TGL_{sta}: Low-resource Textual Graph Learning with Semantic and Topological Awareness via LLMs

Qin Zhang[1], Xiaowei Li[1*], Ziqi Liu[1*], Xiaochen Fan[2], Xiaojun Chen[1], Shirui Pan[3 †]

[1]College of Computer Science and Software Engineering, Shenzhen University, China
[2] Institute for Electronics and Information Technology in Tianjin, Tsinghua University, China
[3]School of Information and Communication Technology, Griffith University, Australia.
{qinzhang, xjchen}@szu.edu.cn, {lixiaowei2022, liuziqi2022}@email.szu.edu.cn,
fanxiaochen33@gmail.com, s.pan@griffith.edu.au

Abstract

Textual Graphs (TGs) present a graph-based representation of textual data and find wide applications in real-world scenarios, such as citation networks, knowledge graphs, and social networks. While the traditional "pre-train, fine-tune" framework effectively addresses tasks requiring abundant labeled data, it falls short in scenarios with limited resource or zero-shot learning capabilities, particularly in low-resource textual graph node classification. Additionally, prevalent approaches that convert text nodes into shallow or manually engineered features fail to capture the rich semantic nuances within the text. The conventional methods often neglect the fusion of semantic and topological information, resulting in suboptimal model learning. To overcome these challenges, we proposed a novel method of low-resource textual graph node classification based on large language models, i.e., *Textual graph learning with semantic and topological awareness (TGL_{sta})*, which comprehensively explores the semantic information, near neighborhood information, and the topology information in textual graphs, where these components are the most important information source contained in textual graphs. Graph prompt tuning for both zero- and few-shot textual graph node classification is further introduced.

Introduction

Owing to the remarkable success in representation learning on graph-structured data, the exploration of learning methodologies applied to textual graphs (TGs) has emerged as a prominent research domain across multiple fields, including graph learning(Zhang et al. 2024c), information retrieval(Reinanda et al. 2020), and natural language processing(Wu et al. 2023). TGs (Zhang et al. 2024b; Ni, Li, and McAuley 2019) are widely utilized in real-world applications, including citation graphs (Zhang et al. 2023, 2024c), knowledge graphs (Wang et al. 2021a), and social networks (Zhang et al. 2024d; Ni, Li, and McAuley 2019; Zeng et al. 2019). They can provide a graph-based representation of text data, illustrating the relationships between phrases, sentences, or documents through edges.

For textual graph learning, where the data is constructed by phrases, sentences or documents and their relations, the goal is to learn both the structure modality (topological information) and the textual description (semantic information). Recent advances in textual graph learning (Hamilton, Ying, and Leskovec 2017; Zhang et al. 2024a; Fang et al. 2024) have achieved remarkable success with adequate high-quality labels. The common routine follows a "pre-train, fine-tune" framework, which utilizes large amounts of unlabeled data and self-supervised training strategies to obtain a pre-trained GNN. The fine-tuning phase is performed in downstream tasks with a certain amount of labeled data. Natural language is inherently characterized by open vocabulary and free-form expression (He et al. 2023). This fundamental characteristic poses significant challenges for textual graph labeling, making it both resource-intensive and time-consuming. Consequently, low-resource classification approaches, which operate with minimal or no labeled samples, have emerged as promising alternatives. Besides, the pre-trained model tends to overfit the few-shot labeled data in many low-resource scenarios. This has led to a proliferation of studies in the field of *low-resource textual graph learning*, which aims to learn fast-adaptable GNNs for unseen tasks with extremely scarce labeled samples.

To effectively solve the low-resource textual graph learning problem, previous approaches generally fall into two categories (Ju et al. 2023; Liang et al. 2024; Liu et al. 2024a). Methods in the first category (Wang et al. 2021b; Lu et al. 2022; Ju et al. 2023; Liang et al. 2024) primarily employ meta-learning strategies, and focus on extracting transferable knowledge from source classes with abundant labeled data by subsequently adapting this knowledge to target classes. They usually assume that the nodes in the meta-training classes are gold-labeled, which is insatiable in many real-world applications. Besides, most existing methods (Chen et al. 2019; Huang and Zitnik 2020; Sun, Zhou, and Zong 2021; Zhang et al. 2020; Wang et al. 2022) focus on task generalization within a single graph, or rely on simplified text representations (You et al. 2020; Hou et al. 2023; Liu et al. 2024b), such as skip-gram or bag-of-words, which fail to capture complex semantics, especially sophisticated relationships and contextual dependencies.

Methods in the other category primarily strive to leverage

*Contributed equally
†Corresponding author

LLMs learning effective text semantic information, which is attributed to their extensive knowledge memorized during the pretraining phase and their generalization on diverse textual datasets (Liu et al. 2024a; Chen et al. 2024; Qiu et al. 2024; Liang et al. 2022; Tan et al. 2024; Tian et al. 2024). This success has spurred interest in combining graph neural networks (GNNs) with LLMs to enhance their capabilities in understanding and modeling graphs, including implementing LLMs as encoders to process features within GNNs, and employing LLMs as aligners with GNNs to enhance performance. The direct application of LLMs to graph-structured data presents significant challenges due to the inherent modality gap between graph representations and textual data (Liu et al. 2024a). For instance, applying a subnetwork to each modality and then fusing them would fail (Sahu and Vechtomova 2021) due to the heterogeneous data distributions across different modalities, which result in learned representations following complex, unknown distributions (Mai, Hu, and Xing 2020). This issue becomes even more challenging in low-resource scenarios. While several graph pre-training methods (Hu et al. 2019, 2020; You et al. 2020; Qiu et al. 2020) use LLMs, they mainly focus on creating graph encoders that need task-specific fine-tuning. In contrast, low-resource graph learning requires cross-graph and cross-task generalization without fine-tuning.

Given these limitations, we proposed a novel method of low-resource textual graph node classification based on large language models, i.e., *Textual graph learning with semantic and topological awareness (TGL$_{sta}$)*, which comprehensively explores the semantic information, near neighborhood information, and the topology information in textual graphs, since these three components are the most important information source contained in textual graphs. The goal of low-resource Textual Graph Node Classification is to learn fast-adaptable models for unseen tasks with extremely scarce ground-truth labels. The graph-grounded pre-training framework creates a dual-modal embedding space by simultaneously training a text encoder, topology encoder, and graph encoder. This process is driven by three interaction types within the graph structure. Based on this, we propose a graph prompt tuning mechanism for zero-shot and few-shot classification of textual graph nodes. To conclude, our contributions are summarized as follows:

- We develop a novel multi-view (node-summary-topology) graph-grounded self-supervised learning model, namely *TGL$_{sta}$*, which enables to capture both semantic and structural information on textual graphs.
- We develop a multi-view contrastive loss to train a transferable pre-training model on few-shot and zero-shot tasks.
- We evaluate our method on various benchmarks, well showing that our method consistently improves the current baselines.

Preliminaries

This study focuses on low-resource node classification for textual graphs. Consider a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$, which are grounded on a graph $\mathcal{G} = (\mathcal{D}, E, X)$. Here, $\mathcal{D}$ is the set of M documents in the graph $\mathcal{G}$, $E = \{e_{i,j} | i, j = 1, \ldots, N, i \neq j\}$ is the set of edges connecting pairs of documents d_i and d_j, and X denotes the feature matrix of documents, where the feature vector of each document d_i is represented as $x_i \in X$. The topological structure of $\mathcal{G}$ is represented by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, where $Ai, j = 1$ if documents d_i and d_j are connected (i.e., $\exists e_{i,j} \in \mathcal{E}$), and $A_{i,j} = 0$ otherwise. The label matrix of $\mathcal{G}$ is $\mathbf{Y} \in \mathbb{R}^{M \times C}$, where C is the number of document classes. If a label c is assigned to a document $d_i \in \mathcal{D}$, then $y_{i,c} = 1$; otherwise, $y_{i,c} = 0$.

The problem of **low-resource node classification** is to learn fast-adaptable GNNs for unseen tasks with extremely scarce ground-truth labels. Conventionally, the tasks are denoted as $\mathcal{W}$-way $\mathcal{S}$-shot $\mathcal{Q}$-query node classification tasks, where $\mathcal{W}$ is the number of classes, $\mathcal{S}$ is the small number of labeled nodes per class (e.g., 1 or 5), and $\mathcal{Q}$ is the number of unlabeled nodes per class. The labeled nodes are referred to as the "support set" and the unlabeled nodes are referred to the "query set" for evaluation.

Approach

To effectively and efficiently leverage the rich but complex information in textual graphs, we propose a novel text-topology-graph aligned pre-training and prompt-tuning framework called *Textual Graph Learning with Semantic and Topological Awareness, i.e., TGL$_{sta}$*. It jointly explores and learns semantic, local neighborhood and global topology information contained in textual graphs during pre-training phase, where these three aspects are the main and the most essential contents the model needs to acquire. In the downstream tasks, we employed prompt-tuning to fine-tune the pre-trained model, enabling its adaptation to these low-resource scenarios.

Text-topology-graph Aligned Pre-training

The graph-grounded pre-training methodology jointly optimizes a text encoder, a topology encoder, and a graph encoder to learn a dual-modal embedding space. This is achieved by leveraging three types of interactions derived from the underlying graph structure.

Unimodal Encoders

Text Encoder Towards textual graphs, such as Cora (McCallum et al. 2000) which comprises scientific papers and their citation relationships, we use $D = \{d_i | i = 1, ..., M\}$ to encapsulate the textual attributes of their nodes, where M is the total number of nodes. To extract the semantic information of the original texts, we augment pre-trained large language models with a trainable text encoder to obtain high-quality representations of nodes' text and also to maintain the flexibility for alignment.

Specifically, we first leverage a pre-trained LLM f_{LLM}, such as *LLaMA2* (Touvron et al. 2023) or *GPT* (Achiam et al. 2023), to obtain the original embedding of each node in terms of text mode, i.e.,

$$z_i = f_{LLM}(d_i), i = 1, \ldots, M. \quad (1)$$

$\mathbf{Z} \in \mathbb{R}^{M \times r_1}$ represents the original embedding matrix for all node. Since pre-trained LLMs are normally trained on extensive text corpus, they would benefit to versatility in handling diverse tasks. Subsequently, we add a text encoder f_θ further to make the text-modal representations of nodes being adapted and aligned efficiently with other modes. Given the document d_i and its LLM-based embedding z_i, the text encoder f_θ outputs the r_2-dimensional embedding vector of d_i, denoted $t_i \in \mathbb{R}^{r_2}$:

$$t_i = f_\theta(z_i; \theta), i = 1, \ldots, M. \tag{2}$$

Here, θ denotes the set of parameters for the encoder. Correspondingly, $\mathbf{T} \in \mathbb{R}^{M \times r_2}$ denotes the text embedding matrix for all nodes.

At the same time, since a document d_i corresponds to a node n_i in the graph, we further utilize text summary s_i for each node to improve the consistency of the local community. The text summary embedding s_i for the i-th node is constructed by applying mean pooling to the k-hop neighborhood of that node in the graph, i.e.,

$$s_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{t}_j, i = 1, \ldots, M. \tag{3}$$

where $s_i \in \mathbb{R}^{r_3}$ represents the text summary embedding for node i, $\mathcal{N}_i$ denotes the set of k-hop neighbors of node i. The text summary embedding matrix for all nodes is denoted as $\mathbf{S} \in \mathbb{R}^{M \times r_3}$.

The text summary of each node represents its k-hop neighborhood as a single corpus that includes all nodes and their connections. This corpus is processed to extract semantic embeddings, capturing both content and structure within the neighborhood, and providing context for the central node's relationships. During pre-training, contrastive learning between the node's text and its summary is used to enhance the consistency of the community in text mode.

Topology Encoder Most existing graph models focus on a node's local community by passing messages within its k-hop neighborhood. However, capturing universal topological properties across multiple networks is also crucial. To learn transferable structural representations that apply both to individual graphs and the collective graph space, we generate a global topological subgraph for each node using a $\mathcal{K}$-hop random walk, and use structure-related features to learn structural similarity and transferability.

Specifically, we first employ a $\mathcal{K}$-hop random walk with restart to generate global topological subgraphs for each node. Given a graph $\mathcal{G}$, we define the augmented subgraph from node i as $\hat{\mathcal{G}}_i$, it assumes a random surfer that begins at node i and, at each step, chooses one of the following actions: **1) Random walk** The walker iteratively moves to a neighboring node chosen randomly with probability $1 - \alpha$. **2) Restart** During the walk, the restart probability α governs the likelihood that the walker returns to the initial node rather than continuing to a neighboring one. A higher value of α increases the frequency of resets to the starting node, influencing the exploration dynamics. We replicate this procedure twice, resulting in two distinct data augmentations that constitute a similar instance pair $\{\hat{\mathcal{G}}_i, \hat{\mathcal{G}}_i'\}$,

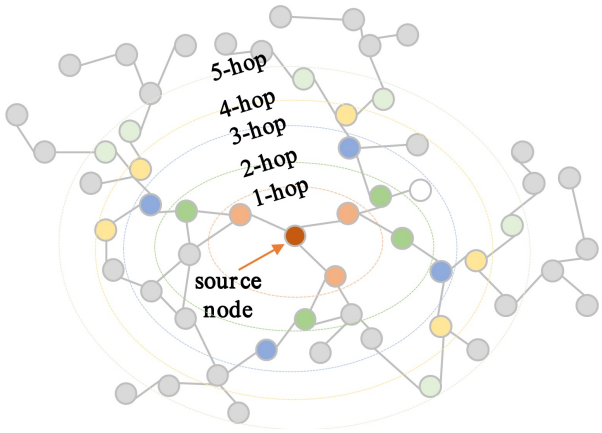


Figure 1: An example for k-hop subgraphs ($k = 1, 2, 3, 4, 5$).

while instances originating from distinct subgraphs of different nodes $\{i, j\}$ are defined as dissimilar pairs $\{\hat{\mathcal{G}}_i, \hat{\mathcal{G}}_j\}$. Figure 1 illustrates example graphs with different k, demonstrating how the node structure view captures different levels of neighborhood information. To enhance our understanding of graph structures, we leverage topological information as node feature embeddings, which are subsequently fed into the graph structure encoder for learning. To be specific, given an augmented subgraph $\hat{\mathcal{G}}_i$ from node i, using the degree matrix $D_{\hat{\mathcal{G}}_i}$ and adjacency matrix $A_{\hat{\mathcal{G}}_i}$, we perform eigen-decomposition on the normalized graph Laplacian:

$$I - D_{\hat{\mathcal{G}}_i}^{-\frac{1}{2}} A_{\hat{\mathcal{G}}_i} D_{\hat{\mathcal{G}}_i}^{-\frac{1}{2}} = U_{\hat{\mathcal{G}}_i} \mathbf{\Lambda} U_{\hat{\mathcal{G}}_i}^T \tag{4}$$

where the top eigenvectors in $U_{\hat{\mathcal{G}}_i}$ are defined as the node embeddings. It allows us to learn pure graph structural information without relying on node text features. Then, the node structure views positive pairs $\{h_i, h_i'\}$ and negative pairs $\{h_i, h_j\}$ are analyzed and compared by a Graph Topology Encoder g_ϕ:

$$h = g_\phi(\hat{\mathcal{G}}; \phi) \tag{5}$$

It is worth noting that we use the larger $\mathcal{K}$ and the smaller k to learn the local and global graph structures. To clarify the comparison, we visulize this distinction by plotting Figure 1. When setting $k = 1$ and $\mathcal{K} = 5$, we observe that the k-hop subgraph closely resembles the immediate neighborhood of the source node. Conversely, the $\mathcal{K}$-hop subgraph provides a broader perspective, capturing global graph structure details around the source node. Subsequently, in our later experiments, we adjust k to 2 and $\mathcal{K}$ to 256.

Graph Encoder We leverage the textual graph topology and the node feature embeddings generated by a pre-trained LLM f_{LLM}, and then feed it into a Graph encoder f_ψ:

$$l = f_\psi(\mathcal{G}; \psi), \mathcal{G} = \{D, E, \mathbf{Z}\} \tag{6}$$

By incorporating these three views: node text, node summary, and node structure, our model leverages local textual information, broader semantic context, and pure structural

properties of the graph. This comprehensive approach enables a more nuanced understanding of complex networks, potentially improving performance across various graph-based machine learning tasks.

Multimodal Alignment

During the pre-training, we jointly train the text encoder, the topology encoder and the graph encoder in a contrastive learning manner, including text-summary alignment, node-topology alignment, and text-topology-graph cross-modal alignment.

Text-summary Alignment To harness the intricate semantic relationships within textual graphs, we leverage text-summary contrastive learning to achieve the semantic alignment. For text d_i of node i, encapsulated in its representation t_i, is paired with its k-hop text summary s_i. These pairs serve as positive samples during the contrastive learning based pretraining, fostering the model's ability to capture and preserve the nuanced meaning and context of node texts and their summarizations. Conversely, negative pairs are formed as the text of a node and the summary of a different node, to ensure the model learns to distinguish between semantically related and unrelated pairs. Thus we use the following text-summary contrastive loss to constraint the training of the model:

$$\begin{aligned}\mathcal{L}_{te-su} = -\log \frac{1}{2}\Big(&\frac{\exp(sim(t_i, s_i)/\tau)}{\sum_j \exp(sim(t_i, s_j)/\tau)} \\ &+ \frac{\exp(sim(t_i, s_i)/\tau)}{\sum_j \exp(sim(t_j, s_i)/\tau)}\Big)\end{aligned} \tag{7}$$

where $sim(t_i, s_j) = \frac{t_i^T s_j}{||t_i|| ||s_j||}$, τ is a temperature parameter. The text-summary contrastive learning can improve the semantic consistency in a relatively wider graph-guided scale, which is reasonable and also helps the model learn more about the inherent connections between nodes.

Node-topology Alignment Traditional GNNs learn structure information along with node attributes and limited to local neighborhood. To acquire robust and sheer graph topological information, we obtain topology embedding for each node (see Section) and use contrastive learning to align them. Our objective is to ensure that nodes with similar local structures possess analogous representations, enabling the model to attune to structural similarities in and across graphs. Specifically, by performing the random walk with restart twice starting from the same source node, we create a set of subgraph pairs, which we consider as positive pairs. Conversely, the set of subgraph pairs generated from different source nodes is treated as negative pairs. To enhance the model's ability to discern meaningful features, we aim to reduce the similarity between negative samples while increasing the similarity between positive samples, thereby encouraging the model to learn more discriminative structural representations:

$$\mathcal{L}_{no-to} = -\log \frac{\exp\left(sim(h_1, h_2)/\tau\right)}{\sum_{i,j} \exp\left(sim(h_i, h_j)/\tau\right)} \tag{8}$$

where (h_1, h_2) denotes the low-dimensional representations of similar subgraph pairs generated from the same centered node i, τ is a temperature parameter, $sim(\cdot)$ is cosine similarity with $sim(z_i, z_j) = \frac{z_i^T z_j}{||z_i|| ||z_j||}$. For similar subgraph pairs (h_1, h_2) should be close, so the numerator is large and the loss is small. For dissimilar pairs generated from different centered nodes, the denominator becomes large and the loss increases.

Text-topology-graph Cross-modal Alignment To achieve synchronization between graph-based and text-based embeddings, we align the representations of text and graph for the same or related nodes using two distinct contrastive strategies: a) node-structure view contrastive learning, and b) summary-structure view contrastive learning.

Node-structure view contrastive learning leverages embeddings from both node text and graph structure to enhance representation learning in textual graphs. For each node, the node semantic embedding l_i from graph encoder and structure embedding h_i from graph encoder are concatenated to form $p_i = l_i \odot h_i$. These concatenated representations p_i are then paired with the node text embedding t_i generated from our trained text encoder, establishing positive samples. Simultaneously, negative pairs are created by pairing p_i with node text embeddings t_j from other nodes. The formulation of the node-structure view contrastive loss is as follows:

$$\begin{aligned}\mathcal{L}_{no-st} = -\log \frac{1}{2}\Big(&\frac{\exp(sim(p_i, t_i)/\tau)}{\sum_j \exp(sim(p_i, t_j)/\tau)} \\ &+ \frac{\exp(sim(p_i, t_i)/\tau)}{\sum_j \exp(sim(p_j, t_i)/\tau)}\Big)\end{aligned} \tag{9}$$

where $sim(p_i, t_i) = \frac{p_i^T t_i}{||p_i|| ||t_i||}$.

For summary-structure view contrastive learning, each node structure view after concatenation p_i, paired with the embedding of node summary view s_i serve as positive samples. While negative pairs are constructed by pairing p_i with node summary embedding s_j from other nodes. The formulation of the summary-structure view contrastive loss is as follow:

$$\begin{aligned}\mathcal{L}_{su-st} = -\log \frac{1}{2}\Big(&\frac{\exp(sim(p_i, s_i)/\tau)}{\sum_j \exp(sim(p_i, s_j)/\tau)} \\ &+ \frac{\exp(sim(p_i, s_i)/\tau)}{\sum_j \exp(sim(p_j, s_i)/\tau)}\Big)\end{aligned} \tag{10}$$

where $sim(p_i, s_i) = \frac{p_i^T s_i}{||p_i|| ||s_i||}$.

Pre-training Objective

Finally, we combine the contrastive loss functions for text-summary, node-topology, and node-graph pairs to improve the model's alignment. We employ a pre-trained model $\psi^0 = (\theta^0, \phi^0)$ consisting of the parameters of the trainable graph encoder and text transformer, given by

$$\arg\min_{\theta,\phi} \frac{1}{M} \sum_{i=1}^{M} (\mathcal{L}_{te-su} + L_{no-to} + \lambda(\mathcal{L}_{no-st} + \mathcal{L}_{su-st})) \tag{11}$$

where $\lambda \in \mathcal{R}^+$ is a hyperparameter to balance the contribution from summary-based pairs. Through this multi-view

contrast learning approach, our model can effectively capture both structural and semantic information from graphs while aligning the graph and text representations in a joint embedding space.

Prompt Tuning for Low-resource Classification

After pre-training the textual graph model, we use prompt tuning to adapt it for low-resource classification tasks. The traditional "pre-train, fine-tune" approach, where a new projection head is added and fine-tuned with the entire model (Wen and Fang 2023), is often used to bridge the gap between pre-training and downstream applications. However, in low-resource scenarios, the lack of sufficient supervised samples and the computational inefficiency of fine-tuning a large pre-trained model makes this approach impractical. Thus, we employ prompt tuning (Liu et al. 2023) to adapt the pre-trained model for specific downstream tasks.

Drawing on the design of the prompt tuning in the NLP field (Fang et al. 2022), we introduce graph prompt tuning for both zero- and few-shot textual graph node classification. Given the pre-trained graph augmented LLM model f and the downstream textual node classification task, our target is to obtain a task-specific *prompt* p_φ parameterized by φ.

In **few-shot classification** scenarios, given downstream task dataset $D_{task} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where m is small, we optimize the parameters of φ to maximize the likelihood of correctly predicting the labels y, without adjusting the pre-trained model f(i.e., keeping the pre-trained model frozen). This can be expressed as:

$$\max_{\varphi} P_{f,\varphi}(y|\mathcal{G}) \tag{12}$$

we utilize a sequence of continuous embeddings $[\mathbf{h}_1, \cdots, \mathbf{h}_H, \mathbf{h}_{\text{CLASS}}]$ as the prompt, where H is a hyperparameter representing the number of context tokens, each $h_i (i \leq H)$ being a learnable vector, and h_{CLASS} corresponds to the embedding of the target class label. This continuous prompt is then passed into the text encoder to produce the classification weights for each class y:

$$\mathbf{w}_y = \varphi_T([\mathbf{h}_1, \cdots, \mathbf{h}_H, \mathbf{h}_{\text{CLASS}}]; \varphi_T^0) \tag{13}$$

Here, each $h_i (i \leq H)$ is of the same dimension as the input word embeddings provided to the text encoder. Subsequently, the class distribution based on the node representation $\mathbf{z}_i$ is predicted as

$$p(y \mid \mathbf{z}_i) = \frac{\exp\left(\langle \mathbf{z}_i, \mathbf{w}_y \rangle\right)}{\sum_{y=1}^{N} \exp\left(\langle \mathbf{z}_i, \mathbf{w}_y \rangle\right)} \tag{14}$$

where $\langle \cdot, \cdot \rangle$ is the cosine similarity.

In **zero-shot Classification** scenarios, in $\mathcal{W}$-way zero-shot classification, we select the class with the highest similarity to the given node from the set of $\mathcal{W}$ possible classes. The weights for classification can be derived from the text transformer using only the textual descriptions of the class labels, eliminating the need for any labeled samples. Specifically, the vector $\mathbf{w}_y$ representing the weight for class $y \in \{1, 2, \ldots, \mathcal{W}\}$ is produced by the pre-trained text transformer, as shown below:

$$\mathbf{w}_y = \varphi_T(\text{"prompt [CLASS]"}; \varphi_T^0) \tag{15}$$

where the "prompt [CLASS]" represents a template where "[CLASS]" corresponds to the label text of the target class y. The term "prompt" refers to a sequence of natural language tokens that are manually crafted to highlight the association with the class label. Next, we apply the same softmax function as shown in Eq. (14).

Experiments

Experimental Setup

Datasets and Evaluation Metrics We evaluated the performance of the *TGL*$_{sta}$ framework by conducting experiments on four publicly available graph-based text corpora: Cora (McCallum et al. 2000), Art, Industrial, and Music Instruments (M.I.) (Ni, Li, and McAuley 2019). In terms of the metrics, we adopt macro-F1 and Accuracy that widely used for classification problems.

Implementation Details For the experiments, we utilize GCN as the core neural network for the graph encoder, which consists of two hidden layers, each with 128 dimensions and LeakyReLU activation. We employ a transformer as the text encoder (Vaswani 2017). In line with CLIP (Radford et al. 2021), our setup features a 63-million-parameter model with 12 layers, each 512 units wide, and equipped with 8 attention heads. It utilizes a lower-cased byte pair encoding (BPE) scheme to represent texts, with a vocabulary size of 49,152 (Sennrich, Haddow, and Birch 2016). We cap the maximum sequence length at 128. The experiments were performed on a workstation with an Intel(R) Xeon(R) Gold 6226R CPU and an Nvidia A100 GPU.

Test Settings To evaluate the effectiveness of *TGL*$_{sta}$, we conduct zero-shot and few-shot node classification tasks. We specifically designed an experiment focused on few-shot text classification. The procedure begins with pre-training on the OGBN-Arxiv dataset, followed by prompt tuning on the Cora dataset. We then proceed with pre-training on the Industrial dataset and conclude with prompt tuning on the M.I. and Art datasets. In the few-shot classification setting, we use a 5-way approach, where each task involves selecting five classes from the full set. For each class, we sample $\mathcal{S}$ examples, where $\mathcal{S} \in \{0, 1, \ldots, 5\}$, to form the $\mathcal{S}$-shot support set used for model training. The corresponding validation set is of the same size as the support set, while the remaining examples are assigned to the query set, which is unlabeled and used for evaluation. Each task considers five classes, and multiple tasks are run to ensure a comprehensive evaluation across all class combinations.

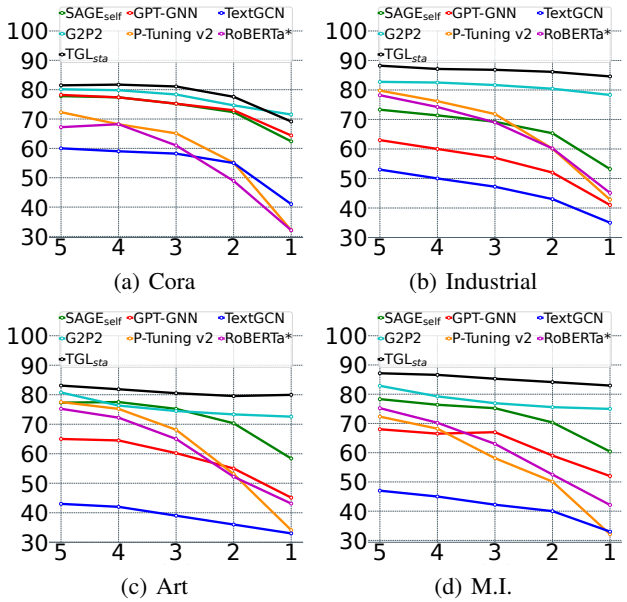
Experimental Results and Analysis

Five- and Fewer-shot Node Classification As presented in Table 1, we compare the performance of *TGL*$_{sta}$ with that of the baseline methods in the five-shot setting. *TGL*$_{sta}$ consistently outperforms the best baseline, achieving an improvement of approximately $2\% - 5\%$ in few-shot classification performance, with 95% confidence intervals and statistical significance. Beyond the 5-shot setting, Figure 2 explores the effect of fewer shots on *TGL*$_{sta}$ and several representative baselines. *TGL*$_{sta}$ consistently outperforms the baselines across different shot settings. As the number of

Models	Cora		Industrial		Art		M.I.	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
GCN	41.15	34.50	21.08	15.23	22.47	15.45	22.54	16.26
Sage_{sup}	41.42	35.14	20.74	15.31	22.60	16.01	22.14	16.69
TextGCN	59.78	55.85	53.60	45.97	43.47	32.20	46.26	38.75
GPT-GNN	76.72	72.23	62.13	54.47	65.15	52.79	46.26	38.75
DGI	78.42	74.58	52.29	45.26	65.41	53.57	68.06	60.64
SAGE_{self}	77.59	73.47	71.87	65.09	76.13	65.25	77.70	70.87
BERT	37.86	32.78	54.00	47.57	46.39	37.07	50.14	42.96
BERT*	27.22	23.34	49.60	43.36	45.31	36.28	40.19	33.69
RoBERTa	62.10	57.21	76.35	70.49	72.95	62.25	70.67	63.50
RoBERTa*	67.42	62.72	77.08	71.44	74.47	63.35	74.61	67.78
P-Tuning v2	71.00	66.76	79.65	74.33	76.86	66.89	72.08	65.44
G2p2	80.14	75.70	82.78	76.54	81.08	69.74	82.71	75.82
TGL_{sta}	**82.26**	**77.41**	**88.23**	**83.31**	**83.10**	**71.75**	**87.16**	**81.60**

Table 1: Accuracy and Macro-F1 scores for five-shot classification (%)

shots decreases, the performance of all methods generally declines. However, while the baselines suffer a substantial drop under extreme fewer-shot conditions, TGL_{sta} remains resilient, with only a modest decrease in performance even with just 1 or 2 shots.



(a) Cora (b) Industrial

(c) Art (d) M.I.

Figure 2: Fewer-shot classification accuracy (percent).

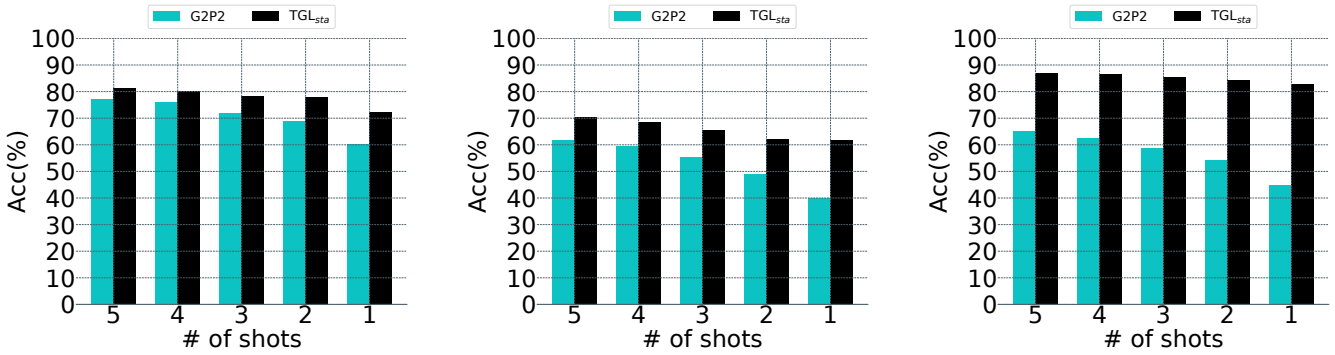
Few-shot Classification Based on Different Training Datasets To rigorously assess the model's robustness, we devised a more challenging evaluation protocol. Initially, we subjected the model to pre-training across various datasets, followed by fine-tuning through prompt-tuning and few-shot classification tasks using additional datasets. In Figure 3, we illustrate this process: leveraging the Arxiv dataset for pre-training, we conduct prompt-tuning and few-shot node classification tasks on the Cora dataset. Subsequently, we employ the Industrial dataset for pre-training and assess the model's performance on the Art and M.I. datasets. This experimental setup was chosen deliberately; while Arxiv and Cora datasets share the citation network domain, the Industrial, Art, and M.I. datasets, utilized for both pre-training and prompt-tuning, stem from the same domain but encompass distinct categories.

In Figure 3, a notable enhancement is observed in the performance of our method compared to previous approaches. This improvement can be attributed to the limitations of the SOTA model, which fails to effectively integrate both semantic and topological information inherent in graphs. In contrast, our approach excels by adeptly leveraging both semantic and topological features, thus ensuring robustness through alignment in the feature space. This strategic combination enables our model to achieve superior performance.

Zero-shot classification The zero-shot performance is presented in Table 2, where our models clearly outperform the baselines. These results emphasize the effectiveness of our multi-hop contrastive pre-training, which is crucial for handling evolving classes without labeled data in real-world applications.

	Cora	**Industrial**	**Art**	**M.I.**
RoBERTa	30.46	42.89	42.80	36.40
RoBERTa*	39.58	37.78	34.77	32.17
RoBERTa*+d	45.53	39.40	36.11	37.65
BERT	23.58	37.32	35.88	37.42
BERT*	23.38	56.02	54.27	50.19
BERT*+d	26.65	55.93	56.61	52.13
G2p2	65.10	76.59	76.94	75.59
TGL_{sta}	**65.48**	**79.93**	**78.32**	**79.68**

Table 2: Accuracy of zero-shot classification performance (percent).



(a) Arxiv dataset for pre-training and test on Cora dataset

(b) Industrial dataset for pre-training and test on Art dataset

(c) Industrial dataset for pre-training and test on M.I. dataset

Figure 3: Pre-training on one dataset and testing on a different dataset for few-shot classification.

L_{no-to}	L_{te-su}	L_{no-st}	L_{su-st}	Cora	Industrial Art	M.I.
		✓		80.29	86.13	81.36 86.16
✓			✓	80.33	86.43	82.02 86.82
	✓	✓	✓	80.26	86.21	81.68 86.53
✓	✓	✓	✓	**82.26**	**87.12**	**83.10 87.16**
Only label text				79.57	86.60	80.91 85.32
Prompt random initialization				80.37	86.96	82.97 87.06

Table 3: Ablation study.

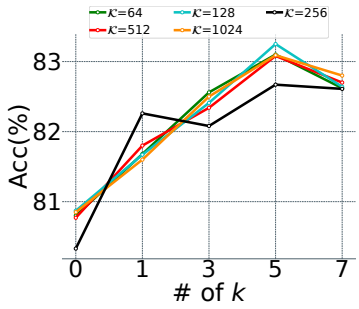


Figure 4: Hyperparameter study of k and $\mathcal{K}$.

Ablation Study

In our detailed analysis of *TGL*$_{sta}$, we focus primarily on node classification accuracy under the few- and zero-shot settings. Our initial study explores the effect of four graph interaction-based contrastive strategies by utilizing different combinations of the proposed loss functions: L_{no-to}, L_{te-su}, L_{no-st}, and L_{su-st}. As shown in Table 3, omitting any loss associated with topology or semantic learning leads to a significant drop in model performance. This highlights the importance of incorporating all four loss functions to effectively capture both topological and semantic information within the graph.

Subsequently, we assess the impact of our prompt-tuning approach. Specifically, we compare *TGL*$_{sta}$ with two ablated versions: one utilizing only label text without trainable prompt vectors, and another with randomly initialized prompt vectors. As detailed in Table 3, relying solely on label text significantly undermines classification performance, demonstrating the importance of learning continuous prompts through prompt-tuning. Moreover, our method, which employs context-based initialization for prompt vectors, consistently outperforms random initialization, suggesting the value of incorporating graph structures in prompt-tuning.

In Figure 4, we explore the effects of the hyperparameter k and $\mathcal{K}$ on performance, where k denotes the local neighborhood of nodes, and $\mathcal{K}$ represents the global neighborhood of nodes. As both k and $\mathcal{K}$ increase, performance generally improves, reaching its peak when $k = 5$ and $\mathcal{K} = 128$. Overall, our findings highlight the critical role of graph information in low-resource node classification, given that graph structures encapsulate rich relationships between documents.

Conclusion

Our study addresses the challenges inherent in low-resource textual graph node classification by proposing a novel approach that leverages pre-trained Large Language Models (LLMs) and a multi-hop contrastive learning strategy. Traditional "pre-train, fine-tune" methods struggle with limited labeled data, leading to poor performance. Our approach harnessing the power of LLMs to transform textual node attributes into comprehensive features, effectively capturing semantic information. Moreover, our multi-hop contrastive learning strategy facilitates the fusion of semantic and topological information, enhancing the model's understanding of text-attributed graphs. Extensive experiments show that our approach consistently outperforms existing methods, offering significant advancements for low-resource textual graph classification, and paving the way for more robust graph-based learning methodologies in real-world applications.

Acknowledgments

Qin Zhang, Xiaowei Li and Ziqi Liu were supported by National Natural Science Foundation of China No. 62206179. Xiaochen Fan was supported by China Postdoctoral Science Foundation No. 2023M742034. Xiaojun Chen was supported by National Natural Science Foundation of China No. 92270122 and 62476174.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, M.; Zhang, W.; Zhang, W.; Chen, Q.; and Chen, H. 2019. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. 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)*, 4217–4226.

Chen, X.; Liu, T.; Fournier-Viger, P.; Zhang, B.; Long, G.; and Zhang, Q. 2024. A fine-grained self-adapting prompt learning approach for few-shot learning with pre-trained language models. *Knowledge-Based Systems*, 111968.

Fang, T.; Zhang, Y. M.; Yang, Y.; and Wang, C. 2022. Prompt tuning for graph neural networks.

Fang, Y.; Fan, D.; Zha, D.; and Tan, Q. 2024. GAugLLM: Improving Graph Contrastive Learning for Text-Attributed Graphs with Large Language Models. *arXiv preprint arXiv:2406.11945*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, S.; Guo, T.; Dai, T.; Qiao, R.; Shu, X.; Ren, B.; and Xia, S.-T. 2023. Open-vocabulary multi-label classification via multi-modal knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 808–816.

Hou, Z.; He, Y.; Cen, Y.; Liu, X.; Dong, Y.; Kharlamov, E.; and Tang, J. 2023. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM web conference 2023*, 737–746.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; and Sun, Y. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 1857–1867.

Huang, K.; and Zitnik, M. 2020. Graph meta learning via local subgraphs. *Advances in neural information processing systems*, 33: 5862–5874.

Ju, W.; Qin, Y.; Yi, S.; Mao, Z.; Zheng, K.; Liu, L.; Luo, X.; and Zhang, M. 2023. Zero-shot node classification with graph contrastive embedding network. *Transactions on Machine Learning Research*.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Liang, W.; Hao, Z.; Liu, H.; and Chen, H. 2024. Boosting Zero-Shot Node Classification via Dependency Capture and Discriminative Feature Learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7360–7364. IEEE.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, Z.; He, X.; Tian, Y.; and Chawla, N. V. 2024a. Can we soft prompt LLMs for graph learning tasks? In *Companion Proceedings of the ACM on Web Conference 2024*, 481–484.

Liu, Z.; Shi, Y.; Zhang, A.; Zhang, E.; Kawaguchi, K.; Wang, X.; and Chua, T.-S. 2024b. Rethinking tokenizer and decoder in masked graph modeling for molecules. *Advances in Neural Information Processing Systems*, 36.

Lu, B.; Gan, X.; Yang, L.; Zhang, W.; Fu, L.; and Wang, X. 2022. Geometer: Graph few-shot class-incremental learning via prototype representation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 1152–1161.

Mai, S.; Hu, H.; and Xing, S. 2020. Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion. In *proceedings of the AAAI conference on artificial intelligence*, volume 34, 164–172.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3: 127–163.

Ni, J.; Li, J.; and McAuley, J. 2019. 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)*, 188–197.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 1150–1160.

Qiu, L.; Zhang, Q.; Chen, X.; and Cai, S. 2024. Multi-Level Cross-Modal Alignment for Image Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14695–14703.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Reinanda, R.; Meij, E.; de Rijke, M.; et al. 2020. Knowledge graphs: An information retrieval perspective. *Foundations and Trends® in Information Retrieval*, 14(4): 289–444.

Sahu, G.; and Vechtomova, O. 2021. Adaptive Fusion Techniques for Multimodal Data. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 3156–3166.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725.

Sun, J.; Zhou, Y.; and Zong, C. 2021. One-shot relation learning for knowledge graphs via neighborhood aggregation and paths encoding. *Transactions on Asian and Low-Resource Language Information Processing*, 21(3): 1–19.

Tan, Z.; Zeng, Q.; Tian, Y.; Liu, Z.; Yin, B.; and Jiang, M. 2024. Democratizing Large Language Models via Personalized Parameter-Efficient Fine-tuning. *arXiv preprint arXiv:2402.04401*.

Tian, Y.; Han, Y.; Chen, X.; Wang, W.; and Chawla, N. V. 2024. TinyLLM: Learning a Small Student from Multiple Large Language Models. *arXiv preprint arXiv:2402.04616*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wang, L.; Li, Y.; Aslan, O.; and Vinyals, O. 2021a. WikiGraphs: A Wikipedia Text-Knowledge Graph Paired Dataset. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, 67–82.

Wang, S.; Ding, K.; Zhang, C.; Chen, C.; and Li, J. 2022. Task-adaptive few-shot node classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1910–1919.

Wang, Z.; Wang, J.; Guo, Y.; and Gong, Z. 2021b. Zero-shot node classification with decomposed graph prototype network. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1769–1779.

Wen, Z.; and Fang, Y. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 506–516.

Wu, L.; Chen, Y.; Shen, K.; Guo, X.; Gao, H.; Li, S.; Pei, J.; Long, B.; et al. 2023. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2): 119–328.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823.

Zeng, H.; Zhou, H.; Srivastava, A.; Kannan, R.; and Prasanna, V. 2019. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*.

Zhang, C.; Yao, H.; Huang, C.; Jiang, M.; Li, Z.; and Chawla, N. V. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 3041–3048.

Zhang, D. C.; Yang, M.; Ying, R.; and Lauw, H. W. 2024a. Text-Attributed Graph Representation Learning: Methods, Applications, and Challenges. In *Companion Proceedings of the ACM on Web Conference 2024*, WWW '24, 1298–1301. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701726.

Zhang, Q.; Li, X.; Lu, J.; Qiu, L.; Pan, S.; Chen, X.; and Chen, J. 2024b. ROG_PL: Robust Open-Set Graph Learning via Region-Based Prototype Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9350–9358.

Zhang, Q.; Liu, Z.; Li, Q.; Xiang, H.; Yu, Z.; Chen, J.; Zhang, P.; and Chen, X. 2024c. Open-world structured sequence learning via dense target encoding. *Information Sciences*, 680: 121147.

Zhang, Q.; Lu, J.; Li, X.; Wu, H.; Pan, S.; and Chen, J. 2024d. CONC: Complex-noise-resistant Open-set Node Classifcation with Adaptive Noise Detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 5481–5489.

Zhang, Q.; Shi, Z.; Zhang, X.; Chen, X.; Fournier-Viger, P.; and Pan, S. 2023. G2Pxy: generative open-set node classification on graphs with proxy unknowns. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 4576–4583.