HeTGB: A Comprehensive Benchmark for Heterophilic Text-Attributed Graphs

Shujie Li[1]*, Yuxia Wu[2]*, Yuan Fang[2], Chuan Shi[1]
[1]Beijing University of Posts and Telecommunications, [2]Singapore Management University
shujieli@bupt.edu.cn, yieshah2017@gmail.com, yfang@smu.edu.sg, shichuan@bupt.edu.cn

Abstract

Graph neural networks (GNNs) have demonstrated success in modeling relational data primarily under the assumption of homophily. However, many real-world graphs exhibit heterophily, where linked nodes belong to different categories or possess diverse attributes, such as webpages, Wikipedia articles, social networks, and e-commerce platforms. Additionally, nodes in many domains are associated with textual descriptions, forming heterophilic text-attributed graphs (TAGs). Despite their significance, heterophilic TAGs remain underexplored due to the lack of dedicated benchmarks that jointly capture heterophilic structures and rich textual attributes. To address this gap, we introduce the **He**terophilic **T**ext-attributed **G**raph **B**enchmark (HeTGB), a novel benchmark comprising five real-world heterophilic graph datasets from diverse domains, with nodes enriched by extensive textual descriptions. HeTGB enables systematic evaluation of GNNs, pre-trained language models (PLMs) and co-training methods on the node classification task. Through extensive benchmarking experiments, we showcase the utility of text attributes in heterophilic graphs, analyze the challenges posed by heterophilic TAGs and the limitations of existing models, and provide insights into the interplay between graph structures and textual attributes.[1]

1 Introduction

Graphs serve as a foundational data structure for representing relational information across diverse domains, including webpages, Wikipedia articles, social networks, citation networks, and knowledge graphs (Wu et al., 2020). Conventional graph neural networks (GNNs) predominantly rely on the assumption of *homophily*, meaning that neighboring nodes on the graph share similar characteristics (Kipf and Welling, 2016; Veličković et al., 2018). However, numerous real-world graphs exhibit *heterophily*, in which linked nodes belong to distinct classes or possess dissimilar attributes (Zheng et al., 2022). In such heterophilic graphs, nodes often contain textual attributes that provide rich semantic information, forming heterophilic text-attributed graphs (TAGs). For example, in the Wikipedia hyperlink graph, articles on diverse topics may link to one another, yet their textual content provides essential context for understanding these connections. However, the data and methodologies tailored for heterophilic TAGs remain underexplored, necessitating a systematic investigation into their unique challenges, requirements and insights. Consequently, there is an increasing demand for benchmarks that jointly evaluate structural and textual modeling under heterophily.

While prior works on heterophilic graph learning have introduced various datasets spanning different domains (Pei et al., 2020; Lim et al., 2021; Platonov et al., 2023), these datasets only contain categorical features or shallow features derived from raw attributes, lacking the rich textual descriptions that could provide deeper semantic insights. Meanwhile, existing approaches to heterophilic graphs mainly focus on GNN-based architectures designed to capture structural patterns, incorporating non-local neighbor expansion or architectural refinements (Zheng et al., 2022; Gong et al., 2024). Notably, these methods do not explicitly integrate textual attributes, limiting their effectiveness in domains where semantic content is crucial. On the other hand, the remarkable advancements of pre-trained language models (PLMs) in text comprehension and generation—ranging from smaller models like BERT (Devlin et al., 2019) to large language models (LLMs)—have driven growing interest in methodologies for TAGs. For instance, the recently proposed CS-TAG benchmark (Yan et al., 2023) provides a comprehensive analysis by

* Co-first authors with equal contribution.

[1]All datasets and baselines are available at https://github.com/honey0219/HeTGB.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 19080–19096
October 24-29, 2026 ©2026 Association for Computational Linguistics

incorporating various learning paradigms. While recent studies explore hybrid architectures that integrate PLMs and GNNs in a unified pipeline (Liu et al., 2024; Tang et al., 2024), their evaluations remain largely confined to homophilic graphs.

Despite recent efforts for both heterophilic graphs and TAGs, a significant research gap persists in two key aspects. (1) *Lack of dedicated heterophilic TAG datasets*: Current heterophilic graph datasets predominantly focus on structural properties while lacking textual content, limiting their ability to evaluate PLM-based and co-training methods. Conversely, most TAG datasets exhibit homophilic connectivity, restricting their relevance for assessing models designed for heterophilic graphs. (2) *Absence of systematic analysis of existing methodologies for heterophilic TAGs.* While many methods exist for heterophilic graphs, it remains unclear whether they can effectively leverage rich textual information. Furthermore, it is uncertain whether state-of-the-art methods for TAGs are sufficiently robust for heterophilic graphs.

To address the above two aspects, we introduce **He**terophilic **T**ext-attributed **G**raph **B**enchmark (HeTGB), which comprises five real-world heterophilic graph datasets from diverse domains, with nodes enriched by extensive textual descriptions. To construct these datasets, we carefully select datasets from various domains and extract their raw texts from publicly accessible data sources. With HeTGB, we provide a systematic evaluation of representative GNN-based, PLM-based and co-training methods on the node classification task. The experimental analysis highlights the challenges and limitations of current algorithms, offering valuable insights into future research directions.

Our main contributions are summarized as follows. (1) We introduce HeTGB, a novel benchmark that integrates heterophilic graph structures with rich textual attributes, addressing the gap left by both existing heterophilic graph benchmarks and text-attributed graph benchmarks. (2) We conduct a systematic evaluation of representative GNN-based, PLM-based and co-training models on the node classification task, identifying key challenges, limitations and insights in heterophilic TAG learning. (3) We publicly release HeTGB with datasets and baseline implementations, to facilitate further research and encourage the development of effective models for heterophilic TAGs.

2 Related Work

Heterophilic graph learning. The widely used benchmarks for heterophilic graphs originate from (Pei et al., 2020), including webpages such as *Texas*, *Cornell*, and *Wisconsin*. Subsequently, Lim et al. (2021) and Platonov et al. (2023) introduced larger and more structurally diverse datasets. More recently, Lin et al. (2025) proposed a large-scale benchmark that considers both heterophily and heterogeneity, further enriching the diversity of graph structures studied in this area. However, these benchmarks predominantly lack rich textual attributes, with node features typically comprising categorical or numerical values, or shallow text representations like bag-of-words or FastText embeddings (Grave et al., 2018), limiting their relevance for text-rich scenarios. On the methods side, GNN-based approaches to heterophily fall into two main categories: neighborhood expansion and architectural refinement. The former enhances expressiveness by incorporating high-order neighborhood aggregation (Song et al., 2023; Yu et al., 2024) or global graph-based identification (Wang et al., 2022; Bi et al., 2024), while the latter improves message propagation by adaptively aggregating information from homophilic and heterophilic neighbors (Bo et al., 2021; Liang et al., 2023) or leveraging inter-layer interactions to capture multi-hop dependency (Chien et al., 2021).

Text-attributed graph learning. TAGs integrate textual content with graph structure, making them suitable for tasks in document classification, citation networks, and knowledge graphs. While benchmarks such as CS-TAG (Yan et al., 2023), dynamic TAGs (Zhang et al., 2024a), and textual-edge graphs (Li et al., 2024) have advanced evaluation in this space, a dedicated benchmark specifically for heterophilic TAGs remains unexplored. On the modeling side, early efforts enhanced node features with static word embeddings, evolving to contextual embeddings using SLMs like BERT (Devlin et al., 2019) and later to LLMs such as GPT-4 and LLaMA (Achiam et al., 2023; Touvron et al., 2023). Integration of PLMs into graph learning has followed two main approaches: PLM-based methods that transform graph data into natural language (Zhao et al., 2023; Ye et al., 2024; Liu and Wu, 2023), and co-training approaches that harness the strengths of both paradigms (He et al., 2024; Tang et al., 2024; Zhang et al., 2024b). Despite these advances, most existing works primarily fo-

cus on homophilic graphs. HeTGB fills this gap by offering a benchmark designed to evaluate the existing methods for heterophilic TAGs. Notably, LLM4HeG (Wu et al., 2025) represents an early attempt to apply PLMs to heterophilic graphs, highlighting the limited exploration in this direction and further emphasizing the need for HeTGB.

3 HeTGB: A Comprehensive Benchmark for Heterophilic TAGs

In this section, we begin with an overview of HeTGB, followed by detailed descriptions of the dataset construction.

3.1 Overview of HeTGB

HeTGB is introduced to address the absence of standardized benchmarks for heterophilic TAGs. As outlined in Table 1(c), HeTGB includes five real-world datasets from diverse domains, ranging from Web pages to e-commerce and social networks. A comparison of HeTGB to previous heterophilic graph benchmarks and text-attributed graphs is shown in Table 1(a) and (b), respectively. In these tables, $\mathcal{H}_e$ denotes the edge homophily ratio. Detailed definitions, dataset statistics, and distribution analyses are provided in Appendix A.

HeTGB facilitates the systematic evaluation of graph learning models including GNNs, PLMs, and co-training methods on the supervised node classification tasks. Node classification represents the most fundamental task in heterophilic graph research (Luan et al., 2022; Zheng et al., 2022), as heterophily is primarily reflected in the interaction between the structural connectivity and the distribution of node class labels. Besides, the benchmark provides standardized data splits, baseline implementations, and performance metrics to ensure reproducible research and foster innovation in heterophilic TAG learning.

3.2 Dataset Construction

To construct HeTGB, we carefully selected five real-world datasets, which are characterized by classic heterophilic structures and associated with rich textual content. The construction process involved sourcing publicly available graph structures and extracting raw textual content from associated metadata. Each dataset was preprocessed into a standardized format with pre-defined splits for training, validation, and testing to facilitate fair comparisons. Note that, since most of the publicly preprocessed datasets used in existing heterophilic graph work lack raw textual content and details of their preprocessing steps, we reconstructed the datasets from their original data sources. As a result, the number of nodes and edges may slightly differ from their preprocessed counterparts. In what follows, we describe the construction details for each dataset.

Cornell, Texas, and Wisconsin. They are widely used subsets of the WebKB dataset[2] collected by Carnegie Mellon University from computer science departments at various universities. Each node corresponds to a Web page, and edges represent hyperlinks between pages. Each node is categorized into seven categories: student, faculty, staff, department, course, project, and other. These graphs were first utilized for heterophilic graph learning by Pei et al. (2020). We obtained a subset of five webpage categories and their hyperlinks from a public repository[3] (Hussain et al., 2021). Since only URLs were provided, we matched them to the original WebKB HTML files[1] and extracted the full text to reconstruct the dataset. Compared to the preprocessed counterparts (Pei et al., 2020) summarized in Table 1(a), the categories of Web pages are the same, namely, student, project, course, staff, and faculty, while the numbers of nodes and edges remain similar as shown in Table 1(c). However, we observe differences in the homophily ratios compared to previous preprocessed versions (Pei et al., 2020). This discrepancy arises due to differences in the node and edge sets, as the dataset provided by Pei et al. (2020) only contains preprocessed node IDs without the full textual content. Consequently, our reconstructed graphs do not correspond one-to-one with theirs, resulting in variations in graph topology and homophily ratios.

Actor. This is an actor-only induced subgraph derived from a film-director-actor-writer network[4], curated by Tang et al. (2009). The dataset was first used for heterophilic graph learning (Pei et al., 2020) without raw text, category names and other preprocessing details. Thus, we rely on the original data source (Tang et al., 2009) crawled from Wikipedia to extract the actor-only subgraph, in which nodes represent actors, and an edge between two nodes indicates their co-occurrence on the

[2] https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

[3] https://github.com/sqrhussain/structure-in-gnn/tree/master/data/graphs/raw/WebKB

[4] https://www.aminer.cn/lab-datasets/soinf/

Table 1: Dataset statistics and comparison with existing datasets.

Dataset	#Class	#Node	#Edge	Domain	Feature	$\mathcal{H}_e(G)$	Hetero.	Text
(a) Previous Heterophilic Graphs								
Cornell (Pei et al., 2020)	5	183	295	Web page	BOW	0.30	✓	×
Texas (Pei et al., 2020)	5	183	309	Web page	BOW	0.06	✓	×
Wisconsin (Pei et al., 2020)	5	251	499	Web page	BOW	0.17	✓	×
Actor (Pei et al., 2020)	5	7600	33544	Social	Keywords	0.22	✓	×
Amazon (Platonov et al., 2023)	5	24492	93050	E-commerce	FastText	0.38	✓	×
Roman-empire (Platonov et al., 2023)	18	22662	32927	Wikipedia	FastText	0.05	✓	×
Tolokers (Platonov et al., 2023)	2	11758	519000	Crowdsourcing	Numerical	0.59	✓	×
(b) Previous Text-attributed Graphs								
Cora (Liu et al., 2024)	7	2708	5429	Academic	PLMs	0.81	×	✓
Pubmed (Liu et al., 2024)	3	19717	44338	Academic	PLMs	0.80	×	✓
Arxiv (Yan et al., 2023)	40	169343	1166243	Academic	PLMs	0.66	×	✓
History (Yan et al., 2023)	12	41551	358574	E-commerce	PLMs	0.66	×	✓
Computers (Yan et al., 2023)	10	97229	721080	E-commerce	PLMs	0.83	×	✓
Sports (Yan et al., 2023)	13	173055	1773500	E-commerce	PLMs	0.90	×	✓
Children (Yan et al., 2023)*	24	76875	1554578	E-commerce	PLMs	0.42	✓	✓
(c) Our HeTGB								
Cornell	5	195	304	Web page	PLMs	0.13	✓	✓
Texas	5	187	328	Web page	PLMs	0.12	✓	✓
Wisconsin	5	265	530	Web page	PLMs	0.20	✓	✓
Actor	5	4416	12172	Social	PLMs	0.56	✓	✓
Amazon	5	24492	93050	E-commerce	PLMs	0.38	✓	✓

* The Children (Yan et al., 2023) exhibits a notably low homophily ratio. Accordingly, we provide additional experiments in Appendix C.

same Wikipedia page. The raw text of each node contains the node ID, the number of words introducing the node on Wikipedia, and the multiple categories extracted from Wikipedia pages. The original source (Tang et al., 2009) shows notable category imbalance, with around 66% of actors labeled as "*American film actors*", and many actors belonging to multiple categories such as "*American film actors*" and "*American television actors*". Inspired by prior preprocessing steps (Tang et al., 2009), we selected actors with high occurrence frequencies and combined related categories to improve balance. Finally, following prior work (Pei et al., 2020), we selected five categories, as follows.

- "American film actors (only)": actors exclusively labeled as "*American film actors*";
- "American film actors and American television actors": category including both "*American film actors*" and "*American television actors*";
- "American television actors and American stage actors": category including both "*American stage actors*" and "*American television actors*";
- "English actors": category including "English" such as "*English film actors*", "*English television actors*" or "*English stage actors*";
- "Canadian actors": category including "*Canadian*".

We removed category terms corresponding to the classification labels from the raw text to prevent label leakage in our classification task. Afterward, we constructed the graph based on the nodes in the five categories and the edges between them and removed isolated nodes from the graph.

Note that our graph has a higher homophily ratio than the previous version (Pei et al., 2020) due to differences in category and graph construction. Ultimately, this is still a form of social network, where actors in the same category co-occur on the same Wikipedia page more frequently, increasing homophily. However, it is still considered heterophilic compared to classic homophilic graphs.

Amazon. It is constructed from the Amazon product co-purchasing network metadata[5] from Stanford Large Network Dataset Collection (Leskovec and Krevl, 2014). Nodes represent products such as books, music CDs, DVDs, and VHS tapes, while edges link products frequently bought together. The classification task is to predict the average product rating, grouped into five classes. This dataset was first used for heterophilic graph learning by Platonov et al. (2023), where preprocessed data included raw textual descriptions such as product name, category, and rating information. However, their rating information includes only metadata (*e.g.*, average rating, time, user ID and corresponding rating). We removed the average rating to prevent label leakage in the classification task. Instead, we retained up to five earliest indi-

[5]https://snap.stanford.edu/data/amazon-meta.html

vidual user ratings per product, along with product descriptions, to synthesize the textual content. Following prior work (Platonov et al., 2023), only the largest connected component of the five-core subgraph is considered.

4 Learning Paradigms for HeTGB

In this section, we introduce the existing learning paradigms for heterophilic graphs and TAGs including GNN-based methods, PLM-based methods, and co-training methods.

4.1 GNN-based Methods

GNN-based methods focus on learning from graph structures. However, they have limited capability in handling textual content, often relying on bag-of-words features or shallow embeddings. Both classic and heterophily-specific GNNs have been proposed, as we elaborate in the following.

Classic GNNs. GNNs typically adopt the message-passing mechanism, where each node representation is updated by aggregating the messages from local neighbors and then combining the aggregated messages with its ego representation (Xu et al., 2018a). This updating process can be stacked over multiple layers to capture higher-order dependencies. Specifically, the representation of node $v \in \mathcal{V}$ in the l-th layer can be obtained by:

$$\mathbf{r}_v^{(l)} = \textsc{Aggr}^{(l)}\left(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}\right), \quad (1)$$

$$\mathbf{h}_v^{(l)} = \textsc{Combine}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{r}_v^{(l)}\right), \quad (2)$$

where $\mathcal{N}(v)$ represents the neighbors of node v, and $\mathbf{r}_v^{(l)}$ and $\mathbf{h}_v^{(l)}$ stand for the message vector and the representation vector of node v at the l-th layer, respectively. $\textsc{Aggr}(\cdot)$ is an aggregation function (*e.g.*, mean and max pooling), and $\textsc{Combine}(\cdot)$ is a combination function (*e.g.*, a linear layer) (Hamilton et al., 2017).

Note that the general message-passing mechanism operates under the assumption of homophily, where neighbors are assumed to have similar attributes or representations as the ego node.

Heterophily-specific GNNs. Unlike classic GNNs, heterophily-specific GNNs are designed to adapt their message passing process to heterophilic graphs. They mainly focus on customizing the neighborhood aggregation and feature updating schemes that specifically model the heterophily characteristics (Zheng et al., 2022). Heterophily-specific GNNs can be broadly categorized into non-local neighbor extension methods (Song et al., 2023; Pei et al., 2020), architecture refinement methods (Bo et al., 2021; Du et al., 2022; Xu et al., 2018b), and hybrid methods (Zhu et al., 2020).

Non-local neighbor extension methods aim to redefine the neighborhood $\mathcal{N}(v)$ in Eq. (2). Various techniques have been proposed, such as adopting high-order neighbor mixing (Abu-El-Haija et al., 2019; Song et al., 2023), and potential neighbor discovery based on various distance measures, $\mathcal{N}_p(v) = \{u : \textsc{Dist}(u, v) \leq \delta\}$ (Pei et al., 2020).

Architecture refinement methods focus on revisiting the $\textsc{Aggr}(\cdot)$ and $\textsc{Combine}(\cdot)$ functions tailored to heterophilic graphs. One line of work adopts adaptive message aggregation by assigning different weights for homophilic and heterophilic neighbors (Bo et al., 2021; Du et al., 2022):

$$\mathbf{r}_v^{(l)} = \textsc{Aggr}^{(l)}\left(\{\alpha_{uv}^{(l)}\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}\right), \quad (3)$$

where $\alpha_{uv}^{(l)}$ denotes the aggregation weight for node pair (u, v) at the l-th layer. Another line of work explores the inter-layer combination to integrate the local information of shallow layers and the global information from deeper layers (Xu et al., 2018b):

$$\mathbf{h}_v = \textsc{Combine}(\mathbf{h}_v^{(1)}, \mathbf{h}_v^{(2)}, \cdots, \mathbf{h}_v^{(L)}). \quad (4)$$

4.2 PLM-based Methods

PLM-based methods harness the advanced language modeling capabilities of PLMs to effectively capture the semantics of textual content on nodes, enabling direct predictions across a broad spectrum of graph-related tasks. Nevertheless, they generally struggle to capture graph structures effectively. A critical component of these methods is transforming graph-structured data into textual descriptions that PLMs can process (Li et al., 2023; Liu et al., 2023). This is typically achieved through carefully designed prompts that encapsulate the task description, node and edge lists, and textual content of each node. However, in heterophilic graphs, prompts that incorporate structural signals (*e.g.,* summaries of neighboring nodes) may introduce conflicting or misleading information due to the heterophilous nature of neighbors (Chen et al., 2024b). This underscores the need for more principled prompt designs that selectively integrate structural information when adapting PLM-based methods to heterophilic settings.

Generally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ with a set of nodes $\mathcal{V}$, edges $\mathcal{E}$ and textual attributes $\mathcal{T}$,

the PLM-based paradigm is formulated as follows:

$$G_{\text{seq}} = \text{FLATTEN}(\mathcal{V}, \mathcal{E}, \mathcal{T}), \quad (5)$$

$$Y = f_{\text{PLM}}(\text{PROMPT}, G_{\text{seq}}), \quad (6)$$

where $\text{FLATTEN}(\cdot)$ is the flatten function to transform the graph $\mathcal{G}$ into a language sequence. The prediction is then obtained by feeding the prompt and flattened graph sequence G_{seq} into a PLM.

4.3 Co-training Methods

Co-training methods aim to integrate the advantages of GNNs for structural representation learning with the powerful semantic modeling capabilities of PLMs. These methods can be further categorized into GNN-centric, symmetric, and PLM-centric approaches (Liu et al., 2023), depending on how they couple the GNN and PLMs.

GNN-centric methods. They leverage PLMs to extract more representative textual features for nodes, while GNNs are primarily responsible for learning graph representations and making predictions (Liu et al., 2024; Wu et al., 2025), as follows.

$$Y = f_{\text{GNN}}(f_{\text{PLM}}(\mathcal{T}), \mathcal{G}), \quad (7)$$

where $f_{\text{PLM}}(\mathcal{T})$ extracts semantic representations from the textual attributes of each node, which are subsequently integrated into a GNN for downstream prediction.

Symmetric methods. They align the structural embeddings generated by GNNs with the semantic ones obtained from PLMs, ensuring consistency between the two modalities (Wen and Fang, 2023):

$$\min \mathcal{L}(f_{\text{GNN}}(\mathcal{G}), f_{\text{PLM}}(\mathcal{T})), \quad (8)$$

where the objective is to minimize the discrepancy between structure- and text-based embeddings.

PLM-centric methods. They utilize GNNs to enhance the performance of PLMs. Recent advancements in this area focus on injecting knowledge of complex graph structures into PLMs to improve their adaptability across various datasets and tasks (Zhang et al., 2024b; Tang et al., 2024):

$$Y = f_{\text{PLM}}(\text{PROMPT}, G_{\text{seq}}, f_{\text{GNN}}(\mathcal{G})). \quad (9)$$

4.4 Discussion

While the message passing framework in classic GNNs excels in learning from graph structures, they face challenges in heterophilic graphs due to inconsistent neighborhood label distributions. Heterophily-specific GNNs address this by extending non-local neighbors and refining message passing via adaptive aggregation or inter-layer fusion. However, they still rely heavily on graph topology and often underutilize rich semantic information from textual content, which is crucial in distinguishing homophilic and heterophilic relationships.

In contrast, PLM-based methods can more effectively leverage the semantic features of textual content. However, they lack intrinsic awareness of graph structures and often resort to graph flattening techniques, which make them less effective in distinguishing homophilic and heterophilic structures. In addition, they often require computationally expensive fine-tuning for adaptation.

Finally, co-training methods attempt to bridge the gap in GNN- and PLM-based methods by integrating GNNs with PLMs, thus benefiting from both graph structure and semantic features. However, their effective integration often depends on expensive fine-tuning, and most existing methods are not specifically targeted at heterophilic graphs. A recent study, LLM4HeG (Wu et al., 2025), achieves significant improvement on heterophilic graphs by incorporating specific designs to distinguish between heterophilic and homophilic neighbors based on semantic features. While it demonstrates the potential of PLMs for heterophilic graph learning, further research is needed to optimize information fusion strategies, enhance structural-semantic consistency, and improve computational efficiency in modeling heterophilic graphs.

5 Benchmarking Results and Analysis

In this section, we evaluate the performance of existing methods on HeTGB and conduct a systematic analysis. More experiments on existing large-scale datasets with relatively low homophily ratios (e.g., Arxiv and Children (Yan et al., 2023)) are provided in Appendix C.

5.1 Experimental Setup

Baselines. We conduct experiments on three learning paradigms. (1) **GNN-based methods**: the classic GCN (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2018), as well as heterophily-specific H2GCN (Zhu et al., 2020), FAGCN (Bo et al., 2021), JacobiConv (Wang and Zhang, 2022), GBK-GNN (Du et al., 2022), OGNN (Song et al., 2023), SEGSL (Zou et al., 2023), and DisamGCL (Zhao et al., 2024).

Table 2: Accuracy benchmarking. The best results for each paradigm are highlighted in distinct colors.

Paradigms		Methods	Cornell	Texas	Wisconsin	Actor	Amazon
GNNs: *Shallow feature*	*w/o Heterophilic*	GCN	51.43±4.9	54.09±1.7	53.68±3.4	63.33±1.3	43.97±0.3
		GraphSAGE	71.43±1.9	68.18±0.2	71.35±4.1	71.09±0.4	47.98±0.3
		GAT	48.10±3.1	57.27±6.8	55.79±6.9	46.23±2.6	43.32±0.1
	Heterophilic	H2GCN	65.71±1.9	79.09±2.7	79.30±1.3	73.86±0.4	50.46±0.3
		FAGCN	73.33±2.8	82.50±5.0	80.35±2.1	73.79±0.6	47.50±0.4
		JacobiConv	71.64±1.3	73.41±4.9	74.20±5.4	72.44±1.2	48.77±0.2
		GBK-GNN	67.62±2.4	79.55±2.5	71.93±2.2	72.10±0.6	47.21±0.6
		OGNN	66.67±3.0	75.91±2.3	77.54±2.3	73.54±1.3	50.24±1.6
		SEGSL	73.33±3.9	80.00±1.9	81.05±1.4	71.36±0.3	46.69±0.4
		DisamGCL	56.67±3.1	65.00±1.2	58.60±2.0	70.60±0.2	44.37±0.1
GNNs: *LLM feature*	*w/o Heterophilic*	GCN	52.86±1.8	43.64±3.3	41.40±1.8	66.70±1.3	39.33±1.0
		GraphSAGE	75.71±1.8	81.82±2.5	80.35±1.3	70.37±0.1	46.63±0.1
		GAT	54.28±5.1	51.36±2.3	50.53±1.7	63.74±6.7	35.12±6.4
	Heterophilic	H2GCN	69.76±3.0	79.09±3.5	80.18±1.9	70.73±0.9	47.09±0.3
		FAGCN	76.43±3.1	84.55±4.8	83.16±1.4	75.58±0.5	49.83±0.6
		JacobiConv	73.57±4.3	81.80±4.1	76.31±1.3	73.81±0.3	49.43±0.5
		GBK-GNN	66.19±2.8	80.00±3.0	72.98±3.3	72.49±1.0	44.90±0.3
		OGNN	71.91±1.8	85.00±2.3	79.30±2.1	72.08±2.4	47.79±1.6
		SEGSL	66.67±4.1	85.00±2.0	79.30±1.8	72.73±0.8	47.38±0.2
		DisamGCL	50.48±2.0	65.00±1.2	57.89±0.0	67.78±0.3	43.90±0.4
PLMs	*w/o Fine-tuning*	Bloom-560M	5.24±2.8	6.82±1.4	7.02±3.1	0.32±0.2	0.03±0.1
		Bloom-1B	10.52±1.5	5.29±2.1	8.28±2.5	0.34±0.5	0.24±0.3
		Vicuna-7B	23.26±3.2	27.03±1.9	25.35±3.7	23.26±1.2	35.89±0.6
	Fine-tuned	Bloom-560M	21.90±3.5	34.55±9.8	44.91±3.3	65.13±1.1	57.85±0.4
		Bloom-1B	27.14±4.9	46.36±1.8	34.39±5.0	64.21±1.0	55.44±0.3
		Vicuna-7B	63.45±2.1	53.38±3.6	78.91±2.8	76.34±0.8	63.00±0.3
		LLaGA	51.27±2.4	58.32±1.9	43.24±3.0	72.09±1.3	38.24±0.7
Co-training	*w/o Heterophilic*	G2P2	43.81±1.2	53.64±1.8	56.14±4.6	61.73±0.3	43.64±0.3
		GraphGPT	30.32±2.0	42.36±1.4	23.58±2.0	40.29±0.9	41.20±0.1
	Heterophilic	LLM4HeG	77.62±2.9	89.09±3.3	86.14±2.1	76.82±0.5	51.53±0.4

(2) **PLM-based methods**: Bloom-560M (Le Scao et al., 2023), Bloom-1B (Le Scao et al., 2023), Vicuna-7B (Zheng et al., 2023) and LLaGA (Chen et al., 2024a). (3) **Co-training methods**: G2P2 (Wen and Fang, 2023) and GraphGPT (Tang et al., 2024) for general TAGs, as well as LLM4HeG (Wu et al., 2025) for heterophilic TAGs. Additional experiments involving Transformer-based models are provided in Appendix D. More details of baselines and implementation can be found in Appendix E and F, respectively. We report Accuracy as the primary metric in Table 2. To provide a more comprehensive evaluation under class imbalance, we additionally report Macro-F1 and Balanced Accuracy for representative methods in Appendix G. All reported experiments are repeated ten times, with mean and standard deviation reported.

5.2 Analysis of GNN-based Methods

To evaluate the ability of GNN-based methods to leverage semantic features in addition to graph structures, we compare the performance of shallow features and LLM-derived features across different GNN-based methods, as shown in Table 2. The results indicate that LLM-derived features generally lead to improved node classification accuracy, particularly for heterophily-specific GNNs. This highlights the advantage of leveraging richer semantic representations from textual node content in differentiating homophilic and heterophilic neighbors. However, the effectiveness of LLM-derived features is not consistent across all GNNs or all datasets, where we may observe a slight performance decrease compared to shallow features in some cases (*e.g.*, H2GCN on Amazon). This variability arises because GNNs aggregate node embeddings into coarse-grained, fixed-dimensional representations, limiting their ability to capture fine-grained semantic knowledge even when enhanced by LLM-derived features. Consequently, while incorporating LLM-derived features can provide additional semantic contexts, it does not always lead to improved performance particularly when structural relationships play a more critical role.

Moreover, compared to classic GNNs, heterophily-specific GNNs achieve overwhelm-

ingly better performance, regardless of whether shallow or LLM-derived features are used. This can be attributed to heterophily-specific designs, including redefining the neighborhood in message passing and adjusting message aggregation to better accommodate heterophilic structures.

5.3 Analysis of PLM-based Methods

On one hand, PLM-based methods without fine-tuning perform poorly, exhibiting limited generalization to graph data and tasks. This is not surprising, as PLMs are pre-trained on plain text corpora, making them unaware of structural dependencies for effective node classification in heterophilic graphs. On the other hand, fine-tuning PLMs often results in a significant performance boost. It indicates that when sufficiently fine-tuned, LLMs can effectively utilize task-relevant semantic patterns in node text, making them a promising solution for heterophilic graphs. Nevertheless, heterophily-specific GNNs with LLM-derived features tend to outperform fine-tuned LLMs, particularly when the homophily ratios are lower, highlighting the importance of structural information in heterophilic graph learning. These findings emphasize the need for task-specific adaptation for LLMs and motivate future research into hybrid co-training methods that integrate message passing in GNNs with LLM-driven reasoning to further enhance performance.

5.4 Analysis of Co-training Methods

To evaluate the effectiveness of co-training methods, we compare the performance of heterophily-specific and non-heterophilic approaches, as shown in Table 2. The results reveal a clear performance gap between general co-training methods and those specifically designed for heterophilic graphs. The non-heterophilic methods struggle to generalize in heterophilic settings, likely due to their lack of explicit adaptation to heterophilic relationships. In particular, they often underperform fine-tuned PLMs, suggesting that directly incorporating structural information without differentiating their homophilic and heterophilic characteristics may be harmful. In contrast, LLM4HeG (Wu et al., 2025) significantly outperforms other co-training methods. This highlights the effectiveness of leveraging both PLMs and GNNs in a heterophily-aware manner, allowing for more effective message passing and representation learning in heterophilic graphs. However, LLM4HeG does not explicitly consider the alignment between graph structures and semantic features (Wen and Fang, 2023), which could further enhance their integration.

Our findings underscore the importance of integrating both structural and textual signals, suggesting that future research could benefit from heterophilic adaptations that enable better alignment between structure- and PLM-based representations.

5.5 Cross-dataset Analysis

To further understand how different learning paradigms behave across datasets, we compare their relative performance under different graph and text characteristics, as shown in Table 2. The results show that no single learning paradigm consistently performs best across all datasets. On the strongly heterophilic WebKB datasets, structure-aware GNNs generally show clear advantages over text-only PLMs, while on Actor and Amazon, fine-tuned PLMs become more competitive and can outperform several GNN-based and co-training methods. This suggests that the relative importance of structural and textual information varies across datasets, and different learning paradigms may be favored under different data characteristics.

Moreover, the performance does not vary monotonically with the graph homophily ratio. For example, both classic and heterophily-specific GNNs perform considerably worse on Amazon than on several WebKB datasets, although Amazon has a higher homophily ratio. This indicates that homophily is insufficient to characterize the difficulty of heterophilic TAG learning, and other dataset properties may also affect model performance.

We further observe that text length does not show a clear relationship with classification performance. For instance, fine-tuned PLMs achieve comparable performance on Wisconsin and Actor despite their substantially different text lengths. These observations suggest that model performance is jointly influenced by structural and textual characteristics rather than by a single dataset property.

5.6 Efficiency Analysis

To further evaluate the computational efficiency of different learning paradigms, we compare the training and inference time of the best-performing method within each paradigm, as shown in Figure 1. GNN-based methods are the most efficient, although using LLM-derived features introduces a moderate overhead compared to shallow features due to more expensive feature processing. In contrast, both PLM-based and co-training methods

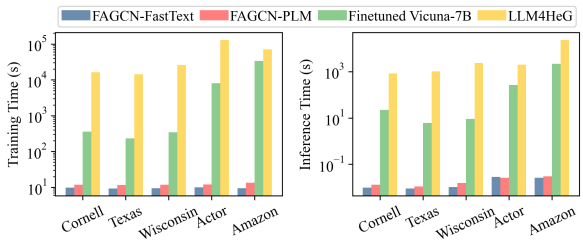


Figure 1: Training and inference times of the best-performing methods across different paradigms.

incur substantially higher computational costs than GNN-based methods. Co-training methods incur the highest costs in both training and inference due to the need to train or fine-tune both PLMs and GNNs. While they achieve strong performance, their computational demands pose challenges for scalability in large-scale graph applications. These results emphasize the trade-off between model expressiveness and efficiency, motivating future research into more efficient co-training approaches without excessive computational overhead.

6 Conclusion

In this work, we introduced HeTGB, a dedicated benchmark for heterophilic text-attributed graphs, addressing the limitations of existing datasets and evaluation frameworks. We collected and provided five datasets from different domains, ensuring high-quality textual content and diverse heterophilic structures. Using these datasets, we systematically investigate the performance of prevailing learning paradigms, including GNN-based, PLM-based, and co-training methods, analyzing their strengths and limitations across different heterophilic datasets. Our experimental results validate the effectiveness of HeTGB as a benchmark and offer valuable insights to guide future research and advancements in heterophilic graph learning.

7 Limitations

While HeTGB provides diverse heterophilic TAG datasets and a unified evaluation framework, several limitations remain. First, the current benchmark focuses primarily on node classification and does not yet cover other relevant tasks such as link prediction, graph classification, and anomaly detection. Second, only a limited number of large-scale heterophilic TAGs are currently included, which may constrain evaluation at larger scales. Extending HeTGB to broader task settings and incorporating additional large-scale datasets are important directions for future work.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62550138, 62192784, 62572064, 62472329)

References

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Wendong Bi, Lun Du, Qiang Fu, Yanlin Wang, Shi Han, and Dongmei Zhang. 2024. Make heterophilic graphs better fit gnn: A graph rewiring approach. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8744–8757.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3950–3957.

Runjin Chen, Tong Zhao, AJAY KUMAR JAISWAL, Neil Shah, and Zhangyang Wang. 2024a. Llaga: Large language and graph assistant. In *Forty-first International Conference on Machine Learning*.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and 1 others. 2024b. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2):42–61.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, volume 1.

Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. 2022. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, pages 1550–1558.

Chenghua Gong, Yao Cheng, Jianxiang Yu, Can Xu, Caihua Shan, Siqiang Luo, and Xiang Li. 2024. A

survey on learning from graphs with heterophily: Recent advances and future directions. *arXiv preprint arXiv:2401.09769*.

Édouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hussain Hussain, Tomislav Duricic, Elisabeth Lex, Roman Kern, and Denis Helic. 2021. On the impact of communities on semi-supervised classification using graph neural networks. In *Complex Networks & Their Applications IX: Volume 2, Proceedings of the Ninth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2020*, pages 15–26. Springer.

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

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2023. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.

Zhuofeng Li, Zixing Gou, Xiangnan Zhang, Zhongyuan Liu, Sirui Li, Yuntong Hu, Chen Ling, Zheng Zhang, and Liang Zhao. 2024. Teg-db: A comprehensive dataset and benchmark of textual-edge graphs. *arXiv preprint arXiv:2406.10310*.

Langzhang Liang, Xiangjing Hu, Zenglin Xu, Zixing Song, and Irwin King. 2023. Predicting global label relationship matrix for graph neural networks under heterophily. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 10909–10921.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902.

Junhong Lin, Xiaojie Guo, Shuaicheng Zhang, Yada Zhu, and Julian Shun. 2025. When heterophily meets heterogeneity: Challenges and a new large-scale graph benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5607–5618.

Chang Liu and Bo Wu. 2023. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, and 1 others. 2023. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2022. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*.

Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. 2023. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. In *The Eleventh International Conference on Learning Representations*.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. 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.

Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Tao Wang, Di Jin, Rui Wang, Dongxiao He, and Yuxiao Huang. 2022. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4210–4218.

Xiyuan Wang and Muhan Zhang. 2022. How powerful are spectral graph neural networks. In *International conference on machine learning*, pages 23341–23362. PMLR.

Zhihao Wen and Yuan Fang. 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*, pages 506–516.

Yuxia Wu, Shujie Li, Yuan Fang, and Chuan Shi. 2025. Exploring the potential of large language models for heterophilic graphs. In *NAACL*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. 2024. Less is more: on the over-globalizing problem in graph transformers. *arXiv preprint arXiv:2405.01102*.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018a. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

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

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, and 1 others. 2023. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1955–1973.

Xingtong Yu, Jie Zhang, Yuan Fang, and Renhe Jiang. 2024. Non-homophilic graph pre-training and prompt learning. *arXiv preprint arXiv:2408.12594*.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. 2024a. Dtgb: A comprehensive benchmark for dynamic text-attributed graphs. *arXiv preprint arXiv:2406.12072*.

Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024b. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In *Proceedings of the ACM on Web Conference 2024*, pages 1003–1014.

Haiteng Zhao, Shengchao Liu, Ma Chang, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. 2023. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in Neural Information Processing Systems*, 36:5850–5887.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2024. Disambiguated node classification with graph neural networks. In *Proceedings of the ACM on Web Conference 2024*, pages 914–923.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804.

Dongcheng Zou, Hao Peng, Xiang Huang, Renyu Yang, Jianxin Li, Jia Wu, Chunyang Liu, and Philip S Yu. 2023. Se-gsl: A general and effective graph structure learning framework through structural entropy optimization. In *Proceedings of the ACM Web Conference 2023*, pages 499–510.

A Distribution and Statistics

Unlike previous heterophilic graph datasets that primarily rely on shallow embeddings derived from textual or numerical features, HeTGB utilizes raw

text collected directly from original data sources. This results in expected variations in key statistics such as the number of nodes and edges, as shown in Table 1. As HeTGB is specifically designed for the investigation of heterophilic structures enriched with rich textual information, we further examine the distribution of the textual content and heterophilic contexts associated with nodes.

First, we analyze the distribution of node text lengths in Figure 2. The distributions highlight the varying extent of textual information associated with nodes, ranging from shorter descriptions in the Actor dataset to more extensive textual content in the Wisconsin and Cornell datasets. Notably, the Cornell, Texas, and Wisconsin datasets exhibit long-tail distributions, characterized by a few nodes with significantly longer text compared to the majority. This is because the raw text consists of entire web pages, which vary in length depending on the individual authors. In contrast, the Actor and Amazon datasets display distributions closer to Gaussian, with text lengths concentrated around the mean, as indicated by the red dashed lines representing the average. This is because their raw text is manually extracted from original pages or user reviews, leading to a more consistent text length distribution. The variation in text lengths and the use of different text extraction or synthesis processes allow for a diverse benchmark for evaluating models' ability to handle heterophilic graphs with varying degrees of textual complexity.

Additionally, we present the distribution of node homophily ratios in Figure 3, offering deeper insights into the heterophily characteristics of our datasets. In HeTGB, each dataset is characterized by heterophilic graph structures and rich text. A heterophilic graph is characterized by its low *homophily ratio* indicating that linked nodes are more likely to have dissimilar features and belong to different classes. The homophily ratio can be defined based on edges (Zhu et al., 2020), representing the proportion of edges connecting nodes with the same class label across the entire graph. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V}$ and edges $\mathcal{E}$, the edge homophily ratio $\mathcal{H}_{\text{e}}$ is computed as

$$\mathcal{H}_{\text{e}}(\mathcal{G}) = \frac{|\{(v,u)\in\mathcal{E}:y_v=y_u\}|}{|\mathcal{E}|}, \tag{10}$$

where y_v denotes the label of node v.

While the edge homophily ratio in Eq. (10), $\mathcal{H}_e$, describes the global homophily characteristics of the entire graph, the node homophily ratio (Pei et al., 2020), $\mathcal{H}_{\text{n}}(v, \mathcal{G})$, captures the local homophily pattern at each node v in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$\mathcal{H}_{\text{n}}(v, \mathcal{G}) = \frac{|\{u\in\mathcal{N}(v):y_v=y_u\}|}{|\mathcal{N}(v)|}. \tag{11}$$

where $\mathcal{N}(v)$ denotes the set of neighbors of node v. The Cornell, Texas, and Wisconsin datasets exhibit highly skewed distributions across nodes with a strong presence of low-homophily nodes, reinforcing their heterophilic nature. In contrast, the Actor and Amazon datasets display a more mixed distribution, capturing both homophilic and heterophilic patterns across nodes. These observations validate that our datasets are characterized by diverse heterophilic graph structures, enabling robust evaluation of models designed for heterophilic TAGs.

B Data Integrity, Privacy, and Label Leakage Analysis

Data sources and privacy. HeTGB is reconstructed entirely from publicly accessible data sources without collecting new data from human participants. Cornell, Texas, and Wisconsin are reconstructed from the original WebKB webpages[6], with their graph structures and URLs obtained from a public repository[7]. Actor is reconstructed from the publicly released film-director-actor-writer network[8], whose textual information is derived from Wikipedia. Amazon is constructed from the Amazon product co-purchasing metadata provided by the Stanford Large Network Dataset Collection[9]. Since the textual attributes originate from real-world public sources, they may retain publicly available person-related or user-contributed information. Our preprocessing focuses on benchmark reconstruction and the removal of task-specific label information rather than exhaustive de-identification or content moderation.

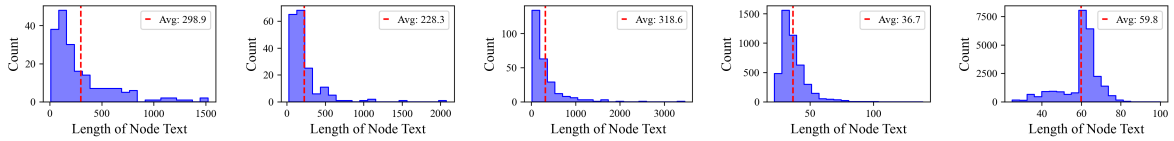
Preprocessing against direct label leakage. We explicitly remove information that directly reveals the prediction targets when constructing the textual attributes. For Cornell, Texas, and Wisconsin,

[6] https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

[7] https://github.com/sqrhussain/structure-in-gnn/tree/master/data/graphs/raw/WebKB

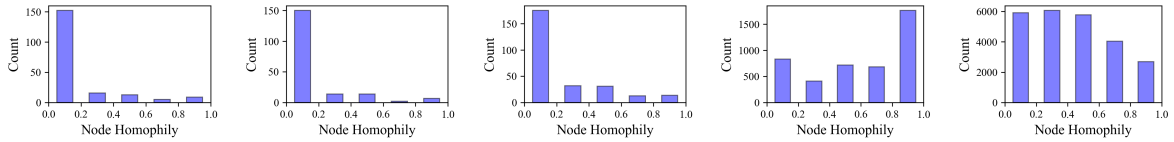
[8] https://www.aminer.cn/lab-datasets/soinf/

[9] https://snap.stanford.edu/data/amazon-meta.html



(a) Cornell (b) Texas (c) Wisconsin (d) Actor (e) Amazon

Figure 2: Distribution of node text lengths, in terms of the number of tokens in each node's textual description.



(a) Cornell (b) Texas (c) Wisconsin (d) Actor (e) Amazon

Figure 3: Distribution of node homophily ratios.

node text is extracted from the original WebKB webpages rather than generated from class labels. For Actor, category terms corresponding to the five prediction labels are removed from the raw text. For Amazon, we remove the final average product rating, which defines the prediction label, and retain only product descriptions and up to five early individual ratings.

Residual label leakage analysis. For Cornell, Texas, and Wisconsin, we examine occurrences of the class names *student*, *course*, *project*, *staff*, and *faculty* in the raw webpage text. In the three datasets, respectively, 24.6%, 40.1%, and 35.5% of nodes contain none of these terms; 47.2%, 30.5%, and 32.1% contain at least one class-name term inconsistent with the ground-truth class; and 30.3%, 17.1%, and 18.9% contain multiple class-name terms. Thus, these naturally occurring terms do not provide a deterministic lexical mapping to the labels. These statistics are not mutually exclusive, as a webpage may satisfy more than one of these conditions. For Actor, after removing explicit category terms, we further audit indirect lexical cues associated with nationality and acting domains. Among all 4,416 nodes, 83.4% contain no detected cues plausibly associated with the ground-truth category, 13.9% contain potentially consistent cues, and 2.7% contain cues associated with a different category. Hence, most Actor instances cannot be classified through straightforward matching of such indirect cues. For Amazon, we measure the dependence between the retained early ratings X and the final rating category Y using normalized mutual information $I(X;Y)/H(Y)$. The resulting value is 21.3%, indicating that early ratings provide partial label-related information but do not deterministically reveal the target class. Taken together, these analyses suggest that, although the textual attributes may contain naturally occurring label-related cues, they do not trivially determine the ground-truth labels.

C Evaluation on Existing Benchmarks

In this section, we further conduct experiments on existing large-scale datasets with relatively low homophily ratios, such as Arxiv and Children (Yan et al., 2023). This setting complements our benchmark by introducing larger graphs with different structural and semantic distributions. We evaluate representative methods across three learning paradigms: GNNs (with both shallow and LLM-derived features), PLMs (zero-shot and fine-tuned), and co-training methods. The results are summarized in Table 3. We observe consistent patterns with our main benchmark: LLM-derived features tend to improve GNN performance; fine-tuned PLMs outperform zero-shot variants; and co-training methods such as G2P2 and LLM4HeG maintain strong performance across datasets. These findings further support the applicability of the studied methods beyond the proposed benchmark datasets.

Table 3: Accuracy on Arxiv and Children datasets.

Paradigms	Method	Arxiv	Children
GNNs (Shallow features)	GraphSAGE	73.51±0.2	57.84±0.5
	FAGCN	64.72±0.5	44.91±0.5
GNNs (LLM features)	GraphSAGE	73.64±0.5	58.40±0.6
	FAGCN	71.89±0.8	53.70±0.7
PLMs w/o Fine-tuning	Vicuna-7B	4.32±0.8	5.73±0.3
PLMs Fine-tuned	Vicuna-7B	73.80±0.9	59.46±0.6
Co-training	G2P2	79.84±1.3	55.91±1.2
	LLM4HeG	72.79±0.8	56.46±0.7

D Evaluation on Graph Transformers

Graph Transformers have emerged as an alternative to message-passing GNNs for graph representation learning. By leveraging self-attention mechanisms, these models can capture interactions between distant nodes and model long-range dependencies, which can be beneficial for heterophilic graphs, where informative signals may not lie in immediate neighborhoods.

To further broaden our evaluation, we additionally include a graph transformer baseline, CoBFormer (Xing et al., 2024), which addresses the over-globalizing problem in Graph Transformers by introducing a bi-level architecture with inter-cluster and intra-cluster attention. Consistent with the feature settings used for other models, we evaluate CoBFormer with both shallow and LLM-derived features. We conduct experiments on both the datasets in our proposed HeTGB benchmark and the additional datasets (Arxiv and Children) used in Appendix C.

The results show that incorporating LLM-derived textual features generally improves the performance of CoBFormer, yielding gains on six of the seven datasets, while a slight decrease is observed on Cornell. Overall, CoBFormer achieves performance comparable to heterophily-specific GNN baselines. These observations further highlight the potential value of rich textual semantics when learning on heterophilic graphs.

Table 4: Accuracy of CoBFormer on HeTGB datasets and additional benchmarks.

Dataset	**CoBFormer** (Shallow features)	**CoBFormer** (LLM features)
Cornell	66.19±3.9	64.33±2.1
Texas	73.82±4.4	81.82±3.6
Wisconsin	77.54±1.9	78.95±2.5
Actor	73.45±0.5	73.86±1.0
Amazon	49.25±0.7	52.56±1.0
Arxiv	69.70±0.4	71.30±0.3
Children	52.28±0.5	54.07±0.4

E Descriptions of Baselines

- GCN (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017) and GAT (Veličković et al., 2018) are widely used classic GNNs that aggregate information from local neighborhoods to learn node representations.
- H2GCN (Zhu et al., 2020) enhances heterophilic graph learning by incorporating higher-order neighbors, ego-neighbor embedding separation, and intermediate-layer representations.
- FAGCN (Bo et al., 2021) employs a self-gating mechanism to adaptively integrate low- and high-frequency signals, enabling robust learning across both homophilic and heterophilic graphs.
- JacobiConv (Wang and Zhang, 2022) eliminates nonlinearity and utilizes Jacobi polynomial bases for spectral filtering, improving flexibility and expressiveness in graph signal learning.
- GBK-GNN (Du et al., 2022) applies a learnable kernel selection mechanism to differentiate homophilic and heterophilic node pairs, optimizing neighborhood aggregation.
- OGNN (Song et al., 2023) introduces an ordered gating mechanism for message passing, enhancing node interactions while mitigating over-smoothing in heterophilic graphs.
- SEGSL (Zou et al., 2023) refines graph topology using structural entropy and encoding trees, improving robustness against noisy edges and adversarial attacks.
- DisamGCL (Zhao et al., 2024) employs topology-aware contrastive learning to disambiguate node embeddings, addressing representation challenges in heterophilic and noisy graphs.
- G2P2 (Wen and Fang, 2023) enhances graph-text alignment via contrastive pre-training and prompt-based strategies, benefiting low-resource text classification tasks.
- GraphGPT (Tang et al., 2024) aligns graph structures with LLMs using self-supervised and task-specific instruction tuning, improving graph reasoning capabilities.
- LLaGA (Chen et al., 2024a) restructures graph nodes into token-compatible sequences, enabling LLMs to process graph data while preserving general-purpose adaptability.
- LLM4HeG (Wu et al., 2025) integrates LLMs into GNNs for heterophilic TAGs, using edge discrimination and adaptive edge reweighting to enhance node classification.

F More Experimental Setup

Data splits and hyperparameters. Following standard practice in prior studies (Zhu et al., 2020), we randomly split the nodes into training, validation, and test sets with a ratio of 48%/32%/20% for

Cornell, Texas, Wisconsin, and Actor. For Amazon, we adopt a 50%/25%/25% split, consistent with prior work (Platonov et al., 2023). The hyperparameters of each method are selected following or tuned based on their respective original papers.

Implementation Details. GNN-based methods are primarily implemented based on publicly available code repositories. To assess their effectiveness in leveraging different types of node representations, we benchmark them with initial node features derived from shallow models (FastText (Grave et al., 2018)) and LLMs (Vicuna-7B (Zheng et al., 2023)). For PLM-based methods, including Bloom-560M, Bloom-1B (Le Scao et al., 2023), and Vicuna-7B (Zheng et al., 2023), we design prompts that directly incorporate textual node attributes, following the structured template used in LLM4HeG (Wu et al., 2025). Our experiments follow two strategies: (1) zero-shot inference, where we directly input the prompts and data into the PLMs for answer generation, and (2) fine-tuning with LoRA (Hu et al., 2022) to better adapt to the node classification task on heterophilic graphs. For the co-training methods, we use the official implementations provided by the respective authors.

Prompt Template. As mentioned in 4.2, we adopt a prompt without structural information to avoid introducing conflicting semantics from heterophilous neighbors (Chen et al., 2024b). We adopt the prompt format used in LLM4HeG (Wu et al., 2025) for PLM-based methods, including Bloom-560M, Bloom-1B (Le Scao et al., 2023), and Vicuna-7B (Zheng et al., 2023). An example prompt for the Cornell, Texas, and Wisconsin datasets is provided below.

> *Background:* I have a dataset containing web page information collected from computer science department websites of various universities. These web pages have been manually categorized into five categories, including student, staff, faculty, course, and project.
>
> *Task:* I will provide you with the text information of a web page, and I would like you to classify it into one of the following categories: student, staff, course, faculty, or project.
>
> The web page content: <text>
>
> You may only output the category name, and do not discuss anything else!

For LLaGA (Chen et al., 2024a), GraphGPT (Tang et al., 2024), and LLM4HeG (Wu et al., 2025), we follow the prompt design as outlined in the official formats provided by the respective authors. Below are the specific prompt formats for each method, using the Cornell, Texas, and Wisconsin datasets as examples.

LLaGA:

> Given a node-centered graph: <node sequence>, each node represents a web page, we need to classify the center node into one of the following categories: student, staff, course, faculty, or project.
>
> Please tell me which class the center node belongs to.

GraphGPT:

> Given a node-centered graph: <graph>, where the 0th node is the target web page, with the following information: <text>
>
> *Question*: Each node represents a web page, and we need to classify the center node into one of the following categories: student, staff, course, faculty, or project.
>
> Please tell me which class the center node belongs to.

LLM4HeG:

> *Background:* I have a dataset containing web page information collected from computer science department websites of various universities. These web pages have been manually categorized into five categories, including student, staff, faculty, course, and project.
>
> *Task:* I will provide you with the content of two web pages, and I want you to determine if they belong to the same category among student, staff, course, faculty, and project.
>
> The first web page: <text>
> The second web page: <text>
>
> *Answer template*: Yes or No. Please think step by step.

For other datasets, we simply modify the dataset description and category information in the prompt. The rest of the prompt structure remains unchanged.

G Additional Evaluation Metrics

To complement the Accuracy results reported in the main text, we further evaluate representative methods using Macro-F1 and Balanced Accuracy. These metrics provide additional perspectives on per-class performance, particularly for datasets with imbalanced class distributions. For open-ended PLM-based methods and GraphGPT, whose generated outputs are not strictly constrained to the predefined label space, Macro-F1 is less directly comparable with closed-set classifiers. We therefore report Balanced Accuracy only for these methods.

The additional metrics reveal performance differences that are less visible from Accuracy alone. For example, GCN with shallow features achieves

Table 5: Macro-F1 (%) of representative methods on HeTGB. Results are reported as mean ± standard deviation over 10 runs.

Paradigm	Method	Cornell	Texas	Wisconsin	Actor	Amazon
GNNs: Shallow feature	GCN	39.31±6.3	50.25±2.4	42.37±3.6	38.29±1.2	27.14±0.4
	H2GCN	44.27±2.4	62.90±4.3	58.40±1.6	55.93±0.8	30.49±0.5
	FAGCN	57.09±1.6	63.57±8.3	64.63±1.9	53.08±0.7	26.96±1.1
GNNs: LLM feature	GCN	41.29±4.3	52.83±4.9	41.26±3.0	42.29±1.1	29.20±0.4
	H2GCN	58.29±3.9	61.29±3.9	65.27±1.0	50.29±0.8	27.05±1.1
	FAGCN	59.93±6.8	68.89±3.4	63.29±5.3	57.63±0.3	31.85±1.1
Co-training	LLM4HeG	63.45±1.1	71.12±3.7	67.17±2.3	61.58±0.9	32.71±0.8

Table 6: Balanced Accuracy (%) of representative methods on HeTGB. Results are reported as mean ± standard deviation over 10 runs.

Paradigm	Method	Cornell	Texas	Wisconsin	Actor	Amazon
GNNs: Shallow feature	GCN	40.25±5.9	49.13±2.3	44.26±2.7	35.24±2.9	29.15±0.9
	H2GCN	43.20±4.2	66.10±5.9	58.29±1.1	50.20±0.7	26.19±0.3
	FAGCN	57.65±1.2	63.94±7.7	64.97±2.7	50.56±0.6	30.00±0.6
GNNs: LLM feature	GCN	40.29±4.2	52.47±2.1	40.37±2.0	45.23±1.4	33.30±0.2
	H2GCN	59.36±5.2	64.58±2.7	67.13±1.4	49.20±0.3	30.22±0.3
	FAGCN	60.24±5.1	67.35±4.5	62.80±5.4	55.92±0.4	33.13±0.8
PLMs: w/o Fine-tuning	Bloom-560M	4.79±3.1	3.98±1.5	7.65±3.9	0.43±0.4	0.09±0.9
	Bloom-1B	10.56±2.5	2.91±2.4	10.50±4.3	0.33±0.4	0.18±0.6
PLMs: Fine-tuned	Bloom-560M	11.11±1.9	15.07±4.6	24.33±1.7	52.38±0.7	51.57±0.5
	Bloom-1B	12.89±2.2	19.18±1.4	21.04±4.8	52.13±1.1	48.60±0.4
Co-training	GraphGPT	15.09±1.4	18.27±1.0	16.09±3.3	33.86±2.1	33.47±0.4
	LLM4HeG	61.30±1.3	68.50±4.5	64.50±1.4	54.17±0.4	34.59±0.2

43.97% Accuracy but only 27.14% Macro-F1 on Amazon, indicating weaker class-wise performance. Nevertheless, the main benchmarking trends remain broadly consistent across metrics: heterophily-specific GNNs generally outperform vanilla GCN, while LLM-derived features provide gains in many settings. The noticeable variance of some PLM-based methods also highlights the importance of repeated evaluation.

H Analysis of Prompt Design

To further investigate the effect of prompt design on PLM-based methods, we conduct a controlled study using zero-shot Vicuna-7B with four prompt configurations. Our default *Text-only* prompt in Appendix F contains only the textual attribute of the target node and excludes structural information. We compare it with variants that introduce either explicit reasoning instructions or neighborhood information. Except for the prompt construction, all other experimental settings are kept unchanged.

Specifically, we consider the following variants:

- **Text-only**: the default prompt used in our main experiments.
- **Text-only + CoT**: the same textual input as the default prompt, with an additional instruction to reason step by step before producing the final prediction.
- **Raw neighbors**: we randomly sample up to three one-hop neighbors of the target node and append their textual attributes to the prompt.
- **Selected neighbors**: we rank the one-hop neighbors according to the cosine similarity between their Vicuna-7B-derived textual representations and that of the target node, and retain the top three most relevant neighbors.

For both neighborhood-based variants, we use all available neighbors when the node degree is smaller than three, ensuring that the two settings differ only in how neighboring nodes are selected.

The results in Table 7 show that PLM performance is sensitive to how textual and structural information is incorporated into the prompt. Adding chain-of-thought instructions leads to limited and inconsistent changes across datasets, suggesting that reasoning instructions alone do not consistently improve classification performance. In contrast, incorporating randomly sampled neighbors generally

Table 7: Effect of different prompt designs for Vicuna-7B on HeTGB. Results are reported as mean Accuracy (%) ± standard deviation over 10 runs.

Prompt Design	Cornell	Texas	Wisconsin	Actor	Amazon
Text-only	23.26±3.2	27.03±1.9	25.35±3.7	23.26±1.2	35.89±0.6
Text-only + CoT	23.10±3.0	27.86±2.2	26.18±3.4	22.45±1.1	37.31±0.7
Raw neighbors	17.74±3.5	24.36±2.5	21.71±3.9	23.48±1.3	35.21±0.8
Selected neighbors	25.31±3.1	28.14±2.0	27.42±3.3	24.67±1.1	35.02±0.7

degrades performance on the strongly heterophilic WebKB datasets, indicating that unfiltered neighborhood information may introduce conflicting semantics. Selecting semantically relevant neighbors effectively mitigates this issue and improves over the text-only baseline on Cornell, Texas, Wisconsin, and Actor, although the improvement does not extend to Amazon. These results suggest that neighborhood information can benefit PLM-based methods when incorporated selectively, while directly introducing heterophilic neighbors may be detrimental.

Prompt modifications. The prompt variants follow the same dataset descriptions and candidate label spaces as the default prompt in Appendix F. For *Text-only + CoT*, we keep the original textual input unchanged and replace the final instruction with:

> Please analyze the webpage content step by step and determine the most likely category. In the final line, only output the category name.

For the two neighborhood-based variants, we augment the default prompt with the target node and up to three neighboring nodes using the following format:

> The target web page: <target text>
>
> The neighboring web pages:
> <neighbor 1 text>
> <neighbor 2 text>
> <neighbor 3 text>
>
> Note that neighboring web pages may belong to different categories from the target page.
>
> You may only output the category name, and do not discuss anything else!

The same neighbor-augmented prompt is used for both *Raw neighbors* and *Selected neighbors*; only the neighbor selection strategy differs. For other datasets, we modify only the dataset description and candidate categories accordingly.