Article Classification with Graph Neural Networks and Multigraphs

Khang Ly, Yury Kashnitsky, Savvas Chamezopoulos, Valeria Krzhizhanovskaya
Elsevier B.V., Elsevier B.V., Elsevier B.V., Unversity of Amsterdam
{k.ly, y.kashnitskiy, s.chamezopoulos}@elsevier.com, v.krzhizhanovskaya@uva.nl

Abstract

Classifying research output into context-specific label taxonomies is a challenging and relevant downstream task, given the volume of existing and newly published articles. We propose a method to enhance the performance of article classification by enriching simple Graph Neural Network (GNN) pipelines with multi-graph representations that simultaneously encode multiple signals of article relatedness, e.g. references, co-authorship, shared publication source, shared subject headings, as distinct edge types. Fully supervised transductive node classification experiments are conducted on the Open Graph Benchmark `OGBN-arXiv` dataset and the `PubMed` diabetes dataset, augmented with additional metadata from Microsoft Academic Graph and PubMed Central, respectively. The results demonstrate that multi-graphs consistently improve the performance of a variety of GNN models compared to the default graphs. When deployed with SOTA textual node embedding methods, the transformed multi-graphs enable simple and shallow 2-layer GNN pipelines to achieve results on par with more complex architectures.

Keywords: Heterogeneous Graph Learning, Graph Neural Networks, Article Classification, Document Relatedness

1. Introduction

Article classification is a challenging downstream task within natural language processing (NLP) (Mirończuk and Protasiewicz, 2018). An important practical application is classifying existing or newly-published articles according to specific research taxonomies. The task can be approached as a graph node classification problem, where nodes represent articles with corresponding feature mappings, and edges are defined by a strong signal of article relatedness, e.g. citations/references. Conventionally, graph representation learning is handled in two phases: unsupervised node feature generation, followed by supervised learning on said features using the graph structure. Graph neural networks (GNNs) can be successfully employed for the second phase of such problems, being capable of preserving the rich structural information encoded by graphs. In recent years, prolific GNN architectures have achieved strong performance on citation network benchmarks (Kipf and Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Frasca et al., 2020; Li et al., 2021).

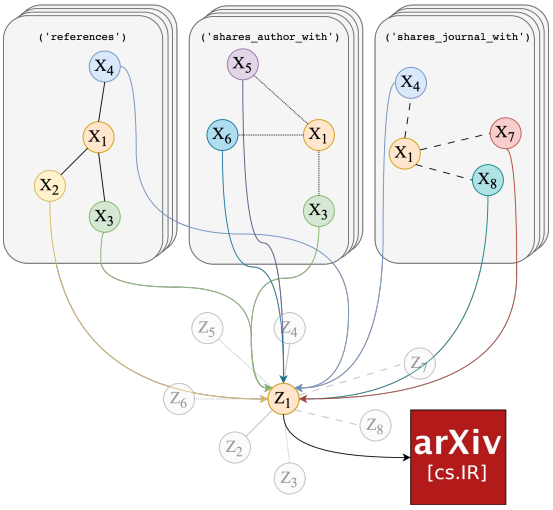

Figure 1: Illustration of the proposed multi-graph input, which enables the neighboring feature aggregation for a node X_1 to be performed across a variety of subgraphs, leveraging multiple signals of article relatedness (References, Authorship, and shared Journal depicted here).

We focus on combining textual information from articles with various indicators of article relatedness (citation data, co-authorship, subject fields, and publication sources) to create a graph with multiple edge types, also known as multi-graphs or heterogeneous graphs (Barabási and Pósfai, 2017). We use two established node classification benchmarks - the citation graphs `OGBN-arXiv` and `PubMed` - and leverage their connection to large citation databases - Microsoft Academic Graph (MAG) and PubMed Central - to retrieve the metadata fields and enrich the graph structure with additional edge types (Hu et al., 2020; Sen et al., 2008). For node feature generation, we experiment with two approaches based on language model (LM) fine-tuning for graph representation learning - SimTG and TAPE - to infer embeddings based on articles' titles and abstracts, with the intention of capturing higher-order semantics compared to the defaults (Duan et al., 2023; He et al., 2024). We test our transformed graphs with a variety of GNN backbone models, converted to support heterogeneous input using the relational graph convolutional network (R-GCN) framework (Schlichtkrull et al., 2018). In essence, we approach a typically ho-

LREC-COLING 2024, pages 1539–1547

mogeneous task using heterogeneous techniques. The method is intuitively simple and interpretable; we do not utilize complex model architectures and training frameworks, focusing primarily on data retrieval and preprocessing to boost the performance of simpler models, thus maintaining a reasonably low computational cost and small number of fitted parameters.

A considerable volume of research is devoted to article classification, graph representation learning with respect to citation networks, and the adaptation of these practices to heterogeneous graphs (Wu et al., 2019b; Bing et al., 2022). However, the application of *heterogeneous* graph enrichment techniques to article classification is not well-studied and presents a research opportunity. Existing works on heterogeneous graphs often consider multiple node types, expanding from article to entity classification; we exclusively investigate the heterogeneity of paper-to-paper relationships to remain consistent with the single-node type problem setting. The emergence of rich metadata repositories for papers, e.g. OpenAlex, illustrates the relevance of our research (Priem et al., 2022).

Scalability is often a concern with GNN architectures. For this reason, numerous approaches simplify typical GNN architectures with varying strategies, e.g. pre-computation or linearization, without sacrificing significant performance in most downstream tasks (Frasca et al., 2020; Wu et al., 2019a; Prieto et al., 2023). Other solutions avoid GNNs altogether, opting for simpler approaches based on early graph-based techniques like label propagation, which outperform GNNs in several node classification benchmarks (Huang et al., 2021). The success of these simple approaches raises questions about the potential impracticality of deep GNN architectures on large real-world networks with a strong notion of locality, and whether or not such architectures are actually necessary to achieve satisfactory performance.

Compared to simple homogeneous graphs, heterogeneous graphs encode rich structural and semantic information, and are more representative of real-world information networks and entity relationships (Bing et al., 2022). For example, networks constructed from citation databases can feature relations between papers, their authors, and shared keywords, often expressed in an RDF triple, e.g. "*paper* $\xrightarrow{\text{(co-)authored by}}$ *author*," "*paper* $\xrightarrow{\text{includes}}$ *keyword*," "*paper* $\xrightarrow{\text{cites}}$ *paper*." Heterogeneous GNN architectures share many similarities with their homogeneous counterparts; a common approach is to aggregate feature information from local neighborhoods, while using additional modules to account for varying node and/or edge types (Yang et al., 2022). Notably, the relational graph convolutional network approach (R-GCN) by Schlichtkrull et al. (2018) shows that GCN-based frameworks can be effectively applied to modeling relational data, specifically for the task of node classification. The authors propose a modeling technique where the message passing functions are duplicated and applied individually to each relationship type. This transformation can be generalized to a variety of GNN convolutional operators in order to convert them into their relational (heterogeneous) counterparts.

2. Methodology

We propose an approach focusing on dataset provenance, leveraging their linkage to large citation and metadata repositories, e.g. MAG and PubMed Central, to retrieve additional features and enrich their graph representations. The proposed method is GNN-agnostic, compatible with a variety of model pipelines (provided they can function with heterogeneous input) and textual node embedding techniques (results are presented with the provided features, plus the SimTG and TAPE embeddings). Figure 1 provides a high-level overview of the method.

The tested GNN backbones (see Section 3) are converted to support heterogeneous input using the aforementioned R-GCN transformation defined by Schlichtkrull et al. (2018), involving the duplication of the message passing functions at each convolutional layer per relationship type; we employ the PyTorch Geometric (PyG) implementation of this technique, using the mean as the aggregation operator (Fey and Lenssen, 2019).

2.1. Datasets

Our experiments are conducted on two datasets: the Open Graph Benchmark (OGB) `OGBN-arXiv` dataset and the `PubMed` diabetes dataset.

The OGB `OGBN-arXiv` dataset consists of 169,343 Computer Science papers from arXiv, hand-labeled into 40 subject areas by paper authors and arXiv moderators, with 1,166,243 reference links (Hu et al., 2020). Default node features are constructed from textual information by averaging the embeddings of words (which are generated with the Skip-Gram model) in the articles' titles and abstracts. The dataset provides the mapping used between papers' node IDs and their original MAG IDs, which can be used to retrieve additional metadata.

The `PubMed` diabetes dataset consists of 19,717 papers from the National Library of Medicine's (NLM) PubMed database labeled into one of three categories: "Diabetes Mellitus, Experimental," "Diabetes Mellitus Type 1," and "Diabetes Mellitus Type 2," with 44,338 references links (Sen et al., 2008). TF-IDF weighted word vectors from a dictionary

Dataset	Edge Type	$\|N\|_{LCC}$	$\|E\|_{LCC}$	$\|E\|$	Avg. Degree	Avg. Clust. Coeff.	Homophily
OGBN-arXiv	References	169,343	2,315,598	2,315,598	13.7	0.310	0.654
	Authorship	145,973	6,697,998	6,749,335	39.9	0.775	0.580
	Source	63	3,906	605,660	3.6	1	0.590
	Subject Area	144,529	8,279,492	8,279,687	48.9	0.630	0.319
PubMed	References	19,716	88,649	88,649	4.5	0.246	0.802
	Authorship	17,683	729,468	731,376	37.1	0.705	0.721
	Source	2,213	4,895,156	11,426,930	579.6	1	0.414
	Subject Area	18,345	1,578,526	1,578,530	80.1	0.481	0.550

Table 1: Properties of constructed subgraphs: number of nodes and edges (post-conversion to undirected) in the largest connected component (LCC), total number of edges, average degree, network average clustering coefficient (Schank and Wagner, 2005), and edge homophily ratio (fraction of edges connecting nodes with the same label) (Ma et al., 2022). Note that the References subgraphs are the only ones without isolated nodes. Note that the network average clustering coefficient computation *excludes* isolated nodes with zero local clustering.

of 500 unique words are provided as default node features. Similarly, the papers' original PubMed IDs can be used to fetch relevant metadata.

2.2. Data Augmentation

For OGBN-arXiv, we used a July-2020 snapshot of the complete Microsoft Academic Graph (MAG) index (240M papers) - since MAG (and the associated API) was discontinued later - to obtain additional metadata (Zhang et al., 2022)[1]. Potential indicators of paper relatedness include: authors, venue, and fields of study. Fields of study, e.g. "computer science," "neural networks," etc. are automatically assigned with an associated confidence score (which we do not use), and each paper can have multiple fields of study, making them functionally similar to keywords. Other metadata (DOI, volume, page numbers, etc.) are not useful for our purposes.

For PubMed, an unprocessed version of the dataset preserving the original paper IDs was used (Namata et al., 2012)[2]. A January-2023 snapshot of the complete PubMed citation database (35M papers) was accessed for additional metadata. Potential indicators of paper relatedness include: authors, journal (indicated by unique NLM journal IDs), and Medical Subject Headings (MeSH®). The latter is an NLM-controlled hierarchical vocabulary used to characterize biomedical article content.

Given the features of interest, we define three additional edge types for each dataset:

- (Co)-Authorship: Two papers are connected if they share an author. This is based on the assumption that a given author tends to perform research on similar disciplines. Note that unlike MAG, PubMed Central does not provide unique identifiers for authors, so exact author names are used for PubMed, which can lead to some ambiguity in a minority of cases, e.g. two distinct authors with the same name.
- Source: Two papers are connected if they were published at the same venue (OGBN-arXiv), or in the same journal (PubMed), with the intuition that specific conferences and journals feature papers contributing to similar research areas.
- Subject Area: Two papers are connected if they share at least one field of study (OGBN-arXiv), or medical subject heading (PubMed).

Since the OGBN-arXiv Source and both datasets' Subject Area relationships result in massive edge lists, posing out-of-memory issues on the utilized hardware, we only create edges between up to k nodes per unique venue/field of study/MeSH, where k is the mean number of papers per venue/field of study/MeSH, in order to reduce the subgraphs' sizes.

In a traditional citation network, the edges are typically directed, but in our experiments, they are undirected to strengthen the connections of communities in the graph. The graph includes only one node type, "paper." Other approaches, notably

[1]The data is hosted by AMiner's **Open Academic Graph** project. All chunks were downloaded locally and metadata of IDs corresponding to papers in OGBN-arXiv were saved.

[2]This version of the dataset is hosted by the LINQS Statistical Relational Learning Group. The 2023 annual baseline on the NLM FTP server is accessed to retrieve metadata. All files were downloaded locally and metadata of matching IDs were extracted (19,716 records matched, 1 missing).

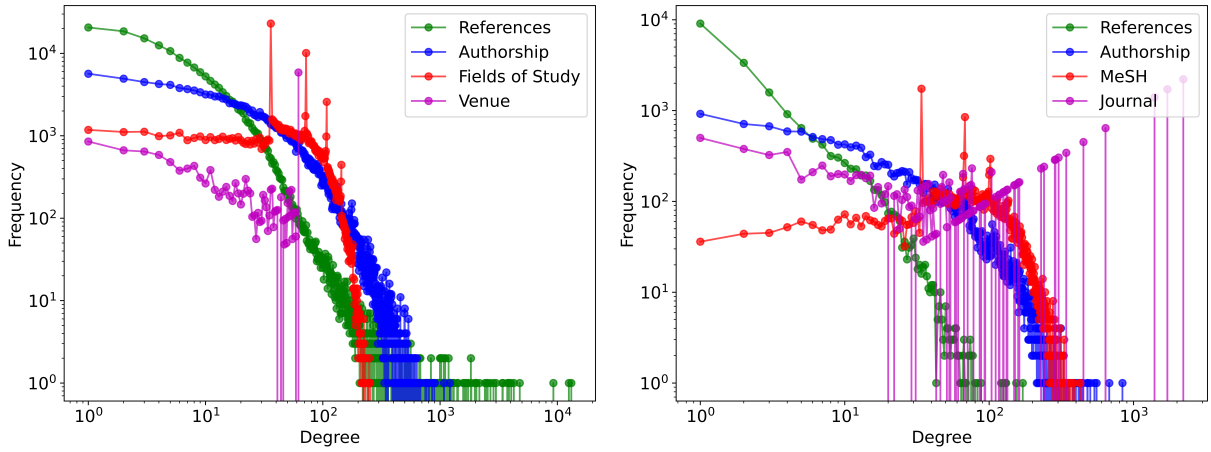

Figure 2: Degree distribution, i.e. frequency of each degree value, of all subgraphs for OGBN-arXiv (left) and PubMed (right), plotted on a log-log scale. Points indicate the unique degree values.

in the citation recommendation domain, leverage node types to represent authors and journals (Guo et al., 2017). However, this work strictly concerns relationships between papers and not between papers and other entities, in order to apply the homogeneous problem settings. Practically, the resultant graph would contain too many nodes, while the number of features and metadata is insufficient to generate informative representations of other node types, limiting their usefulness in the feature aggregation step. Hence, we specify our transformed graph as a multi-graph, i.e. possessing one node set, with distinct edges that are permitted to connect the same pair of nodes, and not a "true" heterogeneous graph.

For textual node feature representation, we leverage the recent SimTG and TAPE frameworks, which both utilize the raw textual features of datasets, in the form of concatenated titles and abstracts, and focus on fine-tuning pre-trained LMs and utilizing their last hidden states to infer node embeddings for training GNNs (Duan et al., 2023; He et al., 2024). These methods are present in the top OGBN-arXiv leaderboard submissions (at the time of writing). SimTG performs supervised parameter-efficient fine-tuning (PEFT) of an LM on the article classification task. Pre-computed embeddings for OGBN-arXiv are provided (using e5-large), which we utilize here. Since the authors do not report results on PubMed, we reproduce their methods to generate embeddings for this dataset, using a SciBERT model (Beltagy et al., 2019). TAPE proposes an LLM-to-LM interpreter, prompting GPT3.5 to perform zero-shot article classification and generate textual explanations for its decision-making process; the GPT3.5 predicted labels and explanations are then used to fine-tune a DeBERTa model. In this case, pre-computed embeddings were available for both datasets.

2.3. Subgraph Properties

Some insights on the characteristics of the defined subgraphs can be derived from Table 1. While the References graphs do not exhibit the tight clustering typical of real-world information networks, the strong signal of relatedness in the edges has nonetheless ensured their compatibility with message passing GNN paradigms (Wu et al., 2019b). This relatedness is also evident in the Authorship graphs, and the high level of clustering confirms the initial hypothesis that researchers co-author papers within similar topics. The Subject Area relationships, include many edges formed between shared generic keywords, e.g. "computer science," leading to rather average homophily. The Source subgraphs consist of isolated fully-connected clusters per unique source, with no inter-cluster connections, as each paper belongs to only one journal or venue. As with the Subject Area relationships, the research scope covered by a given publication conference or journal can be quite broad with respect to the paper labels.

Figure 2 shows the degree distribution of all edge type subgraphs in both datasets, which gives a clear view of the subgraphs' structures when interpreted with the above metrics. The high frequency of large node degrees in the PubMed Source subgraph corresponds to large journals; the size of the LCC (2,213) is the number of papers in the largest journal. While not visible for the OGBN-arXiv Source subgraph due to the aforementioned sampling in Section 2.2, a similar distribution would occur for large venues if all possible edges had been included. In contrast, the lower occurrence of high degree nodes and low clustering in the Refer-

Edge Types				OGBN-arXiv			PubMed		
Refs.	Auth.	Srce.	Subj.	Default	SimTG	TAPE	Default	SimTG	TAPE
✓	-	-	-	69.55 ± 0.31	74.07 ± 0.03	73.97 ± 0.01	87.72 ± 0.29	93.21 ± 0.24	93.04 ± 0.01
-	✓	-	-	61.51 ± 0.15	67.29 ± 0.23	67.50 ± 0.09	79.99 ± 0.29	82.93 ± 0.09	83.28 ± 0.03
-	-	✓	-	53.78 ± 0.26	73.30 ± 0.08	73.05 ± 0.02	48.99 ± 6.82*	61.97 ± 0.44	62.23 ± 1.37
-	-	-	✓	49.87 ± 0.11	55.54 ± 0.31	56.07 ± 0.32	73.57 ± 0.42	74.96 ± 1.90	75.78 ± 0.12
✓	✓	-	-	71.40 ± 0.19	75.80 ± 0.11	75.98 ± 0.06	88.91 ± 0.18	**93.62 ± 0.01**	**93.45 ± 0.17**
✓	-	✓	-	68.72 ± 0.30	76.05 ± 0.02	75.85 ± 0.10	87.97 ± 0.25	93.14 ± 0.05	93.05 ± 0.20
✓	-	-	✓	70.01 ± 0.05	74.42 ± 0.08	74.45 ± 0.05	88.14 ± 0.06	92.79 ± 0.29	93.30 ± 0.11
✓	✓	✓	-	70.97 ± 0.19	**77.26 ± 0.04**	**77.14 ± 0.10**	88.86 ± 0.10	93.54 ± 0.10	92.89 ± 0.24
✓	✓	-	✓	**71.81 ± 0.06**	75.94 ± 0.07	76.15 ± 0.12	**89.22 ± 0.15**	93.11 ± 0.32	93.23 ± 0.15
✓	-	✓	✓	69.30 ± 0.14	76.08 ± 0.05	75.97 ± 0.05	88.35 ± 0.17	92.84 ± 0.16	92.73 ± 0.50
✓	✓	✓	✓	71.30 ± 0.08	77.17 ± 0.02	77.07 ± 0.11	88.47 ± 0.28	93.16 ± 0.29	93.12 ± 0.16

Table 2: References, Authorship, Source (venue or journal), and Subject Area (fields of study or MeSH) subgraph ablation study for both datasets, 3-run average test accuracy with a 2-layer GCN and consistent hyperparameter values per dataset. The best results for each column are highlighted in bold. Asterisk indicates (significant) overfitting and instability.

ences subgraphs of both datasets indicates greater average distance across the LCC compared to the other subgraphs; such a structure stands to benefit the most from the multi-hop neighborhood feature aggregation performed by GNNs. Relative to the References, the Authorship and Subject Area subgraphs exhibit increased skewness in the distribution and higher average clustering, which indicates the presence of more (near-)cliques, i.e. subsections of the graph wherein (almost) any two papers share an author or topic. Hence, these subgraphs bear the closest structural resemblance to small-world networks (Watts and Strogatz, 1998). The impact of these degree distributions on classification performance is further investigated in Section 3.1.

3. Experiments and Results

We evaluate model performance on the task of *fully supervised transductive node classification*. The metric is multi-class accuracy on the test set. The proposed data preparation scheme is tested with several GNN architectures commonly deployed in benchmarks. We consider two GCN setups (base one and with a jumping knowledge module using max-pooling as the aggregation scheme), as well as GraphSAGE (Kipf and Welling, 2017; Xu et al., 2018; Hamilton et al., 2017). We also run experiments with the simplified graph convolutional operator (SGC) (Wu et al., 2019a). The increased graph footprint can lead to scalability concerns, hence the performance of such lightweight and parameter-efficient methods is of interest.

For `OGBN-arXiv`, the provided time-based split is used: train on papers published until 2017, validate on those published in 2018, test on those published since 2019. For `PubMed`, nodes of each class are randomly split into 60% - 20% - 20% for training - validation - and testing. Ablation experiments are also performed to examine the impact of the different edge types (averaged across 3 runs) and to identify the optimal edge type configuration for both datasets, on which we then report final results (averaged across 10 runs). Experiments were conducted on a `g4dn.2xlarge` EC2 instance (32 GB RAM, 1 NVIDIA Tesla T4 16 GB VRAM). Models are trained with negative log-likelihood loss, early stopping based on validation accuracy (patience of 20 epochs, with an upper limit of 500 epochs), and linear learning rate scheduling.

3.1. Ablation Study

Ablation results for both datasets are presented in Table 2, separated by node embedding method. First, all possible homogeneous subgraphs are inspected, as this is the conventional input data for this task (see the first 4 rows). The best performance is consistently achieved on the References graphs. Then we build upon the References graph by adding different combinations of other subgraphs. The results demonstrate that transitioning to multi-graphs can yield up to 3.19% performance improvement on `OGBN-arXiv` and 1.50% on `PubMed` (see differences between References-only and bold configurations). These results were obtained with a 2-layer GCN base, using an initial learning rate of 0.001 and hidden feature dimensionality of 128. For `PubMed`, we add an optimizer weight decay of 0.005.

Cross-checking with the metrics in Table 1 implies improvements from multi-graphs roughly cor-

Dataset	GNN	Default Graph			Multi-graph					
		Default	SimTG	TAPE	Default	Δ	SimTG	Δ	TAPE	Δ
OGBN-arXiv	GCN	69.67 ± 0.17	73.98 ± 0.11	74.08 ± 0.10	71.88 ± 0.06	+2.21%	77.30 ± 0.09	+3.32%	77.10 ± 0.10	+3.02%
	GCN+JK	70.24 ± 0.17	75.01 ± 0.15	75.15 ± 0.16	71.56 ± 0.21	+1.32%	77.05 ± 0.10	+2.04%	76.66 ± 0.10	+1.51%
	SAGE	68.99 ± 0.18	75.65 ± 0.11	75.41 ± 0.13	71.37 ± 0.21	+2.38%	77.39 ± 0.15	+1.74%	76.68 ± 0.06	+1.27%
	SGC	68.73 ± 0.14	73.95 ± 0.03	73.65 ± 0.25	70.24 ± 0.05	+1.51%	77.24 ± 0.01	+3.29%	75.93 ± 0.17	+2.28%
PubMed	GCN	87.67 ± 0.25	92.92 ± 0.12	92.92 ± 0.17	89.15 ± 0.14	+1.48%	93.49 ± 0.16	+0.57%	93.59 ± 0.26	+0.67%
	GCN+JK	87.13 ± 0.28	93.68 ± 0.18	93.49 ± 0.36	87.53 ± 0.62	+0.40%	94.11 ± 0.18	+0.43%	94.17 ± 0.13	+0.68%
	SAGE	88.30 ± 0.10	95.46 ± 0.07	94.87 ± 0.10	89.75 ± 0.09	+1.45%	95.51 ± 0.10	+0.05%	94.93 ± 0.13	+0.06%
	SGC	86.87 ± 0.16	90.31 ± 0.30	90.57 ± 0.32	86.56 ± 0.57	-0.31%	91.41 ± 0.13	+1.10%	91.20 ± 0.21	+0.63%

Table 3: Results with a variety of GNN backbones on the best multi-graph configuration per embedding method, based on the ablation study in Table 2, so e.g. the multi-graph for OGBN-arXiv GCN with SimTG embeddings consists of the References, Authorship, and Source graphs; the same multi-graph configuration is re-used for all other GNNs trained on OGBN-arXiv with SimTG embeddings. The baseline results on the default graph and the accuracy difference over the baseline are also displayed per embedding method. Green, gray, and red indicate increase, insignificant increase, and decrease, respectively.

respond to the edge homophily ratio of the utilized subgraphs, as strong homophily is implicitly assumed by the neighborhood aggregation mechanism of GCN-based models. Subsequently, their performance can be erratic and unpredictable in graphs with comparatively low homophily (Kipf and Welling, 2017; Ma et al., 2022). Since the R-GCN transformation collects neighborhoods from input subgraphs with equal weighting, including a comparatively noisy subgraph, e.g. PubMed Source, can worsen predictive performance. Changing the R-GCN aggregation operator, e.g. from mean to concatenation, does not alleviate this.

The Source subgraphs benefit substantially from the LM-based features, as the extent of feature aggregation is comparatively limited, due to the aforementioned tight clustering and isolation. Hence, the classifier relies more on the raw separability of the textual node features. This also explains the breakdown in performance when using the PubMed Source subgraph in a homogeneous setting, as a paper might possess only a few non-zero feature dimensions when using the default word vectors. The Subject Area subgraphs are more structurally preferable, but noisy edges (from keywords tied to concepts that are higher-level than the paper labels) reduce their usefulness in classification. In addition, the semantic information they encode is dominated by the dense LM-based features, reflected by the fact that they only appear in the optimal multi-graph configuration when using the default embeddings. Across all experimental settings, the Authorship subgraph enables consistent gains, and can outperform configurations that use more subgraphs. These trends are expected, given the characteristics discussed in Section 2.3.

3.2. Optimal Configuration

Results with the optimal configuration identified from the ablation study are listed in Table 3, for both datasets.

In most cases, preliminary experiments indicated that deeper (3 or more layers) networks either worsen or do not benefit performance of the tested models in multi-graph configurations (however, note that tested single-layer models underfit and thus do not improve performance). Likely, the additional feature averaging step from the R-GCN transformation increases the risk of oversmoothing even on shallow networks. These hypothesized effects are more pronounced when using graphs with high average degree, e.g. the Source and Subject Area subgraphs; nodes with high degree aggregate more information from their neighbors, increasing the likelihood of homogenization as network depth increases (Chen et al., 2020).

The results demonstrate that the additional structural information provided by multi-graphs generally improves final performance of a variety of hetero-transformed GNN frameworks compared to their homogeneous counterparts on both datasets, with more pronounced effects on OGBN-arXiv, when making optimal subgraph choices (though, suboptimal choices can still situationally improve performance). These improvements are independent of the tested textual embedding methods, and can occur even when the added subgraphs possess suboptimal graph properties, e.g. lower edge homophily ratio and presence of isolated nodes, compared to the starting References graph. Notably, the best results are competitive with the SOTA, while operating on a limited compute budget and low level of complexity (simple 2-layer GNN model pipelines with comparatively few trainable parameters). On OGBN-arXiv, we can achieve a top-5

result (at the time of writing) deploying our multi-graph with a GraphSAGE backbone and SimTG embeddings.

4. Conclusions and Future Work

In this paper, we propose a data transformation methodology leveraging metadata retrieved from citation databases to create enriched multi-graph representations based on various additional signals of document relatedness: co-authorship, publication source, fields of study, and subject headings. We also test the substitution of default node features with LM-based embeddings to capture higher-dimensionality textual semantics. By nature, the methodology is GNN- and embedding-agnostic. Deploying optimal configurations of the transformed multi-graph with a variety of simple GNN pipelines leads to consistent improvements over the starting graph, and enables results on par with the SOTA in full-supervised node classification. Overall, results show that our methodology can be an effective strategy to achieve respectable performance on datasets with readily-available article metadata, without necessitating complex GNN architectures and lengthy (pre-)training procedures.

As the methodology is compatible with any hetero-transformable GNN backbone and textual node embedding technique, we expect that deploying the transformed data with SOTA GNN frameworks, e.g. RevGAT by Li et al. (2021) on `OGBN-arXiv`, will lead to greater raw performance. Though, the larger memory footprint of the graph may complicate the application of such frameworks.

Refining the edge type definitions, e.g. connect papers that share at least two fields of study and/or remove "generic" fields applicable to a majority of papers in the set, can help de-noising and improving the properties of the respective subgraphs. A custom aggregation scheme could be implemented for the heterogeneous transformation dependent on individual subgraph properties, such as a weighted average based on some metric of subgraph "quality," e.g. homophily. To mitigate the increased risk of oversmoothing induced by multi-graphs and stabilize convergence behavior, additional regularization techniques, e.g. DropEdge by Rong et al. (2020), could be considered.

5. References

Albert-László Barabási and Márton Pósfai. 2017. *Network science: graph theory*, page 17–29. Cambridge University Press.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Rui Bing, Guan Yuan, Mu Zhu, Fanrong Meng, Huifang Ma, and Shaojie Qiao. 2022. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artificial Intelligence Review*, pages 1–40.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1725–1735. PMLR.

Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. 2023. Simteg: A frustratingly simple approach improves textual graph learning.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. 2020. Sign: Scalable inception graph neural networks.

Lantian Guo, Xiaoyan Cai, Fei Hao, Dejun Mu, Changjian Fang, and Libin Yang. 2017. Exploiting fine-grained co-authorship for personalized citation recommendation. *IEEE Access*, 5:12714–12725.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

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*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *CoRR*, abs/2005.00687.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R. Benson. 2021. Combining label propagation and simple models outperforms graph neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. 2021. Training graph neural networks with 1000 layers. *CoRR*, abs/2106.07476.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2022. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations*.

Marcin Michał Mirończuk and Jarosław Protasiewicz. 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, 106:36–54.

Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. 2012. Query-driven active surveying for collective classification. In *International Workshop on Mining and Learning with Graphs*, Edinburgh, Scotland.

Jason Priem, Heather Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts.

Lucas Prieto, Jeroen Den Boef, Paul Groth, and Joran Cornelisse. 2023. Parameter efficient node classification on homophilic graphs. *Transactions on Machine Learning Research*.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*.

Thomas Schank and Dorothea Wagner. 2005. Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine*, 29(3):93.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*. Accepted as poster.

Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Felix Wu, Tianyi Zhang, Amauri H. Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019a. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019b. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. *CoRR*, abs/1806.03536.

Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. 2022. Simple and efficient heterogeneous graph neural network.

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Evgeny Kharlamov, Bin Shao, Rui Li, and Kuansan Wang. 2022. Oag: Linking entities across large-scale heterogeneous knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14.

A. Reproducibility Statement

For reproducibility, our implementation is available at this GitHub repository, including the required external metadata, node feature embeddings, and other misc. information.

B. Supplementary Results

The validation accuracy and number of trainable GNN parameters for all results in Table 3 can be seen in Table 4. Note that the feature dimensionality for SimTG differs between datasets - 1024-dim. (`e5-large`) for `OGBN-arXiv`, 768-dim. (`scibert-scivocab-uncased`) for `PubMed`.

Dataset	GNN	Metric	Default Graph			Multi-graph		
			Default	SimTG	TAPE	Default	SimTG	TAPE
OGBN-arXiv	GCN	Val. Acc.	70.78 ± 0.15	75.51 ± 0.15	75.41 ± 0.06	73.15 ± 0.12	78.39 ± 0.16	78.07 ± 0.08
		# Params	21,928	136,616	103,848	65,272	409,336	311,032
	GCN+JK	Val. Acc.	71.29 ± 0.09	76.20 ± 0.12	76.10 ± 0.11	73.13 ± 0.13	78.38 ± 0.12	77.78 ± 0.07
		# Params	38,686	153,384	120,616	104,744	448,808	350,504
	SAGE	Val. Acc.	70.16 ± 0.25	77.01 ± 0.15	76.27 ± 0.07	72.90 ± 0.10	78.69 ± 0.15	77.63 ± 0.04
		# Params	43,432	272,808	207,272	129,784	817,912	621,304
	SGC	Val. Acc.	69.82 ± 0.12	75.04 ± 0.00	74.80 ± 0.16	71.59 ± 0.06	78.39 ± 0.00	77.11 ± 0.09
		# Params	5,160	41,000	30,760	15,480	123,000	92,280
PubMed	GCN	Val. Acc.	88.17 ± 0.16	93.67 ± 0.28	93.81 ± 0.09	89.39 ± 0.16	94.52 ± 0.06	94.63 ± 0.12
		# Params	64,771	99,075	99,075	193,801	197,894	197,894
	GCN+JK	Val. Acc.	87.93 ± 0.18	94.77 ± 0.15	94.56 ± 0.27	88.74 ± 0.12	95.30 ± 0.23	95.48 ± 0.21
		# Params	81,539	115,843	115,843	242,819	230,787	230,787
	SAGE	Val. Acc.	89.15 ± 0.24	96.36 ± 0.17	96.27 ± 0.16	90.64 ± 0.13	96.57 ± 0.13	96.59 ± 0.05
		# Params	129,155	197,763	197,763	386,953	395,270	395,270
	SGC	Val. Acc.	87.05 ± 0.12	91.00 ± 0.17	91.59 ± 0.17	86.72 ± 0.39	92.31 ± 0.06	92.49 ± 0.17
		# Params	1,503	2,307	2,307	4,509	4,614	4,614

Table 4: Validation accuracy and number of trainable GNN parameters for all results in Table 3.

Dataset	Hyperparameter	GCN	GCN+JK	SAGE	SGC
OGBN-arXiv	# Layers	2	2	2	2
	Hidden Channels	128	128	128	-
	Dropout	0	0	0	-
	Init. Learning Rate	0.001	0.001	0.001	0.1*
	Weight Decay	0	0	0	0
PubMed	# Layers	2	2	2	2
	Hidden Channels	128	128	128	-
	Dropout	0	0.2**	0	-
	Init. Learning Rate	0.001	0.001	0.001	0.1
	Weight Decay	0.005	0***	0.005	0

Table 5: Hyperparameters used for all results in Table 3. *0.01 for TAPE embeddings on multi-graph. **0.5 for default embeddings on multi-graph. ***0.001 for default embeddings on multi-graph.

C. Hyperparameters

The hyperparameters used for all results in Table 3 can be seen in Table 5. Note that these parameters were not comprehensively tuned per (dataset, GNN, graph type) combination, to illustrate the generality of our methods; a more extensive hyperparameter search can yield better results. Also, expanding the ablation study in Table 2 for specific GNN backbones rather than generalizing the GCN-applicable optimal configuration can further optimize results.