Dynamic Bundling with Large Language Models for Zero-Shot Inference on Text-Attributed Graphs

Yusheng Zhao[1], Qixin Zhang[2], Xiao Luo[4]*, Weizhi Zhang[5], Zhiping Xiao[3]*, Wei Ju[1], Philip S. Yu[5], Ming Zhang[1]*

[1] State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University,
[2] College of Computing and Data Science, Nanyang Technological University,
[3] Paul G. Allen School of Computer Science and Engineering, University of Washington
[4] Department of Computer Science, University of California, Los Angeles,
[5] Department of Computer Science, 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. Our code is available at https://github.com/YushengZhao/bundle-neurips25.

1 Introduction

Text-attributed graphs (TAGs) [83, 87] 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 representations in many applications, including citation networks [67], social networks [61], e-commerce networks [48], and webpage networks [11]. As complete labeling of these large networks is often time-consuming and costly, efforts have been made to utilize semi-supervised learning [33, 74, 51], transfer learning [12, 102, 95, 52], and few-shot/zero-shot learning [14, 91, 86] to understand text-attributed graphs with limited labels.

Large language models (LLMs) [84, 44] have been observed to exhibit strong zero-shot generalization capability, enhancing the performance on various types of data, including visual signals [97], audio

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

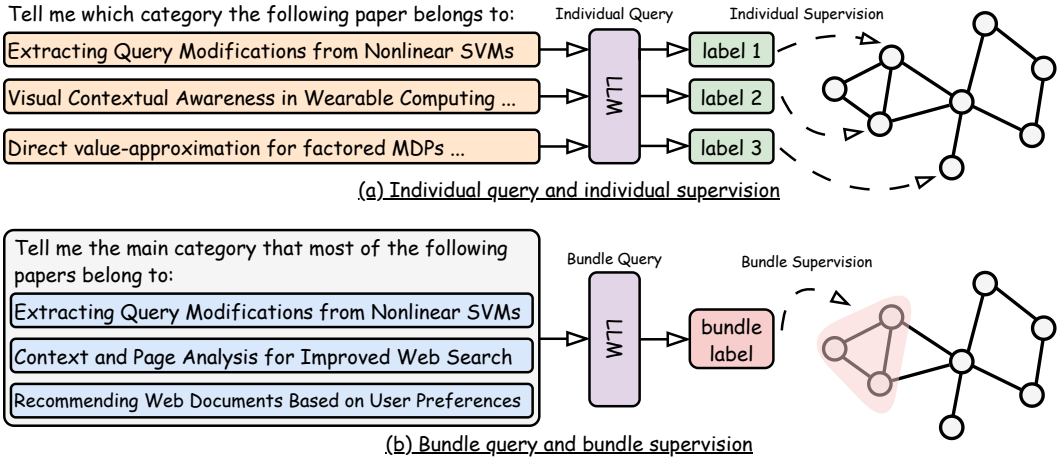

Figure 1: (a) Querying LLMs with individual texts and supervising graph learning with individual labels. (b) By creating text bundles, we perform bundle queries to obtain bundle labels for supervision.

signals [92], texts [45], programming code [85, 94], time series [93], and graphs [76]. Recently, there have been efforts in integrating LLMs in text-attributed graphs [20, 25, 101] for zero-shot inference. One line of research integrates the graph topology into language models [9, 53, 101], converting non-Euclidean topology into a sequence of tokens. However, building such foundation models requires a large amount of data [77], and the conversion to Euclidean data inevitably incurs information loss [47]. Another line of research directly utilizes the zero-shot generalization ability of existing LLMs to understand node attributes [8, 38, 77], and utilizes the output of LLMs as supervision signals for training graph neural networks (GNNs) [33] or as clustering centers [77]. 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 is illustrated in Figure 1, conventional methods [10, 77] query LLMs with individual text items (for example, 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 in the bundle) 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). We perform both theoretical and empirical studies to demonstrate the effectiveness of our method.

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.

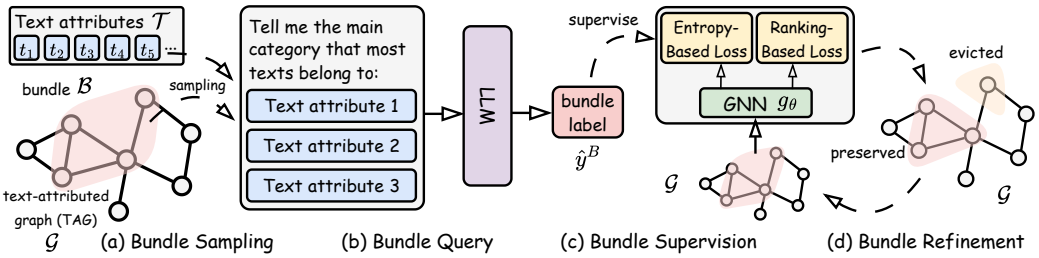

Figure 2: The overall framework of our method. We first sample nodes of proximity to form bundles (a), which are then used to query the LLM about their main categories (b). Subsequently, the bundle labels from the LLM's response are used to supervise a graph neural network (c). During optimization, we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

2 Related Works

Text-Attributed Graphs. Text-attributed graphs (TAGs) are a special type of graphs whose nodes are associated with textual attributes [83, 87]. They are common forms of data in many fields, such as citation networks [4], knowledge graphs [66], social networks [37], web page networks [19], *etc.* Research on TAGs generally focuses on combining textual attributes with graph structures, with the help of text embedding methods [75, 56] and network embedding methods [78, 42, 31]. As the annotation costs of TAGs are usually high, efforts have been made in semi-supervised learning [50, 90], transfer learning [89, 21, 102, 96], and few-shot learning [24, 91, 86]. With the advancement of large language models, this work makes a step further, focusing on the zero-shot inference of text-attributed graphs [10, 76, 77] with the help of LLMs.

Large Language Model for Graphs. Large language models (LLMs) [84, 44] have shown impressive performance in understanding data beyond natural languages, including programming languages [28], sequences of numbers [29], mathematics [69], and graphs [10, 30, 65, 41]. 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 [81, 16, 43, 76]. These methods often require training to align the graph structure and natural language [88, 102], 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 [71, 10, 8, 77]. However, they often use isolated nodes [77] or explicit descriptions that are hard for LLMs to understand [72]. 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 $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{Y} \rangle$, 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_θ, *i.e.*, $\boldsymbol{x}_i = f_\theta(t_i) \in \mathbb{R}^d$. 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 [33, 18]. 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:

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

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 [100, 99, 98]. Therefore, we turn to semantic proximity utilizing vectorized representations of nodes. Specifically, given embeddings of each node $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ and a core node v_c with corresponding embedding $\boldsymbol{x}_c$, we construct the node bundle based on the closeness in the embedding space $\mathbb{R}^d$:

$$\mathcal{B} = \left\{ i \mid \boldsymbol{x}_i \in \mathcal{N}_{\mathcal{X}}^{n_B}(\boldsymbol{x}_c) \right\}, \tag{2}$$

where $\mathcal{N}_{\mathcal{X}}^{n_B}(\boldsymbol{x}_c)$ denotes the set of top n_B vectors in $\mathcal{X}$ that are closest to 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 $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_S}\}$, 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:

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

where the 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 $\hat{y}^B$.

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_θ, and it generates probability distributions for each node as:

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

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

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

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

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 (with the proof in Appendix A):

Theorem 3.1. *Given a bundle $\mathcal{B}$, its corresponding bundle class distribution $\boldsymbol{p}(\mathcal{B}) = (p_1, p_2, \ldots, p_C)$, an outlier node $v_o, o \in \mathcal{B}$ with probability distribution $\boldsymbol{p}_o = (p'_1, p'_2, \ldots, p'_C)$, 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:*

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

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 [7, 57, 79], and design a ranking-based loss as follows:

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

where $p(\mathcal{B})_i \in \mathbb{R}$ denotes the i-th component of vector $p(\mathcal{B})$ (*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_θ, 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 $\boldsymbol{p}(\mathcal{B})_{\hat{y}^B}$ (which does not necessitate all $\boldsymbol{z}_i, i \in \mathcal{B}$ 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:

$$\mathcal{L} = \mathcal{L}_{BE} + \mathcal{L}_R. \tag{8}$$

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 (with the proof in Appendix B):

Theorem 3.2. *Given a graph neural network g_θ, if its corresponding first-order and second-order partial derivatives are bounded, that is, $\|\nabla z_{i,c}(\theta)\|_\infty \leq G$ and $\max(\left|\nabla^2 z_{i,c}(\theta)\right|) \leq M$ where $z_{i,c}$ is the 'c'-th logit of the output vector $\boldsymbol{z}_i \triangleq (z_{i,1}, \ldots, z_{i,C})$ provided by GNN g_θ, 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, $\max\left(\left|\nabla^2\mathcal{L}_{BE}(\theta)\right|\right) \leq \frac{2(M+G^2)}{|\mathcal{B}|}$, 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* θ*.*

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_∞ *norm, respectively.*

Remark 3. *Theorem 3.2 indicates that the smoothness and differentiability of the graph neural network* g_θ *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_θ 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 $t \in \{1, 2, \ldots, T\}$. Subsequently, we present the detailed results regarding the convergence of our adopted bundle supervision process, that is,

Theorem 3.3 (Proof is deferred to Appendix C)**.** *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* θ_t *provided by gradient descent algorithm can effectively fit the predicted bundle label* $\hat{y}^B$*, namely,* $\hat{y}^B \in \arg\max_{i\in\{1,\ldots,C\}}\{\boldsymbol{p}_{\theta_t}(\mathcal{B})_i\}$*, then we can verify that the final obtained model parameter* θ_{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* $T \to \infty$*.*

Remark 4. *The proof of Theorem 3.3 builds upon the standard non-convex optimization frameworks [27, 2, 36]. Moreover, it is important to emphasize that when our GNN model* g_θ *possesses certain structural properties, extensive research has shown that the resulting stationary point of the aforementioned gradient descent algorithm can exhibit strong generalization capabilities [35, 5] and in some cases, may even correspond to a global minimum [70, 59, 55].*

3.5 Bundle Refinement

During optimization of the graph neural network g_θ, 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.*, $\boldsymbol{p}_i, i \in \mathcal{B}$, we denote the confidence of $\boldsymbol{p}_i$ with respect to class $\hat{y}^B$ as $\boldsymbol{p}_{i,\hat{y}^B}$. We evict the less confident node in the bundle as:

$$\mathcal{B} \leftarrow \left\{ i \mid i \in \mathcal{B} \ \wedge \ \boldsymbol{p}_{i,\hat{y}^B} > \min_{j\in\mathcal{B}} \boldsymbol{p}_{j,\hat{y}^B} \right\}, \tag{9}$$

where $\leftarrow$ denotes the update of the bundle. Bundle refinement is performed multiple times during the optimization process of g_θ. 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. The overall algorithm is provided in Appendix D.

4 Experiments

4.1 Experimental Setup

Datasets. In the experiments, we use ten representative datasets, *i.e.*, Cora [54], CiteSeer [17], Wikics [58], History [62], Children [62], Sportsfit [62], Cornell [11], Texas [11], Wisconsin [11], and Washington [11]. 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. More details about the datasets can be found in Appendix E.

Compared Baselines. We compare a spectrum of methods with our method. The compared methods include the following categories. ▶ *Text encoders*, including SBERT [68], RoBERTa [49], OpenAI's Text-Embedding-3-Large (TE-3-Large) [63] and LLM2Vec [3]. ▶ *Generative LLMs*, including GPT-3.5-turbo [1] and GPT-4o [26]. ▶ *Graph self-supervised learning methods*, including DGI [73] and GraphMAE [23]. ▶ *Graph foundation models or graph learning methods with LLMs*, including OFA [46], GOFA [34], UniGLM [15], ZeroG [40], GraphGPT [72], LLAGA [6], and LLM-BP [77]. More details about the baseline methods can be found in Appendix F.

Implementation Details. In the experiments, we use GPT-4o [26] 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. For homophilic graphs, we use topological proximity for sampling and GCN [33] as the default GNN, whereas semantic proximity and GloGNN [39] are used in heterophilic graphs. We train the GNN on an NVIDIA RTX 3090 GPU for 500 epochs, and bundle refinement is performed at the 300th and 400th epochs. More details of our implementation can be found in Appendix G.

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	67.21	45.54	56.14	58.86	51.09
GPT-4o	70.29	64.77	66.10	53.30	30.76	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	68.62	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	72.59	69.51	67.75	59.86	24.81	61.92	83.28	81.66	77.75	73.14
DENSE (ours)	**75.09**	**72.37**	**71.03**	**67.31**	**31.75**	**75.88**	**84.82**	**92.51**	**87.17**	**81.66**

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).

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 [26] (used as default), GPT-3.5-turbo [1], GPT-4.1-nano [64], Deepseek-V3 [44], and Gemini-2.5-flash [13]. 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

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	75.09	67.31	**75.88**	92.51
GPT-3.5-turbo	73.25	69.87	69.82	89.30
GPT-4.1-nano	70.11	**71.09**	66.11	90.37
Deepseek-V3	**75.28**	67.00	73.52	85.56
Gemini-2.5-flash	73.25	70.08	74.98	**93.05**

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)	**75.09**	**67.31**	**75.88**	**92.51**

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 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 Theroem 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.

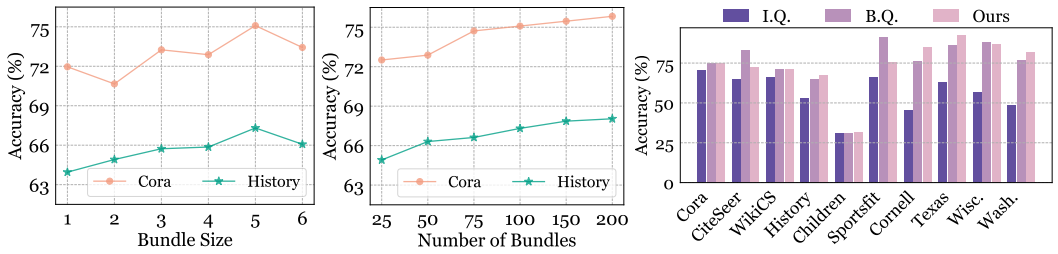

Figure 3: **Left**: prediction accuracies under different bundle sizes (*i.e.*, n_B). **Middle**: prediction accuracies with different numbers of bundles (*i.e.*, n_S). **Right**: accuracy comparison of individual query (I.Q.), bundle query (B.Q.), and our method (Ours).

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

(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_ploral>.
[<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).

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.

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 Discussions

Decision of Graph Homophily. In Section 3.2, we provide two approaches for bundle sampling: topological proximity and semantic proximity, depending on the graph homophily. In practice, when homophily is not obvious from the graph's metadata, it is non-trivial to determine which sampling technique to adopt. Therefore, we introduce an approach initially proposed by Wang et al. [77] that estimates the graph homophily by querying the LLM with text pairs of adjacent nodes. The LLM is asked to determine whether they belong to the same category. The ratio of positive response (*i.e.*, "Yes") can be used as an indicator of graph homophily. In their paper, Wang et al. [77] show that it is possible to effectively approximate the homophily degree with limited queries of less powerful LLMs (*e.g.*, GPT-4o-mini). Other potentially useful techniques include computing the cosine similarity of feature pairs from adjacent nodes instead of querying LLMs.

Potential Extension to Non-Text-Attributed Graphs. The proposed framework can be extended to a more general setting where text attributes are not provided. Non-text-attributed graphs can be converted to text-attributed graphs, which have been explored by Wang et al. [80]. By augmenting non-text-attributed graphs with textual descriptions, we can apply the proposed DENSE framework for zero-shot inference on these graphs.

Bundle Refinement Configurations. In Section 3.5 and Section 4.1, we mention that the bundle refinement is performed at the 300th and 400th epochs, evicting one item each time. We make a further explanation of these heuristic values: we want to refine the bundle multiple times, and as the optimal bundle size is 5 (as empirically demonstrated in Section 4.4), evicting one item seems a reasonable choice (evicting more than one item will significantly reduce the richness of information in the bundle). As for the time of refinement, we observe that the model generally converges at the 300th epoch, and after refinement, it usually converges within another 100 epochs.

Limitations. 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.

Broader Impacts. 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.

6 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.

Acknowledgement

Ming Zhang and Yusheng Zhao are supported by grants from the National Key Research and Development Program of China with Grant No. 2023YFC3341203 and the National Natural Science Foundation of China (NSFC Grant Number 62276002). The authors are grateful to the anonymous reviewers for critically reading this article and for giving important suggestions to improve this article.

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] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

[3] 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.

[4] 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.

[5] Thijs Bos and Johannes Schmidt-Hieber. Convergence rates of deep relu networks for multiclass classification. *Electronic Journal of Statistics*, 16(1):2724–2773, 2022.

[6] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Google DeepMind. Gemini 2.5 flash model. https://deepmind.google/technologies/gemini/#gemini-25, 2024.

[14] 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.

[15] Yi Fang, Dongzhe Fan, Sirui Ding, Ninghao Liu, and Qiaoyu Tan. Uniglm: Training one unified language model for text-attributed graphs. *arXiv preprint arXiv:2406.12052*, 2024.

[16] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–363, 2017.

[28] 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.

[29] 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.

[30] 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.

[31] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, page 106207, 2024.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[34] 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.

[35] Sanjukta Krishnagopal and Luana Ruiz. Graph neural tangent kernel: Convergence on large graphs. In *International Conference on Machine Learning*, pages 17827–17841. PMLR, 2023.

[36] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

[37] 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.

[38] Rui Li, Jiwei Li, Jiawei Han, and Guoyin Wang. Similarity-based neighbor selection for graph llms. *arXiv preprint arXiv:2402.03720*, 2024.

[39] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*, pages 13242–13256. PMLR, 2022.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] Xiao Luo, Yusheng Zhao, Zhengyang Mao, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. Rignn: A rationale perspective for semi-supervised open-world graph classification. *Transactions on Machine Learning Research*, 2023.

[51] Xiao Luo, Yusheng Zhao, Yifang Qin, Wei Ju, and Ming Zhang. Towards semi-supervised universal graph classification. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):416–428, 2024.

[52] Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Bei Wu, Ziyue Qiao, Chenwei Tang, and Jiancheng Lv. Pala: Class-imbalanced graph domain adaptation via prototype-anchored learning and alignment. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 3198–3207, 2025.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

[59] 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.

[60] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[61] David F Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013.

[62] 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.

[63] OpenAI. GPT text-embedding-3-large. https://platform.openai.com/docs/guides/embeddings, 2024.

[64] OpenAI. Gpt-4.1-nano model. https://platform.openai.com/docs/models/gpt-4.1-nano, 2025.

[65] 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.

[66] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102, 2023.

[67] Filippo Radicchi, Santo Fortunato, and Alessandro Vespignani. Citation networks. *Models of science dynamics: Encounters between complexity theory and information sciences*, pages 233–257, 2011.

[68] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[69] 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.

[70] Marek Rychlik. A proof of convergence of multi-class logistic regression network. *arXiv preprint arXiv:1903.12600*, 2019.

[71] Shengyin Sun, Yuxiang Ren, Chen Ma, and Xuecang Zhang. Large language models as topological structure enhancers for text-attributed graphs. *arXiv preprint arXiv:2311.14324*, 2023.

[72] 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.

[73] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[74] 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.

[75] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. Can llms convert graphs to text-attributed graphs? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1412–1432, 2025.

[81] 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.

[82] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

[83] 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.

[84] 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.

[85] 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.

[86] Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] Yusheng Zhao, Junyu Luo, Xiao Luo, Weizhi Zhang, Zhiping Xiao, Wei Ju, Philip S Yu, and Ming Zhang. Multifaceted evaluation of audio-visual capability for mllms: Effectiveness, efficiency, generalizability and robustness. *arXiv preprint arXiv:2504.16936*, 2025.

[93] Yusheng Zhao, Xiao Luo, Haomin Wen, Zhiping Xiao, Wei Ju, and Ming Zhang. Embracing large language models in traffic flow forecasting. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria*, 2025.

[94] Yusheng Zhao, Xiao Luo, Weizhi Zhang, Wei Ju, Zhiping Xiao, Philip S Yu, and Ming Zhang. Marco: Meta-reflection with cross-referencing for code reasoning. *arXiv preprint arXiv:2505.17481*, 2025.

[95] Yusheng Zhao, Changhu Wang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. Traci: A data-centric approach for multi-domain generalization on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13401–13409, 2025.

[96] Yusheng Zhao, Qixin Zhang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. Test-time adaptation on graphs via adaptive subgraph-based selection and regularized prototypes. In *Forty-second International Conference on Machine Learning*, 2025.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. For details, please refer to Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: In Appendix A, B and C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: In Appendix G and the code url.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Repeated experiments over all baselines are costly.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In Section 4.1 and Appendix G.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The LLM is used in our algorithm.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Proof of Theorem 3.1

Proof. For a bundle $\mathcal{B}$, we denote its (predicted) bundle-level class probability distribution as $\boldsymbol{p} \triangleq \boldsymbol{p}(\mathcal{B}) = (p_1, p_2, \ldots, p_C)$, with corresponding logits as $\boldsymbol{z}$. Each node in the bundle corresponds to a predicted logit vector $\boldsymbol{z}_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,C}), i \in \mathcal{B}$. The bundle-level logits can be written as:

$$\boldsymbol{z} = (z_1, z_2, \ldots, z_C) = \left(\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,1}, \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,2}, \ldots \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,C} \right)$$

For an outlier node v_o in the bundle, it has the (predicted) class probability distribution $\boldsymbol{p}_o = (p'_1, p'_2, \ldots, p'_C)$. We also denote their corresponding logits as $\boldsymbol{z}_o = (z'_1, z'_2, \ldots, z'_C)$, where $\boldsymbol{p}_o = \mathrm{softmax}(\boldsymbol{z}_o)$. The most likely class for v_o is $m' \triangleq \mathrm{argmax}_i \{p'_i\}_{i=1}^C$. The bundle label predicted by the LLM $\hat{y}$ has its one-hot form of $\boldsymbol{y} = (y_1, y_2, \ldots, y_C)$. The bundle supervision loss $\mathcal{L}_{BE}$ and individual supervision loss $\mathcal{L}_{IE}$ can be written as:

$$\begin{aligned}
\mathcal{L}_{BE} &= -\sum_{i=1}^{C} y_i \log p_i = -\sum_{i=1}^{C} y_i \log \left(\frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_i)} \right) = -\sum_{i=1}^{C} y_i z_i + \log \sum_{j=1}^{C} \exp(z_i), \\
\mathcal{L}_{IE} &= -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{C} y_i \log p'_i = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{C} y_i \log \left(\frac{\exp(z'_i)}{\sum_{j=1}^{C} \exp(z'_i)} \right) \\
&= -\frac{1}{|\mathcal{B}|} \left(\sum_{i=1}^{C} y_i z'_i - \log \sum_{j=1}^{C} \exp(z'_i) \right).
\end{aligned}$$

We can then calculate the derivatives as follows. For $\mathcal{L}_{BE}$, we have:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{BE}}{\partial z'_i} &= \frac{\partial \mathcal{L}_{BE}}{\partial z_i} \cdot \frac{\partial z_i}{\partial z'_i} = \frac{1}{|\mathcal{B}|} \cdot \frac{\partial}{\partial z_i} \left(-\sum_{i=1}^{C} y_i z_i + \log \sum_{i=1}^{C} \exp(z_i) \right) \\
&= \frac{1}{|\mathcal{B}|} \left(-y_i + \frac{\exp(z_i)}{\sum_{i=1}^{C} \exp(z_i)} \right) = \frac{1}{|\mathcal{B}|} (p_i - y_i).
\end{aligned} \tag{10}$$

Similarly, we can calculate the derivative of $\mathcal{L}_{IE}$ as follows:

$$\frac{\partial \mathcal{L}_{IE}}{\partial z'_i} = \frac{1}{|\mathcal{B}|} \cdot \frac{\partial}{\partial z'_i} \left(-\sum_{i=1}^{C} y_i z'_i + \log \sum_{j=1}^{C} \exp(z'_i) \right) = \frac{1}{|\mathcal{B}|} (p'_i - y_i).$$

For $i = m'$, we have:

$$\frac{\partial \mathcal{L}_{BE}}{\partial z'_{m'}} = \frac{1}{|\mathcal{B}|} (p_{m'} - y_{m'}), \quad \frac{\partial \mathcal{L}_{IE}}{\partial z'_{m'}} = \frac{1}{|\mathcal{B}|} (p'_{m'} - y_{m'}).$$

Given the condition $\hat{y} \neq m$ and $p'_{m'} \geq p_{m'}$, we have $y_{m'} = 0$, and

$$\frac{\partial \mathcal{L}_{BE}}{\partial z'_{m'}} = \frac{1}{|\mathcal{B}|} (p_{m'} - y_{m'}) = \frac{1}{|\mathcal{B}|} p_{m'} \leq \frac{1}{|\mathcal{B}|} p'_{m'} = \frac{1}{|\mathcal{B}|} (p'_{m'} - y_{m'}) = \frac{\partial \mathcal{L}_{IE}}{\partial z'_{m'}}.$$

Obviously, $p_{m'} \geq 0$, and therefore,

$$0 \leq \frac{\partial \mathcal{L}_{BE}}{\partial z'_{m'}} \leq \frac{\partial \mathcal{L}_{IE}}{\partial z'_{m'}}.$$

As the probabilities $p'_{m'}$ is computed from the logits through softmax operations, we have $z'_i = \log p'_i + \mathrm{Const.}$, and therefore we have

$$0 \leq \frac{\partial \mathcal{L}_{BE}}{\partial \log p'_{m'}} \leq \frac{\partial \mathcal{L}_{IE}}{\partial \log p'_{m'}}.$$

□

B Proof of Theorem 3.2

Proof. Like Appendix A, for a fixed bundle $\mathcal{B}$, we denote its (predicted) bundle-level class probability distribution as $\boldsymbol{p}(\theta) \triangleq \boldsymbol{p}_\theta(\mathcal{B}) = (p_1(\theta), p_2(\theta), \ldots, p_C(\theta))$, with corresponding logits as $\boldsymbol{z}$. More specifically, we can rewrite the $\boldsymbol{z}$ and $\boldsymbol{p}$ as:

$$\boldsymbol{z}(\theta) \triangleq (z_1(\theta), z_2(\theta), \ldots, z_C(\theta)) \triangleq \left(\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,1}(\theta), \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,2}(\theta), \ldots \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} z_{i,C}(\theta) \right);$$

$$p_i(\theta) \triangleq \frac{\exp(z_i(\theta))}{\sum_{j=1}^{C} \exp(z_i(\theta))} \quad \forall i \in \{1, 2, \ldots, C\}.$$

Moreover, we also denote the one-hot form of bundle label $\hat{y}$ predicted by the LLM as $\boldsymbol{y} = (y_1, y_2, \ldots, y_C)$. With this $\boldsymbol{z}(\theta)$ and $\boldsymbol{y}$ as well as $\boldsymbol{p}(\theta)$, we then can rewrite the cross-entropy loss function $\mathcal{L}_{BE}(\theta)$ as:

$$\mathcal{L}_{BE}(\theta) = -\sum_{i=1}^{C} y_i \log p_i(\theta) = -\sum_{i=1}^{C} y_i \log \left(\frac{\exp(z_i(\theta))}{\sum_{j=1}^{C} \exp(z_i(\theta))} \right) = -\sum_{i=1}^{C} y_i z_i(\theta) + \log \sum_{j=1}^{C} \exp(z_i(\theta)).$$

after that, according to the chain rule of differentiation, we also can show

$$\nabla L_{BE}(\theta) \triangleq \sum_{i=1}^{C} \frac{\partial L_{BE}(\theta)}{\partial z_i(\theta)} \nabla z_i(\theta) = \sum_{i=1}^{C} \frac{1}{|\mathcal{B}|} (p_i(\theta) - y_i) \nabla z_i(\theta), \tag{11}$$

where the final equality follows from Eq.(10) in Appendix A. As a result, we have

$$\begin{aligned}
\|\nabla L_{BE}(\theta)\|_\infty &\leq \sum_{i=1}^{C} \frac{1}{|\mathcal{B}|} |p_i(\theta) - y_i| * \|\nabla z_i(\theta)\|_\infty \\
&= \sum_{i=1}^{C} \frac{1}{|\mathcal{B}|} |p_i(\theta) - y_i| * \| \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \nabla z_{j,i}(\theta) \|_\infty \\
&\leq \sum_{i=1}^{C} \frac{1}{|\mathcal{B}|} |p_i(\theta) - y_i| * \left(\frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \|\nabla z_{j,i}(\theta)\|_\infty \right) \\
&\leq \left(\sum_{i=1}^{C} \frac{1}{|\mathcal{B}|} (p_i(\theta) + y_i) \right) G = \frac{2G}{|\mathcal{B}|},
\end{aligned}$$

where the first equality comes from $\nabla z_i(\theta) \triangleq \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \nabla z_{j,i}(\theta)$, the final inequality comes from $|p_i(\theta) - y_i| \leq |p_i| + |y_i| = p_i(\theta) + y_i$ and the final equality from $\sum_{i=1}^{C} p_i(\theta) \triangleq 1$ and $\sum_{i=1}^{C} y_i \triangleq 1$. Therefore, we verify the part **i)** in Theorem 3.2.

Next, we prove the part **ii)** in Theorem 3.2. At first, we show the second-order partial derivatives of the cross-entropy loss function $L_{BE}(\theta)$ with respect to θ is also bounded. Similarly, from the result of Eq.(11) and the chain rule of differentiation, we also can show that

$$\nabla^2 L_{BE}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{C} \left((p_i(\theta) - y_i) \nabla^2 z_i(\theta) + \nabla z_i(\theta) [\nabla p_i(\theta)]^T \right), \tag{12}$$

where the symbol $[\nabla p_i(\theta)]^T$ denotes the transport of the column gradient $\nabla p_i(\theta)$.

Subsequently, from the definition of $p_i(\theta)$ and the chain rule, we can show that

$$\nabla p_i(\theta) = \sum_{j=1}^{C} \frac{\partial p_i(\theta)}{\partial z_j(\theta)} \nabla z_j(\theta) = p_i(\theta) \left(\nabla z_i(\theta) - \sum_{j=1}^{C} p_j(\theta) \nabla z_j(\theta) \right), \tag{13}$$

where the final equality follows from the results that $\frac{\partial p_i(\theta)}{\partial z_i(\theta)} \triangleq p_i(\theta)\left(1 - p_i(\theta)\right)$ and $\frac{\partial p_i(\theta)}{\partial z_j(\theta)} \triangleq -p_i(\theta) p_j(\theta)$ when $j \neq i$.

Merging Eq.(13) into Eq.(12), we have that

$$\nabla^2 L_{BE}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C}(p_i(\theta)-y_i)\nabla^2 z_i(\theta) + \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C} p_i(\theta)\Big(\nabla z_i(\theta)[\nabla z_i(\theta)]^T\Big) - \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C}\sum_{j=1}^{C} p_i(\theta)p_j(\theta)\nabla z_i(\theta)[\nabla z_j(\theta)]^T.$$

As a result, we have

$$\begin{aligned}
&\max(|\nabla^2 L_{BE}(\theta)|)\\
&\leq \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C}\Bigg(|p_i(\theta)-y_i|\max(|\nabla^2 z_i(\theta)|) + p_i(\theta)\Bigg(\max\left(\left|\nabla z_i(\theta)[\nabla z_i(\theta)]^T\right|\right)\\
&+\sum_{j=1}^{C} p_j(\theta)\max\left(\left|\nabla z_i(\theta)[\nabla z_j(\theta)]^T\right|\right)\Bigg)\Bigg)\\
&\leq \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C}\left(|p_i(\theta)-y_i| * M + p_i(\theta)\left(G^2 + \sum_{j=1}^{C} p_j(\theta)G^2\right)\right)\\
&\leq \frac{1}{|\mathcal{B}|}\sum_{i=1}^{C}\left((p_i(\theta)+y_i)M + 2G^2 p_i(\theta)\right) = \frac{2(M+G^2)}{|\mathcal{B}|},
\end{aligned}$$

where the symbol $\max(|M|)$ represents the maximum absolute value among the elements of matrix M and the second inequality follows from $\max(|\nabla^2 z_i(\theta)|) \leq M$ and $\max\left(\left|\nabla z_i(\theta)[\nabla z_j(\theta)]^T\right|\right) \leq G^2$ for any $i, j \in \{1, 2, \ldots, C\}$.

From the previously established boundedness of the second-order partial derivatives of $L_{BE}(\theta)$, we next show the smoothness of the cross-entropy loss function $L_{BE}(\theta)$.

Firstly, if we suppose the dimension of the unknown parameter θ is n_d, we have that

$$\begin{aligned}
&\|\nabla\mathcal{L}_{BE}(\theta_1) - \nabla\mathcal{L}_{BE}(\theta_2)\|_2\\
&= \|\int_{\lambda=0}^{1}\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))\,(\theta_1-\theta_2)\,\mathrm{d}\lambda\|_2\\
&\leq \sqrt{n_d}\|\int_{\lambda=0}^{1}\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))\,(\theta_1-\theta_2)\,\mathrm{d}\lambda\|_\infty\\
&\leq \sqrt{n_d}\left\|\int_{\lambda=0}^{1}\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))\mathrm{d}\lambda\right\|_{2,\infty}\|\theta_1-\theta_2\|_2
\end{aligned}$$

where the first equality comes from the fundamental theorem of calculus, the first inequality from $\|\boldsymbol{x}\|_2 \leq \sqrt{m}\|\boldsymbol{x}\|_\infty$ where m is the dimension of $\boldsymbol{x}$ and the final inequality comes from the definition of $(2,\infty)$-norm [22], i.e., for any matrix $A \in \mathbb{R}^{n\times n}$, the $(2,\infty)$-norm of matrix A is defined as $\|A\|_{2,\infty} \triangleq \sup\{\|A\boldsymbol{x}\|_\infty : \boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_2 = 1\}$.

From the definition of the norm $\|\cdot\|_{2,\infty}$, we can show that

$$\begin{aligned}
&\left\|\int_{\lambda=0}^{1}\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))\mathrm{d}\lambda\right\|_{2,\infty}^2\\
&\leq \int_{\lambda=0}^{1}\left\|\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))\right\|_{2,\infty}^2\mathrm{d}\lambda\\
&= \int_{\lambda=0}^{1}\max_{j\in\{1,\ldots,n\}}\left\|\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))[j:]\right\|_2^2\mathrm{d}\lambda \leq \frac{4n_d(M+G^2)^2}{|\mathcal{B}|^2},
\end{aligned}$$

where $\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))[j:]$ is the j-th line of the Hessian matrix $\nabla^2\mathcal{L}_{BE}(\lambda\theta_1 + (1-\lambda))$ and the final inequality follows from the boundedness of the second-order partial derivatives of $L_{BE}(\theta)$.

As a result, we have

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

□

C Proof of Theorem 3.3

Proof. Note that when the model parameter θ_t can effectively fit the predicted bundle label $\hat{y}^B$, namely, $\hat{y}^B \in \arg\max_{i\in\{1,\dots,C\}}\{\boldsymbol{p}_{\theta_t}(\mathcal{B})_i\}$, we have $\mathcal{L}_R(\theta_t) = 0$ such that $\mathcal{L}(\theta_t) = \mathcal{L}_{BE}(\theta_t) + \mathcal{L}_R(\theta_t) = \mathcal{L}_{BE}$ and $\nabla\mathcal{L}(\theta_t) = \nabla\mathcal{L}_{BE}(\theta_t)$ when the iteration index t is large. From the results of Theorem 3.2, we know that, when the corresponding first-order and second-order partial derivatives of our adopted GNN g_θ are bounded, that is, $\|\nabla z_{i,c}(\theta)\|_\infty \leq G$ and $\max(\left|\nabla^2 z_{i,c}(\theta)\right|) \leq M$ where $z_{i,c}$ is the 'c'-th logit of the output vector $\boldsymbol{z}_i \triangleq (z_{i,1},\dots,z_{i,C})$ provided by g_θ, the cross-entropy loss function $\mathcal{L}_{BE}(\theta)$ is $\left(\frac{2n_d(M+G^2)}{|\mathcal{B}|}\right)$-smooth. Therefore, we have that

$$\begin{aligned}
\mathcal{L}_{BE}(\theta_{t+1}) &\leq \mathcal{L}_{BE}(\theta_t) + \langle\nabla\mathcal{L}_{BE}(\theta_t), \theta_{t+1} - \theta_t\rangle + \frac{n_d(M+G^2)}{|\mathcal{B}|}\|\theta_{t+1} - \theta_t\|_2^2 \\
&= \mathcal{L}_{BE}(\theta_t) - \eta\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2 + \frac{n_d(M+G^2)}{|\mathcal{B}|}\|\theta_{t+1} - \theta_t\|_2^2 \\
&= \mathcal{L}_{BE}(\theta_t) - \eta\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2 + \frac{n_d(M+G^2)}{|\mathcal{B}|}\eta^2\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2 \\
&= \mathcal{L}_{BE}(\theta_t) - \left(\eta - \frac{n_d(M+G^2)}{|\mathcal{B}|}\eta^2\right)\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2,
\end{aligned} \tag{14}$$

where the first inequality follows from the $\left(\frac{2n_d(M+G^2)}{|\mathcal{B}|}\right)$-smoothness of the adopted cross-entropy loss function $\mathcal{L}_{BE}(\theta)$ [60] and the three remaining equalities comes from $\theta_{t+1} \triangleq \theta_t - \eta\nabla\mathcal{L}_R$.

Finally, from Eq.(14), we have that

$$\left(\eta - \frac{n_d(M+G^2)}{|\mathcal{B}|}\eta^2\right)\sum_{t=1}^{T}\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2 \leq \sum_{t=1}^{T}(\mathcal{L}_{BE}(\theta_t) - \mathcal{L}_{BE}(\theta_{t+1})) = \mathcal{L}_{BE}(\theta_1) - \mathcal{L}_{BE}(\theta_{T+1}).$$

As a result, if $\eta < \frac{|\mathcal{B}|}{n_d(M+G^2)}$, we have that,

$$\frac{\sum_{t=1}^{T}\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2}{T} \leq \frac{\mathcal{L}_{BE}(\theta_1) - \mathcal{L}_{BE}(\theta_{T+1})}{\left(\eta - \frac{n_d(M+G^2)}{|\mathcal{B}|}\eta^2\right)T}.$$

In other words, $\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2}{T} = 0$. Furthermore, from the classic theory of calculus, we know that, for a positive sequence $\{a_1,\dots,a_n,\dots\}$, if $\lim_{n\to\infty}\frac{\sum_{i=1}^{n}a_i}{n} = 0$, then $a_n \to 0$ when $n \to \infty$. From this foundational result of calculus and the previous established limitation $\lim_{T\to\infty}\frac{\sum_{t=1}^{T}\|\nabla\mathcal{L}_{BE}(\theta_t)\|_2^2}{T} = 0$, we infer that $\lim_{T\to\infty}\|\nabla\mathcal{L}_{BE}(\theta_{T+1})\|_2 \to 0$ such that $\|\nabla\mathcal{L}(\theta_{T+1})\|_2 = \|\nabla\mathcal{L}_{BE}(\theta_{T+1})\|_2 \to 0$, when $T \to \infty$. □

D Overall Algorithm

We present the overall algorithm in Algorithm 1.

Algorithm 1 The overall algorithm of our method.

Require: A text-attributed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E}, \mathcal{T} \rangle$, bundle size n_B, and the number of bundles n_S. The set of epochs $\mathcal{R}$ where bundle refinement is performed. The total number of epochs for training T.

Ensure: The predicted category of each node in $\mathcal{V}$.

for i in $1, 2, ..., n_S$ **do**
  Sample a core node as v_c;
  if graph $\mathcal{G}$ is a homophilic graph **then**
    Sample $(n_B - 1)$ nodes from $\mathcal{N}_{\mathcal{G}}^{k}(v_c)$ according to Eq. 1;
    Construct the bundle using the core node v_c and the $(n_S - 1)$ nodes sampled in the previous step;
  else
    Sample n_B nodes according to Eq. 2 to form the bundle;
  end if
  Construct the prompt for each bundle using Eq. 3;
  Query the LLM to obtain the label for the bundle;
end for
for i in $1, 2, ..., T$ **do**
  Calculate the entropy-based supervision using Eq. 5;
  Calculate the ranking-based supervision using Eq. 7;
  Calculate the final loss function using Eq. 8;
  Update the parameters in the graph neural network g_θ using gradient descent;
  if $i \in \mathcal{R}$ **then**
    Update all the bundles using Eq. 9;
  end if
end for
Predict the categories of each node in $\mathcal{V}$ using the graph neural network g_θ.
return The predicted category of each node in $\mathcal{V}$.

Table 4: The statistics of the datasets.

Datasets	Cora	CiteSeer	WikiCS	History	Children	Sportsfit	Cornell	Texas	Wisc.	Wash.
Number of Nodes	2708	3186	11701	41551	76875	173055	191	187	265	229
Number of Edges	10556	8450	431726	503180	2325044	3020134	292	310	510	394
Number of Classes	7	6	10	12	24	13	5	5	5	5
Homophily Ratio	0.809	0.764	0.678	0.662	0.464	0.900	0.115	0.067	0.152	0.149

E Details about the Datasets

We provide more details about the datasets as follows:

- **Cora.** The Cora dataset is introduced by McCallum et al. [54], consisting of computer science research papers as nodes and their citation links as edges. Each paper is represented by its title and abstract, and the nodes are labeled into 7 classes corresponding to different research topics.
- **CiteSeer.** The CiteSeer dataset originates from the CiteSeer system described by Giles et al. [17]. Similar to Cora, this dataset is also a citation network, with nodes representing papers and edges denoting citation relationships.
- **WikiCS.** The WikiCS dataset is a graph of computer science Wikipedia articles proposed by Mernyei and Fazekas [58]. Each node is represented by the text of Wikipedia articles, classified into 10 subfields of computer science.
- **History, Children, and Sportsfit.** These three datasets were constructed by Ni et al. [62] from Amazon co-purchase data. Nodes represent products, edges represent frequent co-purchase relationships, and text attributes contain titles or reviews. Each dataset uses a three-level product

taxonomy for node labels. The History dataset contains history books as nodes with their co-purchase edges across 12 categories. The Children dataset contains children's books as nodes connected by their co-purchase relationships as edges across 24 categories. The Sportsfit dataset contains sports and fitness products as nodes with co-purchase edges across 13 categories.

- **Cornell, Texas, Wisconsin, and Washington.** These datasets are collected by Craven et al. [11], consisting of web pages from four universities. Nodes are web pages, edges are hyperlinks between pages, and the node attributes are the corresponding page content. Each page is labeled as one of seven types (*e.g.*, student, faculty, department).

We also present the statistics of these datasets in Table 4.

F Details about the Baseline Methods

We provide more details about the compared baseline methods as follows:

- **SBERT** [68]. Sentence-BERT (SBERT) modifies pretrained BERT with Siamese and triplet networks to obtain semantically meaningful sentence representations that can be compared with similarity metrics. We use this as a text embedding method.
- **RoBERTa** [49]. RoBERTa is an optimization of BERT removing the next-sentence prediction objective, trains longer on more data with larger batches, and uses dynamic masking. It achieves better results than BERT by carefully tuning hyperparameters and training recipes.
- **OpenAI's Text-Embedding-3-Large** [63]. OpenAI's text-embedding-3-large is a text embedding model that generates vectorized representations of texts [2].
- **LLM2Vec** [3]. LLM2Vec is an unsupervised solution converting decoder-only LLMs into powerful text encoders with bidirectional attention, masked next-token prediction, and contrastive learning.
- **GPT-3.5-turbo** [1]. GPT-3.5-turbo is OpenAI's chat-optimized GPT-3.5 model. It offers a cost-effective chat model in the GPT-3.5 series at a relatively low cost [3].
- **GPT-4o**[26]. GPT-4o is an auto-regressive multi-modal model that is cheaper than GPT-4 [4]. We use this model for a fair comparison with Wang et al. [77].
- **DGI** [73]. DGI is an unsupervised approach for learning node embeddings by maximizing mutual information (MI) between local patch embeddings and a global summary of the graph with GCN [33]. We use this as a graph unsupervised learning baseline.
- **GraphMAE** [23]. GraphMAE is a masked auto-encoder that reconstructs masked node features using a masking strategy and scaled cosine error loss. We use this as a graph unsupervised learning baseline.
- **OFA** [46]. OFA is a graph foundation model that describes the nodes and edges with natural language, which is then processed by large language models to obtain graph embeddings.
- **GOFA** [34]. GOFA integrates GNN layers into a frozen pre-trained LLM to combine semantic and topological modeling abilities. The model is then pretrained on various graph-level tasks.
- **UniGLM** [15]. UniGLM trains a unified graph language model across various text-attributed graphs (TAGs) using a self-supervised contrastive learning objective with positive sampling and a lazy contrastive module.
- **ZeroG** [40]. ZeroG is a graph foundation model that encodes node attributes and class semantics via prompts and a prompt-based subgraph sampling module.
- **GraphGPT** [72]. GraphGPT is another graph foundation model that applies graph instruction tuning to LLMs by grounding graph structures in text and uses dual-stage instruction tuning with graph-text alignment.
- **LLAGA** [6]. LLaGA is a graph foundation model that translates graph inputs into token embeddings through structure-aware translation and alignment tuning, preserving graph information in tokens without modifying the base model.

[2] https://platform.openai.com/docs/models/text-embedding-3-large

[3] https://platform.openai.com/docs/models/gpt-3.5-turbo

[4] https://platform.openai.com/docs/models/gpt-4o

- **LLM-BP** [77]. LLM-BP is a zero-shot inference method on text-attributed graphs. It proposes task-adaptive embeddings and adopts belief propagation with LLM-estimated parameters.

Table 5: The training time on various datasets.

Datasets	Cora	CiteSeer	WikiCS	History	Children	Sportsfit	Cornell	Texas	Wisc.	Wash.
Training Time (s)	11	14	47	59	155	258	20	16	24	20

G Additional Implementation Details

We perform experiments using various datasets. For dataset split, we follow previous works [9, 77] and their official implementations [5] [6]. For the raw text data, we also use the data sources from existing works [11, 83, 9, 77] and the version from a Hugging Face repository [7]. For the text encoder f_θ, we use the task-adaptive embedding proposed in [77]. For the GNN classifier g_θ, we use GCN [33] for Cora, CiteSeer, WikiCS, History, Children, and Sportsfit. The GCN is implemented with two layers of convolutions with jumping knowledge [82] by default. We use GloGNN [39] for the Cornell, Texas, Wisconsin, and Washington datasets. The hyperparameters of GloGNN follow its official implementation [8]. For optimization, we use the Adam optimizer [32] with a learning rate of 0.001, and optimize for 500 epochs by default. For bundles, the bundle size is set to 5, and the number of bundles is set to 100. For hardware, we use an NVIDIA RTX 3090 GPU with 24GB of memory. As the time for querying GPTs from online sources depends on network conditions, we only measure the time for training GNNs in Table 5.

Table 6: The prompt parameters of different datasets.

Datasets	<node_description>	<graph_description>	<node_refer_name_ploral>	<node_refer_name_singular>
Cora	opening text of machine learning papers	citation network	papers	Paper
Citeseer	description or opening text of scientific publications	citation network	papers	Paper
WikiCS	entry and content of wikipedia	knowledge graph	entries	Entry
History	description or title of the book	e-commerce network	books	Book
Children	description or title of the child literature	e-commerce network	books	Book
Sportsfit	the title of a good in sports & fitness	e-commerce network	products	Product
Cornell	webpage text	university webpage network	webpages	Webpage
Texas	webpage text	university webpage network	webpages	Webpage
Wisconsin	webpage text	university webpage network	webpages	Webpage
Washingt	webpage text	university webpage network	webpages	Webpage

H Prompt Details

We also provide additional details about the prompt. Specifically, given the prompt template in Figure 4, we fill the prompt parameters using the texts in Table 6 together with the number of classes in Table 4 and specific classes attached to each dataset to form the final prompts.

[5] https://github.com/CurryTang/TSGFM
[6] https://github.com/Graph-COM/LLM_BP
[7] https://huggingface.co/datasets/Graph-COM/Text-Attributed-Graphs
[8] https://github.com/RecklessRonan/GloGNN