

# Dynamic Text Bundling Supervision for Zero-Shot Inference on Text-Attributed Graphs

Yusheng Zhao<sup>1</sup>, Qixin Zhang<sup>2</sup>, Xiao Luo<sup>4</sup>, Weizhi Zhang<sup>5</sup>,  
Zhiping Xiao<sup>3</sup>, Wei Ju<sup>1</sup>, Philip S. Yu<sup>5</sup>, Ming Zhang<sup>1</sup>

<sup>1</sup> Peking University, <sup>2</sup> Nanyang Technological University, <sup>3</sup> University of Washington

<sup>4</sup> University of California, Los Angeles, <sup>5</sup> University of Illinois Chicago,

yusheng.zhao@stu.pku.edu.cn, qixinzhang1106@gmail.com

xiaoluo@cs.ucla.edu, {wzhan42, psyu}@uic.edu,

patxiao@uw.edu, {juwei, mzhang\_cs}@pku.edu.cn

## Abstract

Large language models (LLMs) have been used in many zero-shot learning problems, with their strong generalization ability. Recently, adopting LLMs in text-attributed graphs (TAGs) has drawn increasing attention. However, the adoption of LLMs faces two major challenges: *limited information on graph structure* and *unreliable responses*. LLMs struggle with text attributes isolated from the graph topology. Worse still, they yield unreliable predictions due to both information insufficiency and the inherent weakness of LLMs (*e.g.*, hallucination). Towards this end, this paper proposes a novel method named Dynamic Text Bundling Supervision (DENSE) that queries LLMs with bundles of texts to obtain bundle-level labels and uses these labels to supervise graph neural networks. Specifically, we sample a set of bundles, each containing a set of nodes with corresponding texts of close proximity. We then query LLMs with the bundled texts to obtain the label of each bundle. Subsequently, the bundle labels are used to supervise the optimization of graph neural networks, and the bundles are further refined to exclude noisy items. To justify our design, we also provide theoretical analysis of the proposed method. Extensive experiments across ten datasets validate the effectiveness of the proposed method.

## 1 Introduction

Text-attributed graphs (TAGs) [72, 76] are an important form of graph data, containing textual descriptions associated with each node. By combining textual information with non-Euclidean graph topology, TAGs serve as natural structured data representation in many applications, including citation networks [56], social networks [50], e-commerce networks [41], and webpage networks [10]. As complete labeling of these large networks is often time-consuming and costly, efforts have been made to utilize semi-supervised learning [28, 63], transfer learning [11, 86], and few-shot/zero-shot learning [13, 80, 75] to understand text-attributed graphs with limited labels.

Large language models (LLMs) [73, 37] have been observed to exhibit strong zero-shot generalization capability, enhancing the performance on various types of data, including visual signals [81], texts [38], programming code [74], and graphs [66]. Recently, there have been efforts in integrating LLMs in text-attributed graphs [19, 23, 85]. One line of research integrates the graph topology into language models [8, 43, 85], converting non-Euclidean topology into a sequence of tokens. However, building such foundation models requires a large amount of data [67], and the conversion to Euclidean data inevitably incurs information loss [40]. Another line of research directly utilizes the zero-shot generalization ability of existing LLMs to understand node attributes [7, 32, 67], and

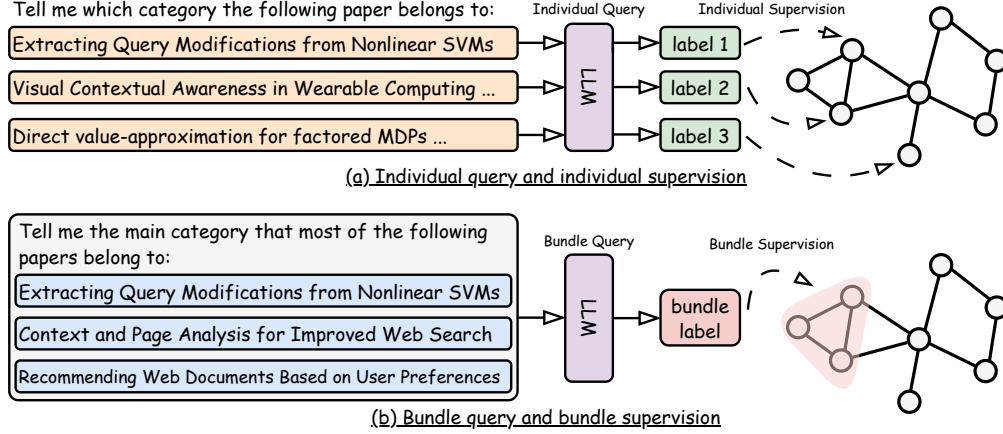


Figure 1: (a) Querying LLMs with individual texts and supervising graph learning with individual labels. (b) By creating text bundles, we perform bundle queries to obtain bundle labels for supervision.

utilizes the output of LLMs as supervision signals for training graph neural networks (GNNs) [28] or as clustering centers [67]. However, the text attributes are often isolated from the graph topology, and the unreliable responses from LLMs also pose challenges for subsequent operations.

Adopting LLMs in zero-shot inference on TAGs faces two major challenges: (1) *LLMs receive limited information on graph structure*. Graph topology is non-Euclidean, making it difficult to transform into token sequences with limited context windows. (2) *LLMs yield unreliable responses*. The inherent weakness of LLMs (e.g., hallucination), together with limited information, makes the responses from LLMs unreliable, damaging subsequent operations like clustering, classification, or supervision.

Towards this end, this paper proposes a novel method named dynamic text bundling supervision that queries LLMs and supervises graph neural networks using text bundles. As can be seen in Figure 1, conventional methods [9, 67] query LLMs with individual text items (e.g., in citation networks, this would be individual papers’ titles and abstracts). The LLMs then return the annotations of these texts, which are used as supervision signals. This paradigm faces the two major challenges mentioned above: the LLMs suffer from limited information, and the downstream supervision signals are unreliable. By comparison, this work proposes to query LLMs and supervise subsequent graph learning with text bundles. We first sample topologically or semantically similar text items to form a text bundle, and then query the LLMs about the *mode category* (i.e., the most frequent category of the text items) as the bundle label. Subsequently, we design bundle supervision that uses the bundle labels to train a graph neural network, and during this process, bundles are further refined to exclude noisy items. In this way, the LLMs receive richer information from multiple interrelated text items in a bundle (challenge 1), and the predicted bundle labels are more robust to the uncertainty or misinterpretation of single text items with bundle supervision and refinement (challenge 2).

The contribution of this paper can be summarized as follows. ❶ We introduce a new perspective that connects bundle structure and text-attributed graphs to provide robust supervision of graph neural networks. ❷ We propose a novel framework consisting of bundle sampling, bundle query, bundle supervision, and bundle refinement. We also provide rigorous theoretical analysis of our method, showing its tolerance to outlier nodes and the convergence properties of optimization. ❸ We perform extensive experiments on ten text-attributed graph datasets across various domains, and the results validate the effectiveness of the proposed method compared to competing baselines.

## 2 Related Works

**Text-Attributed Graphs.** Text-attributed graphs (TAGs) are a special type of graphs whose nodes are associated with textual attributes [72, 76]. They are common forms of data in many fields, such as citation networks [3], knowledge graphs [55], social networks [31], web page networks [18], etc. Research on TAGs generally focuses on combining textual attributes with graph structures, with the help of text embedding methods [64, 46] and network embedding methods [65, 68, 35]. As the annotation costs of TAGs are usually high, efforts have been made in semi-supervised learning [70, 79], transfer learning [78, 20, 86], and few-shot learning [22, 80, 75]. With the advancement

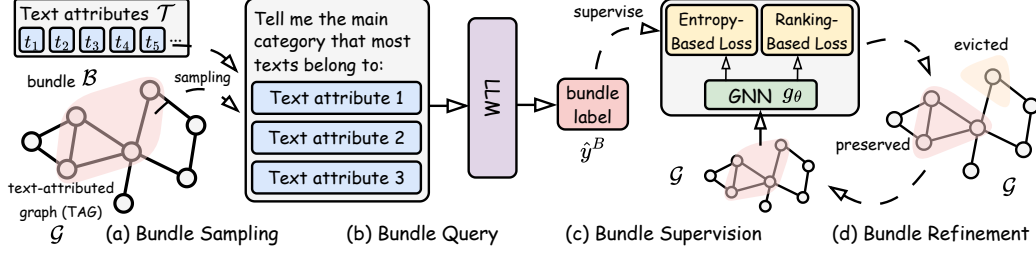


Figure 2: The overall framework of our method. We first sample nodes of proximity to form bundles (a), which are then used to query the LLM about their main categories (b). Subsequently, the bundle labels from the LLM’s response are used to supervise a graph neural network (c). During optimization, we further refine the bundle to exclude noisy nodes (d).

of large language models, this work makes a step further, focusing on the zero-shot inference of text-attributed graphs [9, 66, 67] with the help of LLMs.

**Large Language Model for Graphs.** Large language models (LLMs) [73, 37] have shown impressive performance in understanding data beyond natural languages, including programming languages [25], sequences of numbers [26], mathematics [58], and graphs [27, 54, 34]. LLMs exhibit strong generalization ability, enabling few-shot or zero-shot inference on graphs. One line of research aims to build a foundation model, incorporating graph structures into current language model architectures [71, 15, 36, 66]. These methods often require training to align the graph structure and natural language [77, 86], involving a large amount of labeled or paired data. Another line of research makes use of the inference capability of existing LLMs to generate labels or related information of graphs [60, 9, 7, 67]. However, they often use isolated nodes [67] or explicit descriptions that are hard for LLMs to understand [61]. Additionally, the noisy labels generated by LLMs can further harm subsequent inference operations (*e.g.*, supervising neural networks, performing clustering) on graphs. Compared to these methods, this paper proposes to use text bundles to query LLMs and supervise graph neural networks, leading to richer information and more robust optimization.

### 3 Methodology

**Problem Definition.** We denote a text-attributed graph as  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{Y})$ , where  $\mathcal{V}$  is the set of nodes,  $\mathcal{E}$  is the set of edges,  $\mathcal{T}$  is the set of textual attributes, and  $\mathcal{Y}$  is the set of node labels. Each node  $v_i \in \mathcal{V}$  is associated with textual descriptions  $t_i \in \mathcal{T}$  and the corresponding label  $y_i$ . For each node, we can obtain its vectorized embedding via a text encoder  $f_\theta$ , *i.e.*,  $\mathbf{x}_i = f_\theta(t_i) \in \mathbb{R}^d$ . We denote the total number of nodes as  $n = |\mathcal{V}|$ . A node bundle is defined as a set of nodes in the graph, and a text bundle corresponding to the node bundle is defined as a set of text attributes associated with the node bundle. For simplicity, we use the term bundle and notation  $\mathcal{B}$  to denote the indices of corresponding node bundles and text bundles. The goal of zero-shot inference on text-attributed graphs is to infer the node labels  $\mathcal{Y}$  according to the graph topology  $\mathcal{V}, \mathcal{E}$ , and the textual attributes  $\mathcal{T}$ .

#### 3.1 Framework Overview

The overall framework of the proposed method is illustrated in Figure 2. We first perform bundle sampling, constructing node bundles according to topological or semantic proximity (Section 3.2). With the obtained node bundles, we transform the corresponding text bundles into prompts and query the LLM about the most frequent category of the bundle (Section 3.3). With these bundle labels, we perform bundle supervision, training graph neural networks with entropy-based and ranking-based supervision. Additionally, theoretical analysis is provided regarding the properties of bundle supervision to justify our design (Section 3.4). During the optimization process, we further refine the bundles dynamically to exclude noisy components (Section 3.5).

#### 3.2 Bundle Sampling

We first introduce the method for sampling bundles. Intuitively, we aim for most nodes within a bundle to belong to the same category (*i.e.* a strong mode), so that LLMs more easily predict the mode category and the bundle label more accurately reflects the nodes it contains. To achieve this,

we sample nodes of close proximity. Specifically, we first randomly sample the core node  $v_c$  from the set of nodes  $\mathcal{V}$ , and then sample the rest of the nodes. We fix the size of a bundle as  $n_B$ , and design two criteria for sampling: topological proximity and semantic proximity.

**Topological Proximity.** For a given core node  $v_c$  in graph  $\mathcal{G}$ , a common assumption is that a node is similar to nodes topologically close to itself [28, 17]. Formally, given two nodes  $v_c$  and  $v$ , their topological proximity can be measured by the length of the shortest path from  $v_c$  and  $v$ , denoted as  $d^{\mathcal{G}}(v_c, v)$ , and we can define topologically similar nodes with respect to  $v_c$  as:

$$\mathcal{N}_{\mathcal{G}}^k(v_c) = \{i \mid 1 \leq d^{\mathcal{G}}(v_i, v_c) \leq k\}, \quad k = \inf \{x \mid |\mathcal{N}^x(v_c)| \geq n_B - 1\} \quad (1)$$

where  $k$  is an adaptive hop size. For core nodes with many ( $k$ -hop) neighbors, a smaller hop size is used, and vice versa. We then sample  $(n_B - 1)$  nodes from the neighborhood  $\mathcal{N}_{\mathcal{G}}^k(v_c)$  to form the bundle  $\mathcal{B}$  together with the original core node  $v_c$ .

**Semantic Proximity.** For graphs with heterophily, topological proximity hardly entails similarity [84, 83, 82]. Therefore, we turn to semantic proximity utilizing vectorized representations of nodes. Specifically, given embeddings of each node  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$  and a core node  $v_c$  with corresponding embedding  $\mathbf{x}_c$ , we construct the node bundle based on the closeness in the embedding space  $\mathbb{R}^d$ :

$$\mathcal{B} = \{i \mid \mathbf{x}_i \in \mathcal{N}_{\mathcal{X}}^{n_B}(\mathbf{x}_c)\}, \quad (2)$$

where  $\mathcal{N}_{\mathcal{X}}^{n_B}(\mathbf{x}_c)$  denotes the set of top  $n_B$  vectors in  $\mathcal{X}$  that are closest to  $\mathbf{x}_c$  in terms of Euclidean distance (i.e.,  $L_2$  distance) in the embedding space.

In practice, different criteria are adopted for different types of graphs. For graphs with high homophily (e.g., citation networks), topological proximity is used. For graphs with high heterophily (e.g., webpage networks), semantic proximity is adopted. We repeatedly sample a set of node bundles as  $\{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_{n_S}\}$ , where  $n_S$  is the number of bundles. For simplicity, we omit the subscript of bundles and use  $\mathcal{B}$  for an arbitrary bundle in the following discussions.

### 3.3 Bundle Query

We then query LLMs to obtain information about the bundles. While it might be straightforward to provide individual text attributes for node-level pseudo-labels, this approach carries the risk of limited information (as the LLMs only receive information from a single isolated node attribute) and unreliable responses (since the output pseudo-labels can be highly noisy). By using bundling, LLMs receive more information from proximate nodes, making the decision regarding the mode category easier than individual classification, which results in more reliable annotations.

With the node bundles selected, we obtain their corresponding text bundles and construct a single prompt  $\mathcal{P}(\mathcal{B})$  for each text bundle with dataset description and task description:

$$\mathcal{P}(\mathcal{B}) = \langle \text{dataset\_description} \rangle \text{Concat}(\{t_i \mid i \in \mathcal{B}\}) \langle \text{task\_description} \rangle, \quad (3)$$

where the  $\text{Concat}(\cdot)$  operator concatenates all the text attributes in the bundle. We then query the LLM with the prompts to obtain the mode category of the bundle, denoted as  $\hat{y}^{\mathcal{B}}$ .

### 3.4 Bundle Supervision

The bundle labels are then used to supervise a graph neural network. Since a bundle label represents the mode category that most nodes in the bundle belong to, nodes from other categories may also be included. Therefore, effective bundle supervision requires tolerance for these "outliers". To address this, we design two supervisions: entropy-based supervision and ranking-based supervision. We denote the graph neural network as  $g_{\theta}$ , and it generates probability distributions for each node as:

$$\{\mathbf{z}_i\}_{i=1}^n = g_{\theta}(\{\mathbf{x}_i\}_{i=1}^n, \mathcal{E}), \quad \mathbf{p}_i = \text{softmax}(\mathbf{z}_i), \quad (4)$$

where  $\mathbf{z}_i \in \mathbb{R}^C$  is the logits,  $\mathbf{p}_i \in \mathbb{R}^C$  is the probability, and  $C$  is the number of classes.

**Entropy-based Supervision.** When a bundle  $\mathcal{B}$  has label  $\hat{y}^{\mathcal{B}}$ , the nodes in it are likely to fall into class  $\hat{y}^{\mathcal{B}}$  on average. Therefore, we compute the bundle class distribution  $\mathbf{p}(\mathcal{B})$  and the corresponding bundle-level entropy-based objective function  $\mathcal{L}_{BE}$  as follows:

$$\mathbf{p}(\mathcal{B}) = \text{softmax}\left(\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{z}_i\right), \quad \mathcal{L}_{BE} = \text{CE}(\mathbf{p}(\mathcal{B}), \hat{y}^{\mathcal{B}}), \quad (5)$$

where  $\text{CE}(\cdot, \cdot)$  is the cross-entropy loss. We then theoretically demonstrate that this bundle supervision (i.e.,  $\mathcal{L}_{BE}$ ) is more tolerant to outliers compared to individual supervision using cross-entropy. Formally, we have the following theorem:

**Theorem 3.1.** *Given a bundle  $\mathcal{B}$ , its corresponding bundle class distribution  $\mathbf{p}(\mathcal{B}) = (p_1, p_2, \dots, p_C)$ , an outlier node  $v_o, o \in \mathcal{B}$  with probability distribution  $\mathbf{p}_o = (p'_1, p'_2, \dots, p'_C)$ , denote  $m' = \text{argmax}_i \{p'_i\}_{i=1}^C$ . If the bundle label  $\hat{y} \neq m'$ , and  $p'_{m'} \geq p_{m'}$ , we have:*

$$0 \leq \frac{\partial \mathcal{L}_{BE}}{\partial \log p'_{m'}} \leq \frac{\partial \mathcal{L}_{IE}}{\partial \log p'_{m'}}, \text{ where } \mathcal{L}_{BE} = \text{CE}(\mathbf{p}(\mathcal{B}), \hat{y}) \text{ and } \mathcal{L}_{IE} = \frac{1}{|\mathcal{B}|} \cdot \text{CE}(\mathbf{p}_o, \hat{y}), \quad (6)$$

where  $\hat{y}$  is the bundle label,  $\mathcal{L}_{BE}$  is bundle supervision and  $\mathcal{L}_{IE}$  is individual supervision.

**Remark 1.** *Theorem 3.1 suggests that when encountering "outlier" nodes that conflict with the predicted mode category and the bundle distribution (i.e., the condition  $\hat{y} \neq m'$  and  $p'_{m'} \geq p_{m'}$  in the theorem), the bundle cross-entropy objective function (i.e.,  $\mathcal{L}_{BE}$  defined in Eq. 5) is more tolerant compared to supervising the nodes in the bundle individually (i.e.,  $\mathcal{L}_{IE}$  defined in the theorem), as evidenced by a smaller penalty imposed by the gradient.*

**Ranking-based Supervision.** To ensure that the supervision focuses more on bundles where the predicted bundle labels do not dominate the bundle's bundle probability distribution, we adopt the concept of ranking loss [6, 47, 69], and design a ranking-based loss as follows:

$$\mathcal{L}_R = -\min \left( \log \mathbf{p}(\mathcal{B})_{\hat{y}^B} - \log \max_{i=1}^C \{\mathbf{p}(\mathcal{B})_i\}, 0 \right), \quad (7)$$

where  $\mathbf{p}(\mathcal{B})_i \in \mathbb{R}$  denotes the  $i$ -th component of vector  $\mathbf{p}(\mathcal{B})$  (i.e., the predicted probability of class  $i$ ). When the category of the bundle label  $\hat{y}^B$  is not the highest in the predicted bundle probability distribution by the GNN  $g_\theta$ , the bundle  $\mathcal{B}$  is penalized by this loss function. On the other hand, when the category of  $\hat{y}^B$  has a high bundle probability  $\mathbf{p}(\mathcal{B})_{\hat{y}^B}$  (which does not necessitate all  $z_i, i \in \mathcal{B}$  to be high), the loss will be zero. In our implementation, a combination of the two supervision objectives is used, leading to the final objective function as follows:

$$\mathcal{L} = \mathcal{L}_{BE} + \mathcal{L}_R. \quad (8)$$

### Theoretical Analysis.

We then aim to present a rigorous theoretical analysis of the proposed method, focusing in particular on the convergence of the bundle supervision process. Before going into the details, we first examine the smoothness of our entropy-based objective  $\mathcal{L}_{BE}$ . More specifically, we have the following results:

**Theorem 3.2.** *Given a graph neural network  $g_\theta$ , if its corresponding first-order and second-order partial derivatives are bounded, that is,  $\|\nabla z_{i,c}(\theta)\|_\infty \leq G$  and  $\max(|\nabla^2 z_{i,c}(\theta)|) \leq M$  where  $z_{i,c}$  is the ' $c$ '-th logit of the output vector  $\mathbf{z}_i \triangleq (z_{i,1}, \dots, z_{i,C})$  provided by GNN  $g_\theta$ , then we can show that the cross-entropy loss function  $\mathcal{L}_{BE}(\theta)$  defined in Eq. 5 satisfies the following conditions:*

- i): *The cross-entropy loss function  $\mathcal{L}_{BE}(\theta)$  has a bounded gradient, i.e.,  $\|\nabla \mathcal{L}_{BE}(\theta)\|_\infty \leq \frac{2G}{|\mathcal{B}|}$  where the symbol  $|\mathcal{B}|$  represents the cardinality of bundle  $\mathcal{B}$ ;*
- ii): *The second-order partial derivatives of the cross-entropy loss function  $\mathcal{L}_{BE}(\theta)$  is also bounded, namely,  $\max(|\nabla^2 \mathcal{L}_{BE}(\theta)|) \leq \frac{2(M+G^2)}{|\mathcal{B}|}$ , which simultaneously means the loss  $\mathcal{L}_{BE}(\theta)$  is  $\left(\frac{2n_d(M+G^2)}{|\mathcal{B}|}\right)$ -smooth, that is,*

$$\|\nabla \mathcal{L}_{BE}(\theta_1) - \nabla \mathcal{L}_{BE}(\theta_2)\|_2 \leq \frac{2n_d(M+G^2)}{|\mathcal{B}|} \|\theta_1 - \theta_2\|_2,$$

where  $n_d$  is the dimension of the unknown parameter  $\theta$ .

**Remark 2.** *It is worth noting that, in Theorem 3.2, the symbol  $\max(|\mathbf{M}|)$  represents the maximum absolute value among the elements of matrix  $\mathbf{M}$ . Moreover,  $\|\cdot\|_2$  and  $\|\cdot\|_\infty$  denote the standard  $L_2$  norm and  $L_\infty$  norm, respectively.*

**Remark 3.** *Theorem 3.2 indicates that the smoothness and differentiability of the graph neural network  $g_\theta$  can, to some extent, be inherited by our adopted cross-entropy loss function  $\mathcal{L}_{BE}$ .*



With the results of Theorem 3.2, we next show that, under some mild conditions, the commonly used gradient descent algorithm for training GNN  $g_\theta$  can finally converge to a stationary point of our adopted loss function  $\mathcal{L} \triangleq (\mathcal{L}_{BE} + \mathcal{L}_R)$ . Before that, we first characterize the dynamics of the general gradient descent algorithm, namely, we suppose  $\theta_{t+1} \triangleq \theta_t - \eta \nabla \mathcal{L}_R$  where  $\eta > 0$  is the learning rate and the time  $t \in \{1, 2, \dots, T\}$ . Subsequently, we present the detailed results regarding the convergence of our adopted bundle supervision process, that is,

**Theorem 3.3.** *Under the assumptions of Theorem 3.2 and the condition  $\eta < \frac{|\mathcal{B}|}{n_d(M+G^2)}$ , if, when the iteration index  $t$  is large, the model parameter  $\theta_t$  provided by gradient descent algorithm can effectively fit the predicted bundle label  $\hat{y}^B$ , namely,  $\hat{y}^B \in \arg \max_{i \in \{1, \dots, C\}} \{\mathbf{p}_{\theta_t}(\mathcal{B})_i\}$ , then we can verify that the final obtained model parameter  $\theta_{T+1}$  will converge to a stationary point of the adopted loss function  $\mathcal{L}(\theta)$ , that is to say,  $\|\nabla \mathcal{L}(\theta_{T+1})\|_2$  can approach toward a small value as  $T \rightarrow \infty$ .*

**Remark 4.** *It is important to emphasize that when our GNN model  $g_\theta$  possesses certain structural properties, extensive research has shown that the resulting stationary point of the aforementioned gradient descent algorithm can exhibit strong generalization capabilities [30, 4] and in some cases, may even correspond to a global minimum [59, 49, 45].*

### 3.5 Bundle Refinement

During optimization of the graph neural network  $g_\theta$ , the bundle  $\mathcal{B}$  may include nodes that do not belong to the category of  $\hat{y}^B$ . To address this, we design the bundle refinement process that excludes these noisy nodes by evicting those with lower confidence in class  $\hat{y}^B$ . Specifically, given the node-level probability distribution in a bundle, *i.e.*,  $\mathbf{p}_i, i \in \mathcal{B}$ , we denote the confidence of  $\mathbf{p}_i$  with respect to class  $\hat{y}^B$  as  $\mathbf{p}_{i, \hat{y}^B}$ . We evict the less confident node in the bundle as:

$$\mathcal{B} \leftarrow \left\{ i \mid i \in \mathcal{B} \wedge \mathbf{p}_{i, \hat{y}^B} > \min_{j \in \mathcal{B}} \mathbf{p}_{j, \hat{y}^B} \right\}, \quad (9)$$

where  $\leftarrow$  denotes the update of the bundle. Bundle refinement is performed multiple times during the optimization process of  $g_\theta$ . By evicting the less confident nodes that are potentially misaligned with the bundle label, the noise in bundle supervision is further reduced. Through bundle refinement, the initial bundles, sampled via topological or semantic proximity, are dynamically adjusted during the supervision of the graph neural network to fulfill the predicted bundle label  $\hat{y}^B$  queried from the LLM, making the proposed method robust.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** In the experiments, we use ten representative datasets, *i.e.*, Cora [44], CiteSeer [16], Wikics [48], History [51], Children [51], Sportsfit [51], Cornell [10], Texas [10], Wisconsin [10], and Washington [10]. Among these datasets, Cora and CiteSeer are citation networks. Wikics is a knowledge graph derived from Wikipedia. History, Children, and Sportsfit are e-commerce networks of different types of products (*i.e.*, history books, children’s literature, sports goods). Cornell, Texas, Wisconsin, and Washington are web page networks of universities. The datasets cover both homophilic and heterophilic graphs, with the first six datasets of high homophily and the last four of low homophily.

**Compared Baselines.** We compare a spectrum of methods with our method. The compared methods include the following categories.  $\blacktriangleright$  *Text encoders*, including SBERT [57], RoBERTa [42], OpenAI’s Text-Embedding-3-Large (TE-3-Large) [52] and LLM2Vec [2].  $\blacktriangleright$  *Generative LLMs*, including GPT-3.5-turbo [1] and GPT-4o [24].  $\blacktriangleright$  *Graph self-supervised learning methods*, including DGI [62] and GraphMAE [21].  $\blacktriangleright$  *Graph foundation models or graph learning methods with LLMs*, including OFA [39], GOFA [29], UniGLM [14], ZeroG [33], GraphGPT [61], LLAGA [5], and LLM-BP [67].

**Implementation Details.** In the experiments, we use GPT-4o [24] as the default LLM for bundle query. In bundle sampling, we set the bundle size  $n_B$  as 5 and the number of bundles  $n_S$  as 100 for all datasets. We train the GNN on an NVIDIA RTX 3090 GPU.

Table 1: Prediction accuracies of our method compared to baselines across datasets. We mark the best results in **bold** and the second-best with underline.

Method	Cora	CiteSeer	WikiCS	History	Children	Sportsfit	Cornell	Texas	Wisc.	Wash.
SBERT	69.75	66.69	59.06	53.53	22.59	43.79	63.66	64.58	62.10	63.52
RoBERTa	70.71	66.95	59.08	55.39	24.25	41.51	61.68	62.25	60.33	60.60
TE-3-Large	71.90	66.24	61.78	50.15	24.68	58.39	81.50	75.42	73.14	66.35
LLM2Vec	67.34	67.13	62.34	53.14	25.56	57.00	81.26	76.68	73.36	65.92
GPT-3.5-turbo	70.11	66.83	65.53	55.07	29.73	<u>67.21</u>	45.54	56.14	58.86	51.09
GPT-4o	70.29	64.77	66.10	53.30	<u>30.76</u>	66.35	45.54	63.10	56.60	48.90
DGI	16.79	15.24	14.98	20.98	2.22	7.48	14.66	11.23	12.08	20.96
GraphMAE	15.13	8.11	8.91	36.36	7.24	30.50	23.04	17.65	23.02	24.89
OFA	20.36	41.31	30.77	8.25	3.05	15.18	29.84	11.77	4.80	6.04
GOFA	71.06	65.72	<u>68.62</u>	56.25	12.15	37.87	39.50	38.37	32.51	31.02
UniGLM	45.57	52.26	55.05	44.24	21.48	33.46	23.03	21.39	27.16	24.01
ZeroG	60.40	50.35	46.74	36.55	12.72	14.27	10.47	53.48	12.66	8.30
GraphGPT	17.48	13.93	33.59	12.31	9.94	4.53	10.18	18.48	12.35	20.64
LLAGA	11.62	19.52	10.98	7.95	10.09	1.84	12.57	15.51	15.09	10.48
LLM-BP	<u>72.59</u>	<u>69.51</u>	67.75	<u>59.86</u>	24.81	61.92	83.28	<u>81.66</u>	<u>77.75</u>	<u>73.14</u>
DENSE (ours)	<b>75.09</b>	<b>72.37</b>	<b>71.03</b>	<b>67.31</b>	<b>31.75</b>	<b>75.88</b>	<b>84.82</b>	<b>92.51</b>	<b>87.17</b>	<b>81.66</b>

## 4.2 Main Results

**Comparison with Existing Methods.** We compare our method against 15 baselines across 10 datasets in Table 1. From the results, we can see that our method consistently outperforms competitive baselines in all 10 datasets, showing the effectiveness of the proposed text bundling method. Text embedding methods (*e.g.*, SBERT, LLM2Vec) and generative LLMs (*e.g.*, GPT-4o) achieve moderate performance on many datasets. However, their ignorance of the graph topology leads to weaker performance, especially when the structures are important. Graph self-supervised learning methods (*e.g.*, DGI, GraphMAE) generally yield low accuracy without the assistance of LLMs and their strong generalization capability. For foundation models (*e.g.*, GOFA, ZeroG) that incorporate graphs in LLMs for joint training, their high performance is not consistent, worsening with graphs out of their original training distribution (*e.g.*, in university web page networks). By comparison, our method consistently outperforms baselines on various datasets covering different domains. Additionally, our method is agnostic to the specific architecture of the graph neural network, allowing us to flexibly benefit from the advancement of GNN architectures when facing different types of graph structures (*e.g.*, homophilic graphs and heterophilic graphs).

Table 2: The prediction accuracies under different LLM backbones on four datasets. The best is marked in **bold** and the second-best underline.

LLM	Cora	History	Sportsfit	Texas
GPT-4o	<u>75.09</u>	67.31	<b>75.88</b>	<u>92.51</u>
GPT-3.5-turbo	73.25	69.87	69.82	89.30
GPT-4.1-nano	70.11	<b>71.09</b>	66.11	90.37
Deepseek-V3	<b>75.28</b>	67.00	73.52	85.56
Gemini-2.5-flash	73.25	<u>70.08</u>	<u>74.98</u>	<b>93.05</b>

Table 3: Ablation studies on four datasets.

Method	Cora	History	Sportsfit	Texas
V1: R.S.	70.48	61.80	65.60	88.24
V2: I.Q.	71.96	63.95	72.61	84.49
V3: w/o $\mathcal{L}_{BE}$	70.11	64.49	65.29	91.44
V4: w/o $\mathcal{L}_R$	73.99	66.73	75.48	86.10
V5: w/ $\mathcal{L}_{IE}$	73.43	66.29	74.05	85.03
V6: w/o B.R.	73.89	66.55	73.00	91.98
DENSE (ours)	<b>75.09</b>	<b>67.31</b>	<b>75.88</b>	<b>92.51</b>

**Performance Under Different LLM Backbones.** We also show the prediction accuracies of our method using different LLMs. Specifically, we provide results on five LLMs, including GPT-4o [24] (used as default), GPT-3.5-turbo [1], GPT-4.1-nano [53], Deepseek-V3 [37], and Gemini-2.5-flash [12]. The results on four datasets (*i.e.*, Cora, History, Sportsfit, Texas) are shown in Table 2. As can be seen from the results, using alternative LLMs generally yields satisfactory performance on average. Among these LLMs, GPT-4o and Gemini-2.5-flash perform relatively better, while cheaper or older LLMs like GPT-3.5-turbo, GPT-4.1-nano, Deepseek-V3 yield decent accuracies as well. This suggests that our method can benefit from the advancement of LLMs.

## 4.3 Ablation Studies

We investigate how the different mechanisms used in our method affect the final accuracy, and we present the ablation studies in Table 3. We construct a set of variants of our method (marked as V1

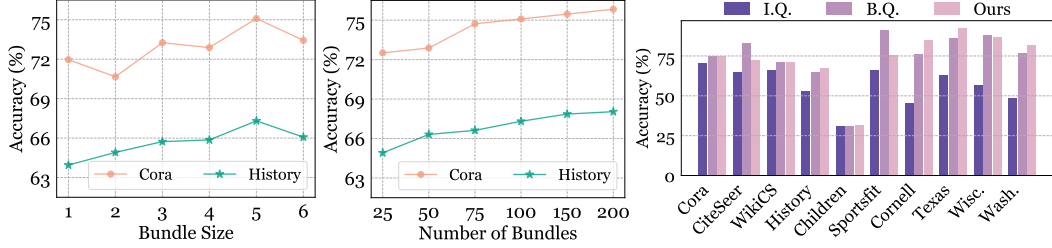


Figure 3: **Left:** prediction accuracies under different bundle sizes (*i.e.*,  $n_B$ ). **Middle:** prediction accuracies with different numbers of bundles (*i.e.*,  $n_S$ ). **Right:** accuracy comparison of individual query (I.Q.), bundle query (B.Q.), and our method (Ours).

to V6): V1 uses random sampling (R.S.) instead of topological proximity or semantic proximity to obtain bundles. V2 uses individual query (I.Q.), asking the LLM about the category of each node with the text attribute. V3 removes the entropy-based loss  $\mathcal{L}_{BE}$ . V4 removes the ranking-based loss  $\mathcal{L}_R$ . V5 uses individual supervision, *i.e.*,  $\mathcal{L}_{IE}$  defined in Theorem 3.1. V6 does not employ bundle refinement. As can be seen from the results, each technique proposed is helpful for the overall accuracy, and removing them causes performance degradation. Additionally, we find that bundle sampling is important, especially when the number of classes is large (in this case, 12 classes for History and 13 classes for Sportsfit, both of which witness a severe drop in accuracy with random bundle sampling). One explanation is that inappropriate sampling causes the nodes in a bundle to be more uniformly distributed across various categories, making it difficult to decide the bundle class (with a weaker mode category) and perform bundle supervision (with noisier bundle labels). Moreover, we find that individual supervision ( $\mathcal{L}_{IE}$ ) is weaker than bundle supervision, which suggests that our supervision method is more tolerant to bundle outliers.

#### 4.4 Hyperparameter Analysis

**Effects of Bundle Size  $n_B$ .** We also show the prediction accuracies using different bundle sizes  $n_B$  in Figure 3 (Left). As can be seen from the figure, setting  $n_B$  to 5 achieves relatively good performance. Smaller bundle sizes make it difficult to provide sufficient information about the neighborhood, while larger bundle sizes have the risk of including categories other than the mode category. Moreover, we observe that odd bundle sizes are relatively better than even ones. A possible explanation is that even bundles are more likely to have ties (*e.g.*, two nodes from class A and two from class B in a four-node bundle), causing confusion and noisy supervision. This is less significant in datasets with a larger number of classes (*e.g.*, the History dataset with 12 classes), where odd bundles may also include a node from the third class, failing to resolve the ties.

**Effects of the Number of Bundles  $n_S$ .** We also show the prediction accuracies with different numbers of bundles in Figure 3 (Middle). As can be seen from the figure, larger numbers of bundles generally yield better accuracy. As one bundle corresponds to one query of the LLM, increasing the number of bundles indicates additional feedback from the LLM, leading to additional supervision signals that better cover the graph topology and bring robustness from redundancy. Nevertheless, additional queries incur additional computation resources. Balancing performance and cost from querying LLMs, we set the number of bundles  $n_S$  to 100 in the experiments.

#### 4.5 Further Analysis

**Bundle Query v.s. Individual Query.** We then show that the LLM can better handle bundle queries than individual queries. Specifically, we provide the LLM’s classification accuracies given individual queries (marked as I.Q.) and bundle queries (marked as B.Q.) in Figure 3 (Right). From the figure, we can see that predicting the main category of the text bundles is generally easier than classifying individual text items, and in some datasets (*e.g.*, CiteSeer, Cornell), the improvement is fairly large. We also show the overall prediction accuracies of our method, and we can see a general connection between the improvement of bundle queries and our method (compared to individual queries). This shows that the proposed text bundling method increases the reliability and robustness of supervision signals from LLMs and thereby improves the overall performance.



(a) Prompt Template:

We have <node\_description> in  
<graph\_description> from the following  
<number\_of\_classes> categories: [<category\_1>,  
<category\_2>, ..., <category\_C>]  
Below are texts from <bundle\_size>  
<node\_refer\_name\_plural>.  
[<node\_refer\_name\_singular> 1]  
<text\_attribute\_of\_node\_1>  
[<node\_refer\_name\_singular> 2]  
<text\_attribute\_of\_node\_2>  
...  
Please tell me the main category that most of the papers  
belong to. Think carefully and then provide the final  
answer in the last line, using the format: "Final Answer:  
<category>".

(b) Prompt Example (CiteSeer):

We have description or opening text of scientific  
publications in a citation network from the following 6  
categories: ['Agents', 'machine learning (ML)',  
'information retrieval (IR)', 'database (DB)', 'human-  
computer interaction (HCI)', 'artificial intelligence (AI)']  
Below are texts from 5 papers.  
[Paper 1]  
Extracting Query Modifications from Nonlinear SVMs...  
[Paper 2]  
Context and Page Analysis for Improved Web Search...  
...  
Please tell me the main category that most of the papers  
belong to. Think carefully and then provide the final  
answer in the last line, using the format: "Final Answer:  
<category>".

(c) LLM's Response (CiteSeer, GPT-4o):

To determine the main category that most of the papers belong to, let's analyze the content of each paper and try to  
classify it according to the given 6 categories:  
Paper 1: Extracting Query Modifications from Nonlinear SVMs. The focus is on query modifications and using support  
vector machines (SVMs) to improve precision and recall in search. SVMs are a common tool in machine learning, and  
the paper discusses dimensionality reduction and feature space, which fit within the machine learning domain. It also  
talks about query modifications which slightly implies information retrieval (IR). Paper 2: Context and Page Analysis  
for Improved Web Search. Discusses the development of a metasearch engine to improve web search efficiency and  
precision by analyzing query contexts. The main focus is on handling web search limitations which aligns with  
information retrieval (IR). Paper 3: ... Paper 4... Paper 5: ... (details omitted for paper 3, 4, and 5)  
Analyzing all papers, while machine learning techniques are frequently used, the persistent theme across most papers  
revolves around improving search engines, query modifications, and personalized search, which are strongly  
indicative of the "information retrieval (IR)" category.  
Final Answer: information retrieval (IR)

Figure 4: The prompt template of bundle query (a), an example of the prompt on the CiteSeer dataset (b), and an example of the response of GPT-4o to the query (c).

**Prompt Examples and the LLM's Response.** We also provide the prompt template, an example of the prompt, and the LLM's response in Figure 4. In the prompt, we provide information about the nodes and the graphs. We then ask the LLM to find the main category that most of the papers in the text bundle belong to. For the LLM's response, we can see that although machine learning is a frequent topic of research among the papers in the bundle, the LLM discovers a "*persistent theme across most papers*" to be strongly related to information retrieval. Without text bundling, the LLM may hesitate between machine learning and information retrieval when classifying Paper 1, as its analysis suggests that this paper "fits within the machine learning domain" and also "slightly implies information retrieval". Such ambiguity would cause noise in classification results and be harmful for potential subsequent operations (e.g., clustering, supervision of GNNs). By comparison, our method allows the LLM to obtain more information, finding a persistent theme that represents most text items in the bundle, improving the reliability of LLM's response.

## 5 Conclusion

This paper investigates the important problem of zero-shot inference on text-attributed graphs with the help of LLMs. While previous efforts suffer from limited information on graph structure and unreliable responses, this paper proposes a novel method named dynamic text bundling supervision that queries the LLM with text bundles to obtain bundle-level labels. Subsequently, the bundle labels are used to supervise a graph neural network, which is then used for classification. We provide theoretical analysis of our method, showing its tolerance of outlier nodes in the bundle and the convergence properties of optimization. We further refine the nodes in the bundle to exclude noisy items. Extensive experiments are performed on ten datasets across different domains against a number of competing baselines, and the results confirm the effectiveness of the proposed method.

**Limitations and Broader Impacts.** This paper focuses on text-attributed graphs, where each node is associated with a textual attribute. For graphs where node attributes are hard for LLMs to understand, the proposed text bundling method is not directly applicable. For graph structures on which GNNs are inherently weak or inferior to alternatives, this method may not be directly applicable. As for broader

impacts, the proposed text bundling method improves the zero-shot inference ability of LLMs on text-attributed graphs, facilitating downstream applications in many fields, including social network analysis, recommendation systems, web page analysis, and knowledge graph understanding.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.
- [3] Dan Berrebbi, Nicolas Huynh, and Oana Balalau. Graphcite: Citation intent classification in scientific publications via graph embeddings. In *Companion proceedings of the web conference 2022*, pages 779–783, 2022.
- [4] Thijs Bos and Johannes Schmidt-Hieber. Convergence rates of deep relu networks for multiclass classification. *Electronic Journal of Statistics*, 16(1):2724–2773, 2022.
- [5] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024.
- [6] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems*, 22, 2009.
- [7] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61, 2024.
- [8] Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, et al. Text-space graph foundation models: Comprehensive benchmarks and new insights. *arXiv preprint arXiv:2406.10727*, 2024.
- [9] Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*, 2023.
- [10] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the world wide web. *AAAI/IAAI*, 3(3.6):2, 1998.
- [11] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4908–4922, 2022.
- [12] Google DeepMind. Gemini 2.5 flash model. <https://deepmind.google/technologies/gemini/#gemini-25>, 2024.
- [13] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 295–304, 2020.
- [14] Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. Uniglrm: Training one unified language model for text-attributed graphs. *arXiv preprint arXiv:2406.12052*, 2024.
- [15] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.
- [16] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.

- [17] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [18] Tao Guo and Baojiang Cui. Web page classification based on graph neural network. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 188–198. Springer, 2021.
- [19] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*, 2023.
- [20] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. *arXiv preprint arXiv:2402.13630*, 2024.
- [21] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 594–604, 2022.
- [22] Xuanwen Huang, Kaiqiao Han, Dezheng Bao, Quanjin Tao, Zhisheng Zhang, Yang Yang, and Qi Zhu. Prompt-based node feature extractor for few-shot learning on text-attributed graphs. *arXiv preprint arXiv:2309.02848*, 2023.
- [23] Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Gnns as adapters for llms on text-attributed graphs. In *The Web Conference 2024*, 2024.
- [24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [25] Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–30, 2024.
- [26] Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. Empowering time series analysis with large language models: A survey. *arXiv preprint arXiv:2402.03182*, 2024.
- [27] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [29] Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. Gofa: A generative one-for-all model for joint graph language modeling. *arXiv preprint arXiv:2407.09709*, 2024.
- [30] Sanjukta Krishnagopal and Luana Ruiz. Graph neural tangent kernel: Convergence on large graphs. In *International Conference on Machine Learning*, pages 17827–17841. PMLR, 2023.
- [31] Quan Li, Xiaoting Li, Lingwei Chen, and Dinghao Wu. Distilling knowledge on text graph for social media attribute inference. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2024–2028, 2022.
- [32] Rui Li, Jiwei Li, Jiawei Han, and Guoyin Wang. Similarity-based neighbor selection for graph llms. *arXiv preprint arXiv:2402.03720*, 2024.
- [33] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1725–1735, 2024.

- [34] Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor W Chan, and Jia Li. Glbench: A comprehensive benchmark for graph with large language models. *Advances in Neural Information Processing Systems*, 37:42349–42368, 2024.
- [35] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2257–2270, 2018.
- [36] Tianqianjin Lin, Pengwei Yan, Kaisong Song, Zhuoren Jiang, Yangyang Kang, Jun Lin, Weikang Yuan, Junjie Cao, Changlong Sun, and Xiaozhong Liu. Langgfm: A large language model alone can be a powerful graph foundation model. *arXiv preprint arXiv:2410.14961*, 2024.
- [37] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- [38] Han Liu, Siyang Zhao, Xiaotong Zhang, Feng Zhang, Wei Wang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. Liberating seen classes: boosting few-shot and zero-shot text classification via anchor generation and classification reframing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18644–18652, 2024.
- [39] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*, 2023.
- [40] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [41] Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick PK Chan, Daniel S Yeung, and Pheng-Ann Heng. Item relationship graph neural networks for e-commerce. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4785–4799, 2021.
- [42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [43] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Forty-first International Conference on Machine Learning*, 2024.
- [44] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000.
- [45] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.
- [46] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. *Advances in neural information processing systems*, 32, 2019.
- [47] Aditya Krishna Menon, Xiaoqian J Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *Proceedings of the... International Conference on Machine Learning. International Conference on Machine Learning*, volume 2012, page 703, 2012.
- [48] Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.
- [49] Alireza Mousavi-Hosseini, Denny Wu, and Murat A Erdogdu. Learning multi-index models with neural networks via mean-field langevin dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025.

- [50] David F Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013.
- [51] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.
- [52] OpenAI. GPT text-embedding-3-large. <https://platform.openai.com/docs/guides/embeddings>, 2024.
- [53] OpenAI. Gpt-4.1-nano model. <https://platform.openai.com/docs/models/gpt-4.1-nano>, 2025.
- [54] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024.
- [55] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102, 2023.
- [56] Filippo Radicchi, Santo Fortunato, and Alessandro Vespignani. Citation networks. *Models of science dynamics: Encounters between complexity theory and information sciences*, pages 233–257, 2011.
- [57] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
- [58] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.
- [59] Marek Rychlik. A proof of convergence of multi-class logistic regression network. *arXiv preprint arXiv:1903.12600*, 2019.
- [60] Shengyin Sun, Yuxiang Ren, Chen Ma, and Xuechang Zhang. Large language models as topological structure enhancers for text-attributed graphs. *arXiv preprint arXiv:2311.14324*, 2023.
- [61] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.
- [62] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.
- [63] Vikas Verma, Meng Qu, Kenji Kawaguchi, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Improved training of gnns for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10024–10032, 2021.
- [64] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.
- [65] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.
- [66] Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in Neural Information Processing Systems*, 37:5950–5973, 2024.
- [67] Haoyu Wang, Shikun Liu, Rongzhe Wei, and Pan Li. Model generalization on text attribute graphs: Principles with large language models. *arXiv preprint arXiv:2502.11836*, 2025.



- [68] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [69] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5207–5216, 2019.
- [70] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14557–14565, 2023.
- [71] Han Xie, Da Zheng, Jun Ma, Houyu Zhang, Vassilis N Ioannidis, Xiang Song, Qing Ping, Sheng Wang, Carl Yang, Yi Xu, et al. Graph-aware language model pre-training on a large graph corpus can help multiple graph applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5270–5281, 2023.
- [72] Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.
- [73] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [74] Tong Ye, Yangkai Du, Tengfei Ma, Lingfei Wu, Xuhong Zhang, Shouling Ji, and Wenhai Wang. Uncovering llm-generated code: A zero-shot synthetic code detector via code rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 968–976, 2025.
- [75] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuechang Zhang. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13087–13095, 2025.
- [76] Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1298–1301, 2024.
- [77] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In *Proceedings of the ACM Web Conference 2024*, pages 1003–1014, 2024.
- [78] Zhen Zhang, Meihan Liu, Anhui Wang, Hongyang Chen, Zhao Li, Jiajun Bu, and Bingsheng He. Collaborate to adapt: Source-free graph domain adaptation via bi-directional adaptation. In *Proceedings of the ACM Web Conference 2024*, pages 664–675, 2024.
- [79] Zheng Zhang, Yuntong Hu, Bo Pan, Chen Ling, and Liang Zhao. Taga: Text-attributed graph self-supervised learning by synergizing graph and text mutual transformations. *arXiv preprint arXiv:2405.16800*, 2024.
- [80] Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Chenhui Zhang, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. Pre-training and prompting for few-shot node classification on text-attributed graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4467–4478, 2024.
- [81] Zengqun Zhao, Yu Cao, Shaogang Gong, and Ioannis Patras. Enhancing zero-shot facial expression recognition by llm knowledge transfer. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 815–824. IEEE, 2025.

- [82] Yilun Zheng, Sitao Luan, and Lihui Chen. What is missing for graph homophily? disentangling graph homophily for graph neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [83] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11168–11176, 2021.
- [84] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.
- [85] Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025.
- [86] Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pages 2183–2197, 2025.