ as follows:

$$y_q = f((v_q^s, v_q^t)|[((v_i^s, v_i^t), y_i)]_{i=1}^{N\times K}, \mathcal{G}_{\text{lc}}). \tag{2}$$

So far we have introduced the basic sequence-based framework to achieve graph in-context learning. Now we need to design a graph in-context learner $f(\cdot)$ which can capture both the in-context information and the structural context from the cross-domain and cross-granularity graph.

### 3.3 Structure-aware Transformer as graph in-context learner

Research shows that the ICL ability of LLMs such as GPT benefits from the basic architecture of Transformer. Therefore, we design SAT to adapt the structure-aware sequence prediction tasks and enable graph ICL.

**Node feature alignment with PLM** The traditional GNN models calculate the statistical characteristics such as TF-IDF or BoW as node features, which will result in significant discrepancies in the feature space of cross-domain graphs. Therefore, we utilize a pretrained language model (PLM) to encode the text information of each node as the initial features. Given a graph where each node $v_i$ is associated with a text tokens $T_i = [w_i^0, w_i^1, w_i^2, ..., w_i^n, w_i^{n+1}]$, we firstly encode the tokens with PLM, and then calculate the mean pooling of word representations as the node features as follows:

$$\mathbf{x}_i = \frac{1}{|T_i|} \sum_{j \in T_i} \text{PLM}(w_i^j), \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $d$ is the dimension of the text embedding, the first and last tokens of the sentence are [CLS] and [SEP].

**ELMES: Symmetrical class representation** In the scenario of graph ICL, the same class may be assigned different label IDs across different graph few-shot learning tasks. For example, the label mapping may be (Neural Network→0, Reinforcement Learning→1, Theory→2) in one task, and could be (Reinforcement Learning→0, Theory→1, Neural Network→2) in another task. However, with the simple and direct one-hot encoding, the Transformer-based model will generate different outputs with different label mappings, which will introduce harmful biases unrelated to the generalizable knowledge. Therefore, we introduce a symmetrical class representation method (Fifty et al. 2024) to ensure that the model is independent of the label mapping strategy.

Specifically, we follow (Fifty et al. 2024) to introduce Equal Length and Maximally Equiangular Set (ELMES) vectors as the class representations, which are theoretically proven to ensure the symmetry of label mapping under the Transformer-based meta-learner.

**SAT: Structure-aware Transformer** The vanilla Transformer is a sequence model and is unable to capture structural patterns in graph data. To address this, we design SAT to extract structural context from the task-specific prompt graph, thereby making the Transformer suitable for graph ICL. Specifically, our proposed SAT improves the Transformer to extract structural information by designing three modules: differential structure retriever, residual graph convolution, and task unification layer.

*Differential structural retriever*. We pre-train SAT on large-scale graphs to learn sufficient generalizable knowledge. The large amount of neighbors of nodes in large graphs leads to the over-smoothing problem, which means that the aggregated node features will converge to a constant vector. Therefore, we design a learnable structure retriever to adaptively select the informative neighbors that benefit the downstream tasks. Specifically, we introduce the reparameterization trick and adopt Gumbel-TopK to perform differential structural retrieval as follows:

$$\begin{aligned} p_j &= \mathbf{x}_i \mathbf{W}_r \mathbf{x}_j^T \\ \hat{\mathcal{N}}(v_i, K) &= \text{argsort}(\log(p_j) + g_j)[:K], \end{aligned} \tag{4}$$

where $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ is the learnable parameters of the bilinear attention to calculate the importance between the features of center node $\mathbf{x}_i$ and neighbor $\mathbf{x}_j$, $g_i$ is the Gumbel noise sampled from $\text{Gumbel}(0, 1)$. Finally, the structural retriever adaptively sampled the Top-K relative neighbors from the distribution of learnable importance metric $p_j$.

*Residual graph convolution*. In order to adaptively leverage structural context in the task-specific prompt graph, we perform residual graph convolution to simultaneously consider node's semantics and the local structure. Specifically, we calculate the aggregated node representations as follows:

$$\begin{aligned} \hat{\mathbf{x}}_i &= \text{GNN}(\mathbf{x}_i, \mathbf{x}_j | v_j \in \hat{\mathcal{N}}(v_i, K)) \\ \mathbf{h}_i &= \mathbf{x}_i + \hat{\mathbf{x}}_i, \end{aligned} \tag{5}$$

where GNN can be GCN (Kipf and Welling 2016), which performs mean aggregation to the retrieved neighbors, or GAT (Veličković et al. 2018), taking the importance of each node into consideration. $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ respectively represent the text semantics and local structural context of $v_i$. The residual connection module is adopted to capture the structural context while retaining the initial semantic information.

*Task unification layer*. Further, to make our model capable of cross-granularity graph tasks, we design a task unification layer to project the target graph instance into a uniform space. Specifically, for few-shot node classification and link classification tasks, we use a feature projection layer to transform the nodes and node pairs as follows.

$$\begin{aligned} \mathbf{h}_i^{\text{n}} &= \mathbf{W}_{\text{n}} \cdot \mathbf{h}_i \\ \mathbf{h}_i^{\text{l}} &= \mathbf{W}_{\text{l}} \cdot [\mathbf{h}_i^s, \mathbf{h}_i^t], \end{aligned} \tag{6}$$

where $\mathbf{W}_{\text{n}} \in \mathbb{R}^{d \times d}$, $\mathbf{W}_{\text{l}} \in \mathbb{R}^{d \times 2 \cdot d}$. The task unification layer enables the SAT to represent tasks across granularities.

### 3.4 Large-scale pre-training with meta-optimization

To learn generalizable knowledge for adaptation to cross-domain graphs, we pre-train the SAT by meta-optimization on the large-scale meta-training tasks. In the following part, we will introduce how to perform the graph in-context inference with SAT and design a meta-optimization objective on cross-domain and cross-granularity meta-training tasks.

**Graph in-context inference** Given a graph few-shot learning tasks $(\mathcal{S}, \mathcal{Q})$, we firstly encode the graph instances and class representation $\mathcal{S} = \{(\mathbf{h}_i, \mathbf{c}_i)\}_{i=1}^{K \times N}$, $\mathcal{Q} = \{(\mathbf{h}_i, \mathbf{c}_i)\}_{i=K \times N+1}^{K \times (N+M)}$, where $\mathbf{h}_i$ is the unified representation of nodes or edges, $\mathbf{c}_i$ is the ELMES encoding of the label of

$v_i$ or $(v_i^s, v_i^t)$. Then, we concatenate the graph instance representation and the category representation in the support set to construct the in-context matrix as follows:

$$\mathbf{S} = \begin{bmatrix} \mathbf{h}_1 & , \mathbf{c}_1 \\ \mathbf{h}_2 & , \mathbf{c}_1 \\ \dots & , \dots \\ \mathbf{h}_{N\times K} & , \mathbf{c}_N \end{bmatrix}, \tag{7}$$

where $\mathbf{S} \in \mathbb{R}^{N\cdot K\times(d+k)}$ is the in-context matrix of the support set. We can predict the category of the query graph instance $\mathbf{h}_q$ with the SAT as follows:

$$\mathbf{o}_q = f(x_q|\mathcal{S},\mathcal{G}) = \mathbf{W}_q \cdot \text{SAT}(\begin{bmatrix} \mathbf{S} \\ \mathbf{h}_q, \mathbf{c}_u \end{bmatrix}), \tag{8}$$

where $\mathbf{W}_q \in \mathbb{R}^{d_o\times N}$ is the classifier head, $d_o$ is the output dimension of SAT, $\mathcal{G}$ is the task-specific prompt graph, $\mathbf{c}_u$ is an unknown class representation with random initialization, and $\mathbf{o}_q \in \mathbf{R}^N$ is the output logits.

**Backward with meta-optimization** Our objective is to learn a graph in-context learner which can predict well on the query instances given the support set. Therefore, we optimize the SAT with the cross-entropy between the graph in-context inference result of $\mathbf{o}_q$ and the corresponding ground-truth label $y_q$, which acts as meta-optimization. For better generalization to cross-domain and cross-granularity tasks, the pre-training phase is conducted on meta-training tasks constructed from diverse graphs, and we calculate the meta-loss as follow:

$$\begin{aligned}\mathbb{E}[\mathcal{L}_\Theta] = &\sum_{\mathcal{G}_i\in\mathbb{G}_{\text{nc}}}\sum_{(\mathcal{S}_{\text{nc}},\mathcal{Q}_{\text{nc}})\in\mathcal{T}_{\mathcal{G}_i}}\sum_{v_q\in\mathcal{Q}_{\text{nc}}}\mathcal{L}(f(\mathcal{S}_{\text{nc}},v_q|\mathcal{G}_{\text{nc}}),y_q)+\\ &\sum_{\mathcal{G}_i\in\mathbb{G}_{\text{lc}}}\sum_{(\mathcal{S}_{\text{lc}},\mathcal{Q}_{\text{lc}})\in\mathcal{T}_{\mathcal{G}_i}}\sum_{(v_q^s,v_q^t)\in\mathcal{Q}_{\text{lc}}}\mathcal{L}(f(\mathcal{S}_{\text{lc}}\},(v_q^s,v_q^t)|\mathcal{G}_{\text{lc}}),y_q),\end{aligned} \tag{9}$$

where $\mathcal{L}$ is the cross-entropy loss for classification tasks, $\mathbb{G}_{\text{nc}}$, $\mathbb{G}_{\text{lc}}$ are respectively the pre-training graphs from which we learn meta-knowledge for few-shot node classification and link classification, $\mathcal{T}_{\mathcal{G}_i}$ is the meta-training task set generated from the graph $\mathcal{G}_i$, $\mathcal{G}_{\text{nc}}$ and $\mathcal{G}_{\text{lc}}$ are the prompt graphs.

# 4 Experiments

## 4.1 Datasets

To perform universal graph ICL on cross-domain and cross-granularity tasks, we conduct experiments on academic networks and knowledge graphs for pre-training and validation.

**Citation network**. MAG240M (Hu et al. 2020) is a large-scale graph collected from Microsoft academic network. Cora and Citeseer (Yang, Cohen, and Salakhutdinov 2016) are citation networks from the domain of artificial intelligence. ogbn-arxiv (Hu et al. 2020) is constructed from the citation links between papers in arXiv. The texts of nodes are extracted from the title and abstract of papers, and the labels are the categories of papers.

**Knowledge graph**. WikiCS dataset (Mernyei and Cangea 2020) is a web graph extracted from knowledge graph Wikipedia. Each node is a page, and the edges are the hyperlinks on the page. The node features are the contents on the web page, and the labels of nodes are the categories divided by Wikipedia. WikiKG90M (Hu et al. 2020) is a large-scale Wikipedia knowledge graph constructed by OGB. FB15K237 is a knowledge graph extracted from Freebase, which contains 237 categories of relations. YAGO10 is the subset of the famous knowledge graph YAGO3 (Mahdisoltani, Biega, and Suchanek 2015).

The detailed statistics of datasets are shown in Table 1.

| Dataset | #Nodes | #Edges / #Triplets | #Classes/ #Relations |
|---|---|---|---|
| MAG240M | 120M | 1.3B | 153 |
| WikiKG90M | 91M | 600M | 1,387 |
| Cora | 2,708 | 10,556 | 7 |
| Citeseer | 3,186 | 3,100 | 6 |
| ogbn-arxiv | 169,343 | 2,315,598 | 40 |
| Wiki-CS | 11,701 | 431,726 | 10 |
| FB15K237 | 14,541 | 310,116 | 237 |
| YAGO10 | 123,182 | 1,089,040 | 37 |

Table 1: The statistics of graph datasets.

## 4.2 Baselines

We use four categories of baseline models including GNN and (graph) ICL models, which are introduced in detail as follows:

**GNN models** include GCN (Kipf and Welling 2016) and GAT (Veličković et al. 2018) are adopted as baselines to evaluate the performance of basic supervised learning.

**ICL models**: PN (Snell, Swersky, and Zemel 2017) uses the PLM features to calculate prototype vectors and perform classification, which reflects the basic distinguishable ability of PLM. SNAIL (Mishra et al. 2018) uses a temporal convolution layer to encode the in-context information and memorize it with a soft attentive layer. GPICL (Kirsch et al. 2024) proposes a pure attention network with Transformer as a meta-learner. CAML (Fifty et al. 2024) further designs ELMES label encoding to guarantee the symmetry under different label mapping strategies.

**Graph ICL models** GPN (Ding et al. 2020) is based on PN and further considers node importance to calculate more accurate prototype representations. Prodigy (Huang et al. 2023) is a graph-based ICL model, which formulates graph few-shot learning as link prediction between data nodes and label nodes. OFA (Liu et al. 2024a) uses LLM to enhance the representation of node's textual information.

## 4.3 Metrics and Settings

We use the metrics of accuracy (ACC) on 5-way 1-shot and 5-way 3-shot node classification (NC) and link classification (LC) tasks to evaluate the performance of **CAGML**. For supervised GNN, we train models with the labels in the support set and use query set data for evaluation. For ICL models, we adopt two large-scale graph datasets: MAG240M and WikiKG90M for pre-training. And the models trained with MAG240M are used to evaluate the NC performance

| Models | Cora | | Citeseer | | ogbn-arxiv | | WikiCS | | FB15K237 | | YAGO10 | | Ave. ACC. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5-way NC | | 5-way NC | | 5-way NC | | 5-way NC | | 5-way LC | | 5-way LC | | |
| | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | |
| **Supervised GNN** | | | | | | | | | | | | | |
| GCN | 65.1 | 77.3 | 49.1 | 60.8 | 61.5 | 74.6 | 53.2 | 69.3 | 68.9 | 78.5 | 43.9 | 52.7 | 62.9 |
| GAT | 65.0 | 76.2 | 48.5 | 60.5 | 60.6 | 73.4 | 56.2 | 69.8 | 67.0 | 76.9 | 41.2 | 50.0 | 62.1 |
| **In-context learning** | | | | | | | | | | | | | |
| PN | 53.2 | 67.4 | 50.1 | **63.8** | 65.2 | 80.6 | 45.8 | 59.5 | 78.3 | 88.4 | 56.1 | 72.3 | 65.1 |
| SNAIL† | 35.3 | 50.6 | 49.4 | 60.9 | **70.3** | 77.3 | 39.3 | 44.6 | 84.6 | 89.9 | 65.9 | 68.7 | 61.4 |
| GPICL† | 47.9 | 60.2 | 50.5 | 61.9 | 67.2 | 80.9 | 43.6 | 52.9 | 86.2 | 91.2 | 64.9 | 74.0 | 65.1 |
| CAML† | 43.6 | 58.7 | **50.8** | 60.9 | 70.2 | **83.2** | 45.8 | 54.0 | 84.9 | **92.0** | 67.0 | 75.1 | 65.5 |
| **Graph in-context learning** | | | | | | | | | | | | | |
| GPN | **58.0** | **75.5** | 44.0 | 58.0 | 55.6 | 70.6 | **50.9** | **67.6** | 54.6 | 67.9 | 41.9 | 51.7 | 58.0 |
| Prodigy | 52.6 | 65.3 | 41.2 | 53.6 | 57.5 | 61.5 | 33.7 | 51.2 | 78.2 | 88.0 | 54.2 | 70.1 | 58.3 |
| OFA | 53.8 | 66.9 | 43.7 | 52.5 | 59.3 | 67.8 | 42.1 | 57.1 | 76.3 | 85.7 | 57.2 | 73.1 | 60.8 |
| **CAGML** | 53.7 | 68.6 | 47.6 | 59.8 | 69.1 | 83.1 | 49.6 | 61.1 | **87.9** | 91.5 | **71.5** | **81.5** | **68.7** |

† indicates the model is trained and evaluated separately on the node classification and link classification tasks.

Table 2: The performance comparison on few-shot generalization.

on citation networks and WikiCS, while the models trained with WikiKG90M are used to evaluate the LC performance on knowledge graphs. For graph ICL and our **CAGML**, we train the models with both datasets and generalize to both cross-domain and cross-granularity tasks.

For the specific experiment settings of **CAGML**, the batch size is set to 16 and we use the AdamW (Loshchilov and Hutter 2019) to optimize the parameters. The learning rate of the optimizer is initialized to 0.00001 and the weight decay is set to 0.0005. For each dataset, we evaluate the model performance on 100 randomly generated meta tasks. We implement the model with Pytorch (Paszke et al. 2019) and run the baselines and our model over NVIDIA Tesla V100 with 32GB memory. We use all-MiniLM-L6-v2 as the PLM to encode node features.

## 4.4 Overall performance analysis

The overall performance comparison of both cross-domain and cross-granularity generalization is shown in Table 2, where the best results are highlighted in **bold**, and the second-best results are underlined. From the statistics in Table 2, we can draw the conclusions below:

- In general, our proposed **CAGML** ranks first regarding the average classification performance in cross-domain and cross-granularity graph ICL scenarios. Even on some datasets, such as FB15K237 and YAGO10, it outperforms supervised learning models and achieves the best performance among various ICL models. In terms of average classification performance across various tasks, we achieve 9.2%, 4.7%, and 13.0% performance improvements compared to the supervised GNN models, SOTA ICL, and SOTA graph ICL models.
- Compared to in-context learning models, **CAGML** not only allows cross-domain and cross-granularity training on two large datasets, but also has the ability to extract structural knowledge. Therefore, it can achieve superior generalization across different datasets. When compared with the CAML model, which is pre-trained on a single task, our model still achieves an average improvement of 5.4% and 4.2% on NC and LC tasks, respectively. Especially for the Cora and WikiCS datasets, where the local structural context is important for node classification, the improvements are 19.6% and 10.9%.
- For graph ICL models, GPN is a metric-based meta-learning model that mainly relies on the distinguishability of PLM embedding. Though Prodigy and OFA can further learn meta-knowledge from pre-trained datasets, the performance of **CAGML** is significantly better than that of Prodigy and OFA, with improvement rates of 17.8% and 13.0%, respectively. Because these graph ICL models rely on GNN with limited parameters to memorize cross-domain knowledge. Benefiting from the powerful representation ability of SAT, our model not only possesses richer cross-domain knowledge but also demonstrates a stronger ability to generalize to new tasks.

## 4.5 Ablation study

We design variant models of **CAGML** to analyze the effectiveness of the proposed modules. **CAGML w/o ST** does not use any structural information but only the semantics of node features. **CAGML w/o DSR** is the variant that removes the differential structural retriever (DSR) and uses all local structural context. **CAGML w/o RGC** removes the residual module in the residual graph convolution (RSC) and directly uses the GNN output. **CAGML-CN** and **CAGML-KG** remove the task unification (TU) layer and pre-train models only on MAG240M and WikiKG90M respectively. The experimental results of the 5 variant models on 5-way 3-shot tasks are shown in Table 3.

From the results, we can draw the conclusion that all the designed modules are beneficial: the structural context is critically important in graph ICL, and our proposed residual graph convolution layer and differential structure retriever can adaptively leverage the structural context and node semantics. Without DSR, **CAGML** cannot learn to effectively extract structural information because the pre-training

| Model | Cora | WikiCS | FB15K237 | YAGO10 |
|---|---|---|---|---|
| **CAGML w/o ST** | 58.7 | 51.5 | 90.2 | 76.9 |
| **CAGML w/o DSR** | 59.3 | 51.6 | 91.3 | 75.1 |
| **CAGML w/o RGC** | 18.6 | 17.5 | 29.3 | 34.1 |
| **CAGML-CN** | 65.5 | 59.7 | 23.2 | 23.9 |
| **CAGML-KG** | 27.8 | 24.2 | 90.3 | 78.2 |
| **CAGML** | **68.6** | **61.1** | **91.5** | **81.5** |

Table 3: The ablation study results on **CAGML**.

graphs are too large, which may lead to over-smoothing. **CAGML w/o RGC** would notably impact performance. Because removing RGC will lead to the model's inability to capture the semantic information of the nodes, which is crucial in our model with PLM-initialized features. In addition, models trained on only one task perform poorly on another task, and training on both datasets can provide complementary knowledge to improve model performance, demonstrating the importance of the TU layer.

### 4.6 Generalization to unseen task settings

To demonstrate how the model generalizes to different unseen task settings in pre-training phases, we gradually increase the number of shots from 1 to 15. The results are shown in Figure 2. The results show that **CAGML** can also

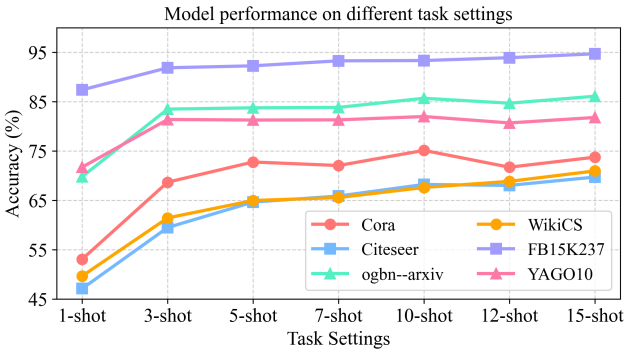


Figure 2: The performance on different task settings with varying numbers of shot.

leverage the more abundant label information in the support set. This indicates that our model has indeed learned how to extract novel patterns from demonstration instances, rather than simply fitting to existing task patterns.

### 4.7 Impact of SAT scaling

We also study how the scale of SAT influences the performance of our model. We investigate the performance of SAT at three different scales, including small (3M), medium (15M), and large (100M) where the results are shown in Figure 3. From the results, we can find that increasing the scale of SAT can increase the generalization performance on all datasets. Because larger model have more powerful representation ability to memorize the cross-domain knowledge. However, the degree of performance improvement gradually reduces as the scale increases.

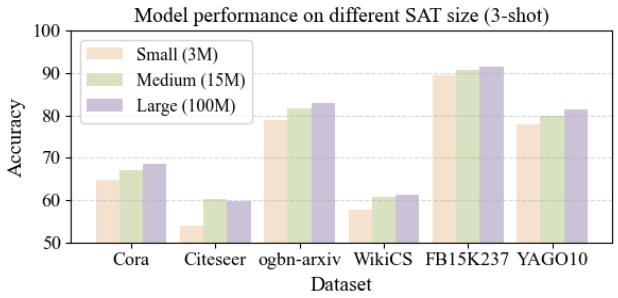


Figure 3: The performance on different scale of SAT.

### 4.8 Visualization of the embedding distribution

To intuitively analyze the embedding distribution of the learned **CAGML**, we conduct a visual analysis shown in Figure 4. We can find that the pre-trained **CAGML** can en-

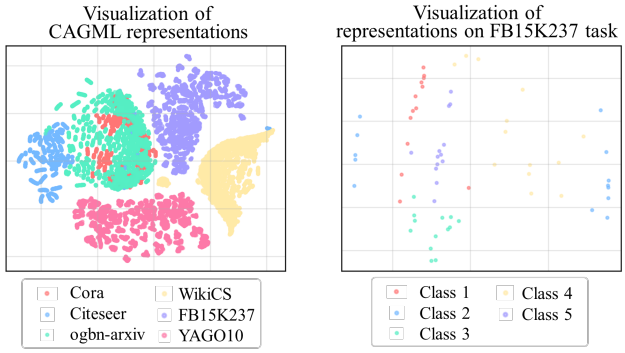


Figure 4: The visualization of the embedding distribution across datasets and within task.

code cross-domain graph data into a unified representation space and effectively distinguish graph data from different domains. Besides, within a specific task, the **CAGML** embeddings can also discriminate the query nodes of different classes without fine-tuning.

## 5 Conclusions

In this paper, we propose **CAGML**, a context-aware graph meta-learning model to achieve effective graph ICL for cross-domain and cross-granularity graph tasks. Specifically, we first formulate graph few-shot learning tasks into a structure-aware sequence modeling problem. Then, we design SAT as a graph in-context learner to extract structural context. Finally, we perform large-scale pre-training on SAT with meta-optimization. The results on cross-domain graph datasets demonstrate the superiority of **CAGML**. For future work, we will consider more complex graph data such as text-free, heterogeneous, and heterophilic graphs.

## References

Alex, G.; Greg, W.; and Ivo, D. 2014. Neural turing machines. In *arXiv preprint arXiv:1410.5401*.

Alex, N.; Joshua, A.; and John, S. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; and Amodei, D. 2020. Language models are few-shot learners. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS)*.

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; and Wei, F. 2023. Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics (ACL)*, 4005–4019. ACL.

Ding, K.; Wang, J.; Li, J.; Shu, K.; Liu, C.; and Liu, H. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, 295–304. ACM.

Fifty, C.; Duan, D.; Junkins, R. G.; Amid, E.; Leskovec, J.; Ré, C.; and Thrun, S. 2024. Context-aware meta-learning. In *International Conference on Learning Representations (ICLR)*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 1126–1135. PMLR.

Gharoun, H.; Momenifar, F.; Chen, F.; and Gandomi, A. H. 2024. Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Computing Survey*, 56(12).

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS)*, 22118–22133.

Hu, Z.; Li, Y.; Chen, Z.; Wang, J.; Liu, H.; Lee, K.; and Ding, K. 2024. Let's Ask GNN: Empowering Large Language Model for Graph In-Context Learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (Finding of EMNLP)*, 1396–1409. ACL.

Huang, K.; and Zitnik, M. 2021. Graph meta learning via local subgraphs. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS)*, 5862–5874. PMLR.

Huang, N.; Zhou, G.; Zhang, M.; and Wang, S. 2025. Structural denoised contrastive self-supervied graph meta-learning. *Proceedings of the 30th International Conference on Database Systems for Advanced Applications (DASFAA)*, 36: 16129–16152.

Huang, Q.; Ren, H.; Chen, P.; Kržmanc, G.; Zeng, D.; Liang, P.; and Leskovec, J. 2023. PRODIGY: Enabling in-context learning over graphs. *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kirsch, L.; Harrison, J.; Sohl-Dickstein, J.; and Metz, L. 2024. General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*.

Liu, H.; Feng, J.; Kong, L.; Liang, N.; Tao, D.; Chen, Y.; and Zhang, M. 2024a. One For All: Towards training one graph model for all classification tasks. In *International Conference on Learning Representations (ICLR)*.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Philip, S. Y. 2022a. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900.

Liu, Y.; Li, M.; Li, X.; Giunchiglia, F.; Feng, X.; and Guan, R. 2022b. Few-shot node classification on attributed networks with graph meta-learning. In *Proceedings of the 45th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*, 471–481. ACM.

Liu, Y.; Li, S.; Zheng, Y.; Chen, Q.; Zhang, C.; and Pan, S. 2024b. ARC: A Generalist Graph Anomaly Detector with In-Context Learning. *Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS)*, 37: 50772–50804.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *The Seventh Biennial Conference on Innovative Data Systems Research (CIDR)*.

Mandal, D.; Medya, S.; Uzzi, B.; and Aggarwal, C. 2022. Meta learning with graph neural networks: Methods and applications. *ACM SIGKDD Explorations Newsletter*, 23(2): 13–22.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 785–794. ACM.

Mernyei, P.; and Cangea, C. 2020. Wiki-CS: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A simple neural attentive meta-learner. In *International Conference on Learning Representations (ICLR)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS)*, 8026–8037. PMLR.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 1842–1850.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 4077–4087. PMLR.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Wang, N.; Luo, M.; Ding, K.; Zhang, L.; Li, J.; and Zheng, Q. 2020. Graph few-shot learning with attribute matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, 1545–1554. ACM.

Wang, S.; Tan, Z.; Liu, H.; and Li, J. 2023. Contrastive meta-learning for few-shot node classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2386–2397. ACM.

Wu, S.; Wang, Y.; and Yao, Q. 2025. Why In-Context Learning Models are Good Few-Shot Learners? In *International Conference on Learning Representations (ICLR)*.

Yan, H.; Li, C.; Long, R.; Yan, C.; Zhao, J.; Zhuang, W.; Yin, J.; Zhang, P.; Han, W.; Sun, H.; et al. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS)*, 36: 17238–17264.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*.

Zhou, F.; Cao, C.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Geng, J. 2019. Meta-GNN: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 2357–2360. ACM.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS)*, volume 33, 7793–7804. PMLR.