TeG-DRec: Inductive Text-Graph Learning for Unseen Node Scientific Dataset Recommendation

Ammar Qayyum[1], Bassamtiano R. Irnawan[1], Fumiyo Fukumoto[2], Latifah M. Kamarudin[3], Kentaro Go[2], Yoshimi Suzuki[2]
[1]Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences
[2]Graduate Faculty of Interdisciplinary Research
University of Yamanashi
[3]Centre of Excellence for Advanced Sensor Technology Universiti Malaysia Perlis
{g25dtsa4,g23dtsa2,fukumoto,go,ysuzuki}@yamanashi.ac.jp
latifahmunirah@unimap.edu.my

Abstract

Scientific datasets are crucial for evaluating scientific research, and their number is increasing rapidly. Most scientific dataset recommendation systems use Information Retrieval (IR) methods that model semantics while overlooking interactions. Graph Neural Networks (GNNs) excel at handling interactions between entities but often overlook textual content, limiting their ability to generalise to unseen nodes. We propose **TeG-DRec**, a framework for scientific dataset recommendation that integrates GNNs and textual content via a subgraph generation module to ensure correct propagation throughout the model, enabling handling of unseen data. Experimental results on the dataset recommendation's dataset show that our method outperformed the baselines for text-based IR and graph-based recommendation systems. Our source code is available at https://github.com/Maqif14/TeG-DRec.git

1 Introduction

Scientific datasets are essential for evaluating scientific research, as it is crucial to examine and verify their behaviour to achieve optimal performance in real-world scenarios (Özgöbek et al., 2014; Fahrudin and Wijaya, 2024). When a dataset is tailored to the specific context of the learning environment, it can significantly improve system performance (Verbert et al., 2011). For example, the Common Crawl dataset significantly contributed to the efficacy of GPT-3 as a formidable Large Language Model (LLM) upon its release in 2020 (Brown et al., 2020). The number of datasets increases annually by hundreds each year (Viswanathan et al., 2023). The growing number of datasets complicates manual search for the optimal dataset, occasionally leading to poor selections (Patankar et al., 2023; Viswanathan et al., 2023; Qin et al., 2024). Consequently, the need for a dataset recommender is greater than ever to enhance research efficiency.

Several studies have explored scientific dataset recommendation systems using text-based IR methods (Wang et al., 2021; Färber and Leisinger, 2021; Keller and Munz, 2022; Yadav et al., 2023; Zhang and Ashraf, 2023), with some extending it using neural bi-encoders to capture richer contextual semantics (Viswanathan et al., 2023). These approaches typically compute lexical or embedding-based similarity between query descriptions and candidate datasets. Despite their scalability, there is no direct interaction between the query and the document, as they are encoded independently during embedding generation, resulting in a loss of structural relationships among them (Humeau et al., 2019; Tran et al., 2024).

Recent advances in GNNs on the scientific dataset recommendation task offer a promising approach to solve the issue (Altaf et al., 2019; Qayyum et al., 2025). However, these methods generally lack inductive capability, which is essential for handling unseen nodes. Such inductive ability is crucial in scientific dataset recommendation, where the number of papers and datasets continues to grow rapidly. Aside from that, most GNNs-based approaches tend to overlook the rich textual content associated with these nodes, resulting in incomplete semantic representations.

Several attempts have been made to address the unseen node using an inductive GNNs approach in the field of recommendation systems (Teru et al., 2020; Xiao et al., 2023), with the ability to categorise labels that did not exist during training. Inductive GNNs have not yet been applied to the scientific dataset recommendation task, although we believe that leveraging them could offer significant benefits.

To address this challenge, we propose **TeG-DRec** (**Te**xtual **G**raph **D**ataset **Rec**ommendation), a framework that integrates textual content with inductive GNNs. **TeG-DRec** is designed to handle realistic scenarios where new scientific papers or

Proceedings of the Third Workshop for Artificial Intelligence for Scientific Publications, pages 103–114

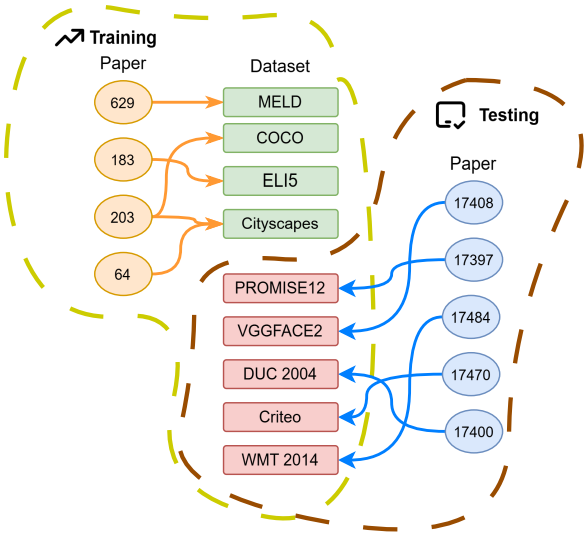

Figure 1: Research problem where the scientific papers in testing did not appear in the training set (unseen paper nodes in blue colour) connected with the label dataset that does not have any existing link with scientific papers during training (unseen dataset nodes in red colour)

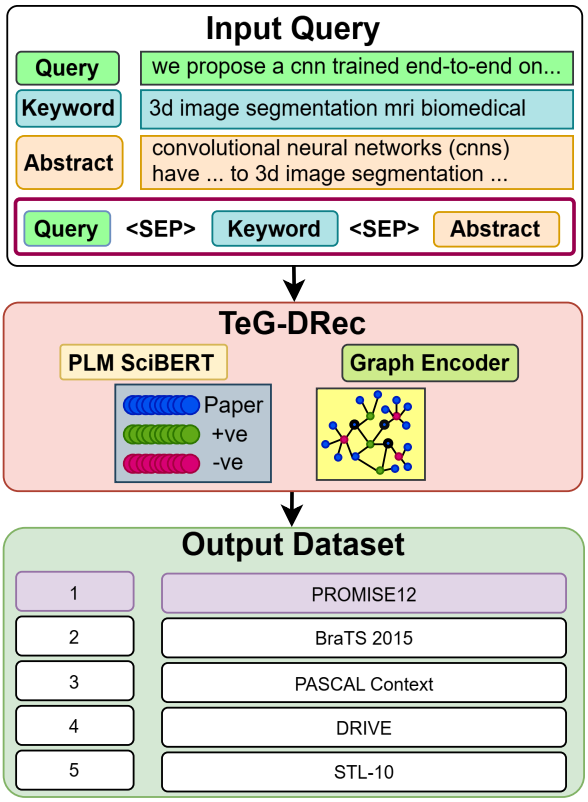

Figure 2: Overview of the scientific dataset recommendation process in **TeG-DRec**, from the input to output

scientific datasets are continuously introduced without explicit links to existing entities. As illustrated in Figure 1, unseen nodes refer to the nodes that do not appear during training or nodes that do not have any connection with any nodes during training.

Aside from that, TeG-DRec introduces a subgraph generation module that jointly enables inductive learning, contrastive learning, and margin-based optimisation within a cohesive training process. By combining the strengths of both textual and structural modalities, **TeG-DRec** effectively captures semantic and relational dependencies, offering robust inductive generalisation and improved scientific dataset recommendation performance.

The recommendation system works by taking a set of input queries, including the query, keyword query, and abstract. These inputs are then passed to **TeG-DRec** for the recommendation process, where the model predicts and outputs the Top-K datasets that best match the given inputs. This process is illustrated in Figure 2. The dataset used in our experiment consists of two node types: scientific papers and datasets, where the datasets serve as the target items to be recommended for each paper.

We compare **TeG-DRec** with text-based IR methods and a graph-based baseline. The text-based IR methods follow a neural bi-encoder framework (Ma et al., 2025), leveraging recent embedding models, which include SciBERT (Beltagy et al., 2019), Contriever (Lei et al., 2023), BGE-M3 (Chen et al., 2024), and E5 (Wang et al., 2024) that provide strong semantic representations for scientific and general-domain retrieval tasks. For graph-based baselines, we compare **TeG-DRec** against GraphSAGE (Hamilton et al., 2017), Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2017) and Graph Attention Networks (GAT) (Veličković et al., 2018), which rely on the structural relations in the graph without incorporating the textual components of **TeG-DRec**. **TeG-DRec** consistently outperforms these baselines across all evaluation metrics, demonstrating its ability to capture both semantic and structural relationships effectively. In summary, this work makes three key contributions:

1. We propose **TeG-DRec**, a framework for scientific dataset recommendation that supports inductive recommendation for newly published scientific papers and scientific datasets, effectively handling unseen nodes without retraining.

2. We introduce a unified framework that integrates inductive GNNs with textual representations to jointly capture structural and semantic information.

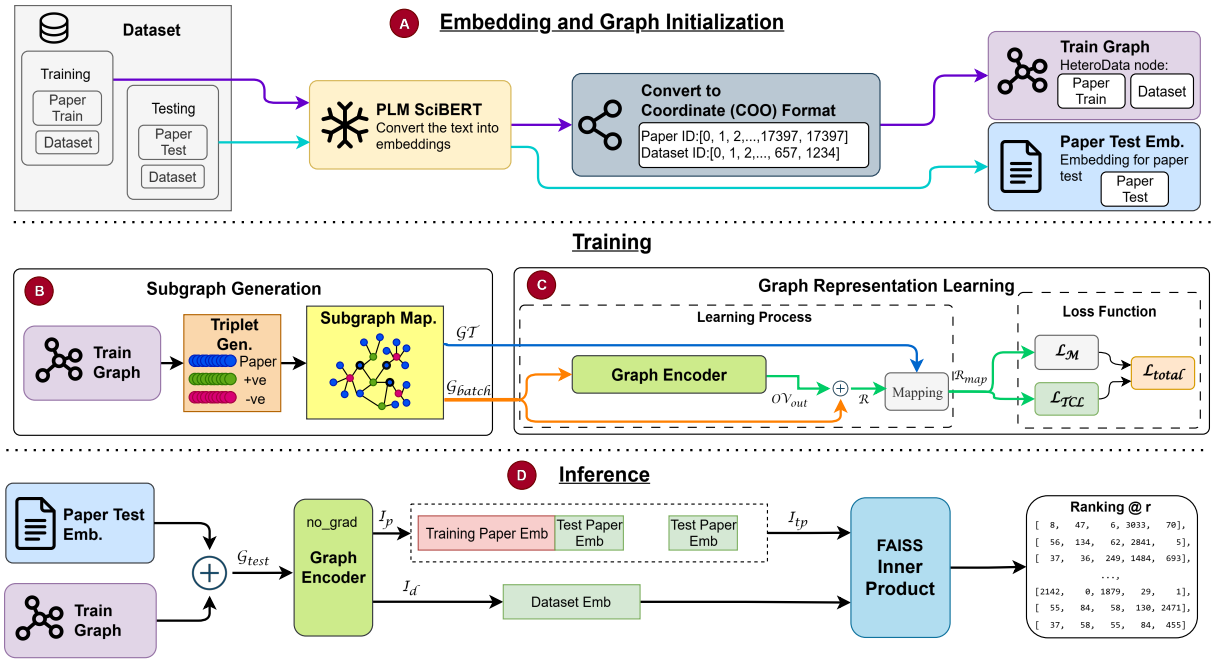

Figure 3: Overview of **TeG-DRec**, which consists of four main modules: (A) Embedding and Graph Initialisation, (B) Subgraph Generation, (C) Graph Representation Learning, and (D) Inference

3. We conduct extensive experiments on a publicly available benchmark and demonstrate that **TeG-DRec** consistently outperforms strong text-based IR and graph-based baselines.

2 Related Work

The techniques used to create the scientific dataset recommendation problem can be categorised into two groups: text-based IR and graph-based methods.

Text-Based IR Text-based IR for scientific dataset recommendation can be categorised into traditional methods and neural bi-encoder methods. The traditional method comprises BM25 (Keller and Munz, 2022), which ranks the dataset based on term-frequency matching, and a SciBERT-based text classification model (Beltagy et al., 2019; Färber and Leisinger, 2021). More recently, the Neural Bi-Encoder method proposed by Viswanathan et al. (2023) adopts a neural bi-encoder with SciBERT embeddings to encode both scientific papers and datasets. However, their model encodes scientific papers and datasets separately, which ignores their structural relationships.

Graph-Based Method The graph-based method can leverage the structural relationships between scientific papers and datasets. These structural relationships refer to the connections between scientific papers and the datasets they use, or they can be citation network among papers, datasets, and other related papers. Altaf et al. (2019) proposed a heterogeneous variational graph autoencoder (HVGAE) that integrates a citation network with paper–dataset associations to generate more informative representations for recommendation. Similarly, Qayyum et al. (2025) utilised GNNs enriched with textual features to recommend relevant datasets. However, their method can only handle transductive graphs, which require the nodes to be present during training. This limits the usage of the model in real-world situations where the nodes are constantly added. To address this limitation, several inductive graph learning frameworks have been proposed for recommendation systems, including GraphSAGE (Hamilton et al., 2017) and Graph Attention Networks (GAT) (Veličković et al., 2018), which enable improved generalisation via neighbourhood aggregation mechanisms. Additionally, Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2017), although not inherently inductive, offer a promising solution by effectively handling multiple edge types within a graph.

3 TeG-DRec Framework

3.1 Overview

TeG-DRec (**Tex**tual **G**raph-**D**ataset **Rec**ommendation) integrates textual content with graph structures, enabling the model to generalise toward unseen nodes by leveraging both the semantic properties and the structural information of the nodes. To achieve this, four main modules have been designed to address the specific requirements: (A) Embedding and Initialisation, (B) Subgraph Generation, (C) Graph Representation Learning, and (D) Inference, as depicted in Figure 3. In particular, the (B) Subgraph Generation module ensures that the textual content and graph structures are correctly aligned and passed through the inductive graph and loss components.

3.2 Embedding and Graph Initialisation

Embedding and Graph Initialisation module is responsible for encoding the textual information into embeddings and constructing the corresponding graph connections. This module ensures that the rich textual content is effectively integrated into the graph structure.

The dataset used in our experiments comprises descriptions of scientific papers, datasets, and the associations indicating which datasets are used by each paper. This relationship refers to the connection between the scientific papers, their corresponding positive datasets, and their corresponding negative datasets. A positive dataset refers to the dataset actually used by a given scientific paper, while a negative dataset represents a hard negative sample that is not used by the scientific paper. This is further illustrated in Figure 4.

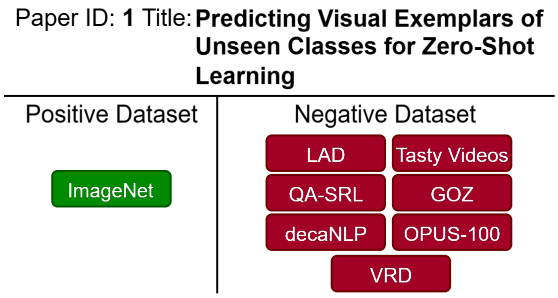

Figure 4: Example of positive and negative dataset sample for a paper with ID number 1

To remove unnecessary symbols and particular words from the dataset descriptions, a cleaning process is applied prior to using SciBERT to produce dense vector embeddings. Conversely, the relationships between scientific papers and datasets are represented in Coordinate Format (COO) as sparse matrices.

The COO maps of the scientific paper ID p with its associated positive dataset ID d^+ and negative dataset ID d^- as illustrated in Figure 3. The dense vector embeddings of the description and the COO of the scientific paper and dataset are subsequently passed on to the HeteroData G class in Pytorch Geometric (PyG) (Fey and Lenssen, 2019). The HeteroData G class utilises dense vector embedding $\mathcal{V}$ and COO format $\mathcal{E}$ to generate a train data graph. Meanwhile, the test paper dense vector embeddings $\mathcal{P}_{test}$ are extracted to be used later in the Inference section.

3.3 Subgraph Generation

The subgraph generation module enables the model to handle subgraphs rather than the entire graph, ensuring computational efficiency when learning on a large-scale graph. Additionally, it guarantees that graph nodes are properly aligned with their corresponding textual features before being passed to the Graph Representation Learning module. Proper alignment is essential, as misalignment would disrupt feature aggregation across connected nodes, thereby hindering inductive learning. Additionally, misalignment could also result in incorrect node pairings during loss computation. Figure 5 illustrates the flow of node IDs within this module. Here, **P** (blue) denotes the IDs of scientific papers, while **+** (green) and **–** (red) represent the positive and negative datasets associated with the scientific papers, as described previously in Figure 4.

This module consists of two subcomponents: a triplet generation process that constructs triplets from the train graph, and a subgraph mapping procedure that extracts subgraphs from the training graph and maps them according to the global node IDs.

Triplet Generation Triplet generation is used to efficiently load and manage the triplet set $\mathcal{T}$, which consists of scientific paper ID p, positive dataset ID d^+, and negative dataset ID d^-, from the train graph G. The triplet set $\mathcal{T}$ is then shuffled and partitioned into batches $\mathcal{T}_b$ as outlined in Algorithm 1. Subsequently, these batches are fed into the subgraph sampling module for subgraph mapping.

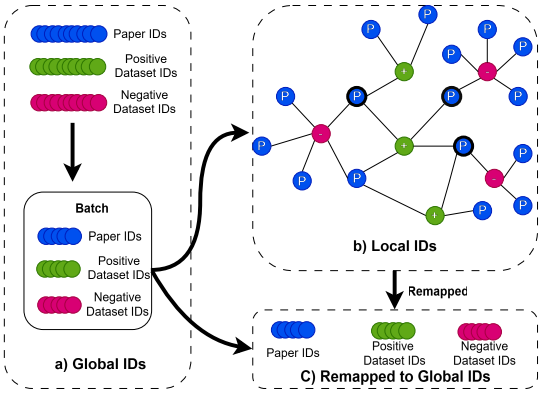

Figure 5: Flow inside Subgraph Module shows that after the subgraph extraction, the node IDs become local IDs. This IDs is remapped back into global IDs to keep track of the IDs

Algorithm 1 Triplet Generation and Subgraph Mapping

Require:

Dense Vector Embeddings $\mathcal{V}$
COO Format $\mathcal{E}$
Heterogeneous Train Graph $G = (\mathcal{V}, \mathcal{E})$
Scientific Paper ID p
Positive Dataset ID d^+
Negative Dataset ID d^-
Set of Triplets $\mathcal{T} = \{(p, d^+, d^-)\}$
Batch Size B
Number of Neighbours k

Ensure:

Mini-batch Subgraph $\mathcal{G}_{batch}$
Mapped Triplet Indices:
$\mathcal{GT} = \{(P_{local}, D^+_{local}, D^-_{local})\}$

Triplet Generation:

Dataset $\leftarrow$TripletGeneration($\mathcal{T}$)
$\mathcal{T}_b \leftarrow$ Shuffle(Dataset)
return $\mathcal{T}_b$

Subgraph Mapping:

for b in $\mathcal{T}_b$ **do**
    $p_{batch} \leftarrow \{p_i\}_{i=1}^{b}$
    $\mathcal{G}_{batch} \leftarrow$ `NeighborLoader`(G, p_{batch}, k)
    $\mathcal{GT} \leftarrow$ `GlobaltoLocal`$(\mathcal{G}_{\text{batch}}, b)$
end for
return $\mathcal{G}_{batch}, \mathcal{GT}$

Subgraph Mapping A batch b is sampled from $\mathcal{T}_b$ and used to generate a subgraph. The scientific paper IDs p from the batch b serve as input nodes p_{batch} for the `NeighborLoader()` from PyG. We perform two-hop subgraph sampling, where the first hop samples twenty neighbours and the second hop samples fifteen. The resulting subgraph $\mathcal{G}_{batch}$ is then remapped to global indices $\mathcal{T}_b$ using the `GlobaltoLocal()` function, producing a triplet local $\mathcal{GT}$.

This remapping ensures that node identities remain consistent, as subgraph construction replaces global indices with local ones. This step ensures that the loss function receives the correct node IDs with its embeddings. The whole procedure is shown in Algorithm 1.

3.4 Graph Representation Learning

Graph Representation Learning module enables **TeG-DRec** to handle unseen nodes (refer Figure 1) as it comprises two main subcomponents: graph encoder and loss functions. The graph encoder uses an inductive GNNs to learn representations from the subgraph, which helps it to generalise towards unseen nodes.

Aside from that, the loss function computes ranking and contrastive losses and uses gradient-based optimisation during training via backpropagation. Ranking losses aim to prioritise positive pairs over negative ones, ensuring that relevant datasets are ranked higher than irrelevant ones. In contrast, contrastive loss enhances representation learning by aligning similar views in the embedding space while separating dissimilar ones, improving the model's ability to distinguish between different data points.

Graph Encoder The graph encoder processes the input graph $\mathcal{G}_{\text{batch}}$ which represents the IDs of scientific papers, positive datasets, and negative datasets (see Figure 5), along with their corresponding embeddings, using an inductive GNN to generate the output views OV_{out}. These output views are then concatenated with the original pre-encoded representations of $\mathcal{G}_{\text{batch}}$ to form the final recommendation embeddings R. The recommendation embeddings R are mapped based on the local triplet mapping $\mathcal{GT}$, resulting in the mapped recommendation embeddings R_{map}, which are then passed to the loss functions. The graph encoder is implemented as a modular component, allowing it to operate with various types of inductive GNNs. The detailed procedure is outlined in Algorithm 2.

In this study, we incorporate multiple inductive GNNs encoders, including GraphSAGE (Hamilton et al., 2017), R-GCN (Schlichtkrull et al., 2017) and GAT (Veličković et al., 2018).

Loss Functions Our model is optimised using two main types of loss functions: ranking loss and

Algorithm 2 Graph Representation Learning

Require:

Mini-batch Subgraph $\mathcal{G}_{batch}$

Mapped Triplet Local:

$\mathcal{GT} = \{(P_{local}, D^{+}_{local}, D^{-}_{local})\}$

Ensure:

Rec Embeddings Mapped $\mathcal{R}_{map}$

Learning Process:

$OV_{\text{out}} \leftarrow \text{GraphEncoder}(\mathcal{G}_{\text{batch}})$

$\mathcal{R} \leftarrow OV_{\text{out}} \circ \mathcal{G}_{\text{batch}}$

$\mathcal{R}_{map} \leftarrow R$ such that $R \subseteq \mathcal{GT}$

return $\mathcal{R}_{map}$

contrastive loss. For ranking loss, we use margin ranking loss $\mathcal{L}_M$, which enforces a margin between positive and negative scores to maximise the difference between them. The loss function is defined in Eq. (1).

$$\mathcal{L}_M = \max(0, -y * (x_1 - x_2) + \text{ margin }), \quad (1)$$

where x_1 denotes the positive sample and x_2 denotes the negative sample, while y is a binary label. In our experiments, we set $y = 1$ to enforce that the positive sample x_1 should always be ranked higher than the negative sample x_2.

For contrastive loss, it is applied between the text embeddings of scientific papers and their corresponding positive datasets. The objective is to encourage the model to draw semantically aligned paper–dataset pairs closer in the embedding space, while pushing apart unrelated pairs. It is given by:

$$\mathcal{L}_{TCL} = \text{InfoNCE}(\mathbf{Z}_p, \mathbf{Z}_d, \tau), \quad (2)$$

where $\mathcal{L}_{TCL}$ refer to text contrastive loss, $\mathbf{Z}_p$ indicates the paper embeddings and $\mathbf{Z}_d$ is the positive dataset embeddings. τ is a temperature, which is a constant. This contrastive loss is formulated using the InfoNCE loss (Rusak et al., 2024), as defined in Eq. (3).

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\frac{\mathbf{z}_i^{(1)} \cdot \mathbf{z}_i^{(2)}}{\tau}\right)}{\sum_{j=1}^{N} \exp\left(\frac{\mathbf{z}_i^{(1)} \cdot \mathbf{z}_j^{(2)}}{\tau}\right)}, \quad (3)$$

where $\mathbf{z}_i^{(1)}$ is the original view of sample i, $\mathbf{z}_i^{(2)}$ is the augmented view of i, and τ is the temperature constant. $\mathbf{z}_j^{(2)}$ refers to the positive sample from the augmented view.

All losses are multiplied by their respective ratios for balanced performance, then combined with a regularisation loss to avoid overfitting. The equation of the batch loss is defined as Eq. (4).

$$\mathcal{L}_{total} = \mathcal{L}_{TCL} + \mathcal{L}_M + \mathcal{L}_{L2reg}, \quad (4)$$

where $\mathcal{L}_{SCL}$ is the structure contrastive loss, $\mathcal{L}_{TCL}$ indicates the text contrastive loss, $\mathcal{L}_M$ refers to the margin loss and $\mathcal{L}_{L2reg}$ shows the L2 regularization loss.

3.5 Inference

The inference module enables **TeG-DRec** to evaluate scenarios involving truly unseen nodes. Evaluating such scenarios is essential for simulating real-world conditions, as new scientific papers and datasets continue to appear. To achieve that, we concatenate the test paper dense vector embeddings $\mathcal{P}_{test}$ with the heterogeneous train graph dense vector embeddings $G(\mathcal{V})$, producing the test graph G_{test} as shown in Algorithm 3. The dense vector embeddings of test papers $\mathcal{P}_{test}$ have no connections to any dataset nodes within the test graph G_{test}. This ensures that the encoder processes test nodes independently of the training structure. The test graph is passed to the GraphEncoder$_{nograd}$ for the encoding process. After obtaining paper I_p and dataset I_d embeddings from the model, we extract test paper embeddings I_{tp} by indexing the unique nodes of test paper dense vector embeddings index $\mathcal{P}_{test}$.

To generate recommendations, we compute the maximum inner product between each test paper embedding I_{tp} and dataset embeddings I_d using FAISS (Douze et al., 2024) and retrieve the top-r results. The overall inference process is illustrated in Figure 3.

4 Experiments

4.1 Experimental Setup

Dataset We use the DataFinder Dataset (Viswanathan et al., 2023) to train and evaluate our model. The dataset is available on GitHub[1]. This dataset contains metadata about scientific papers and their associated datasets. It is pre-split into training and test sets. The training data includes true positive and hard negative dataset pairs for each publication, sourced from the Papers with

[1] https://github.com/viswavi/datafinder/tree/main

Algorithm 3 Inference

Require:

Dense Vector Embeddings $\mathcal{V}$

COO Format $\mathcal{E}$

Heterogeneous Train Graph $G = (\mathcal{V}, \mathcal{E})$

Test Scientific Papers $\mathcal{P}_{test} = \mathcal{V}$

Ensure:

Top-r results top-r

Inference Process:

$\mathcal{G}_{test} \leftarrow G(\mathcal{V}) \circ \mathcal{P}_{test}$

$I_p, I_d \leftarrow \text{GraphEncoder}_{nograd}(\mathcal{G}_{test})$

$I_{tp} \leftarrow I_p[\text{Unique}(\mathcal{P}_{test})]$

top-$r \leftarrow \text{FAISSInnerProduct}(I_{tp}, I_d)$

return top-r

Code[2] website. The hard negative datasets are selected using BM25. These hard negatives do not necessarily overlap with true positives. The test data consists of expert-annotated evaluations from SciREX (Jain et al., 2020).

To ensure that the test data align with our truly unseen node scenario, we remove scientific papers that interact with positive datasets. The remaining connected datasets in the test data are then removed from the hard negative datasets in the train data. This is to ensure that the test data are truly unseen. Table 1 summarises the statistics of the dataset. Appendix A outlines the available features within the dataset.

Data	Train	Test
# of scientific papers	17,397	88
# of positive datasets	461	74
# of positive interactions	20,789	126
# of negative datasets	2,570	–
# of negative interactions	118,997	–

Table 1: The statistics of scientific papers and datasets in Datafinder Dataset

Evaluation metrics We evaluate our method using five standard recommender system metrics: Precision **(P)**, Recall **(R)**, Normalised Discounted Cumulative Gain **(NDCG)**, Mean Average Precision **(MAP)**, and Mean Reciprocal Rank **(MRR)**. For top-r, we set $r = 5$, reflecting real-world usage where users engage with the highest-ranked suggestions. This is particularly relevant for Precision, Recall, and NDCG, all of which involve the top-r metric in their calculation.

Implementation To ensure separation between node features, a unique token is added before each feature during encoding. For training stability and convergence, we implemented a learning rate scheduler that combines linear warmup with cosine annealing. The implementation was done using PyTorch and PyG (Fey and Lenssen, 2019), with experiments run on an NVIDIA RTX 6000 Ada GPU with 48GB VRAM. The hyperparameters used in this experiment are detailed in Appendix B to facilitate reproducibility.

4.2 Baselines

To evaluate the effectiveness of our proposed method, we compare it against seven baseline approaches, which are classified into two groups:

Text-Based IR Method consists of four baselines, each of which utilises the neural biencoder framework by Ma et al. (2025) with four different embedding models, including:

1. **SciBERT** (Beltagy et al., 2019) is a pretrained BERT-based language model specifically designed for scientific and scholarly text.
2. **Contriever** (Lei et al., 2023) is an unsupervised dense information retrieval model that leverages contrastive learning to train a bi-encoder that maps queries and documents to a shared embedding space.
3. **BGE-M3** (Chen et al., 2024) is a multilingual embedding model designed to handle various retrieval tasks efficiently.
4. **E5-Large** (Wang et al., 2024) is a text embedding that is trained using weakly-supervised contrastive learning on a large-scale dataset of text pairs.

Graph-Based Method consist of three baselines:

1. **GraphSAGE** (Hamilton et al., 2017) GraphSAGE is an inductive graph whose primary goal is to learn node embeddings that generalise towards unseen nodes, rather than only represented nodes seen during training.
2. **Relational Graph Convolutional Networks (R-GCN)** (Schlichtkrull et al., 2017) is an extension of Graph Convolutional Networks (GCNs), which is designed to handle graphs where edges have types or relations.

[2] https://huggingface.co/papers/trending

Methods	Datafinder Dataset (Unseen Configuration)				
	P@5	R@5	NDCG@5	MAP	MRR
Text-based IR Method					
Neural Biencoder (SciBERT)	0.015	0.053	0.032	0.023	0.029
Neural Biencoder (Contriever)	0.018	0.064	0.039	0.028	0.034
Neural Biencoder (BGE-M3)	0.017	0.063	0.051	0.042	0.055
Neural Biencoder (E5-Large-V2)	0.011	0.038	0.026	0.020	0.026
Graph-Based Method					
GraphSAGE	0.005	0.017	0.008	0.004	0.006
R-GCN	0.002	0.011	0.011	0.011	0.011
GAT	0.009	0.045	0.020	0.012	0.012
TeG-DRec (GraphSAGE)	0.066	0.237	0.160	0.124	0.153
TeG-DRec (RGCN)	0.050	0.176	0.123	0.096	0.119
TeG-DRec (GAT)	**0.111**	**0.419**	**0.315**	**0.260**	**0.316**

Table 2: The recommendation performance of our method against baselines for the text-based IR method and the graph-based method. **Bold** is the best, underline is the second best.

Model	P@5	R@5	NDCG@5	MAP	MRR
w/o SciBERT	0.009	0.045	0.020	0.012	0.012
w/o GNNs (GAT)	0.015	0.053	0.032	0.023	0.029
TeG-DRec(GAT)	**0.111**	**0.419**	**0.315**	**0.260**	**0.316**

Table 3: The ablation study for each component. **Bold** is the best, underline is the second best.

3. **Graph Attention Networks (GAT)** (Veličković et al., 2018) introduces attention mechanisms, enabling the model to learn the importance of neighbouring nodes dynamically.

4.3 Results

Table 2 compares the performance of the graph-based method with **TeG-DReC** with text-based IR and graph-based methods alone on the DataFinder dataset, evaluated under a truly unseen configuration. Our methods consistently outperform all baselines across all metrics, demonstrating their effectiveness and robustness.

The results show that graph-based models combined with **TeG-DRec** outperform their graph-only counterparts across all evaluation metrics. In particular, **TeG-DReC**(GAT) achieves a substantial improvement in R@5, surpassing its baseline by 0.374. It also exhibits superior ranking performance, with gains of 0.304 in MRR, 0.295 in NDCG@5, and 0.240 in MAP compared with **TeG-DRec**(GAT). These metrics assess ranking quality where NDCG considers both relevance and position, MRR reflects the rank of the first relevant result, and MAP measures the average precision of the ranking. The P@5 metric also increases by 0.102 over the GAT baseline. Beyond **TeG-DRec**(GAT), both **TeG-DRec**(GraphSAGE) and **TeG-DRec**(RGCN) also achieve significant improvements over their respective graph-only baselines.

Although the neural bi-encoder using Contriever as the embedding model achieves the highest results among all text-based IR methods, all graph-based models integrated with **TeG-DRec** still outperform it. The lowest-performing variant, **TeG-DRec** (RGCN), surpasses the Contriever-based bi-encoder by 0.032, 0.068, 0.084, 0.085, and 0.112 for P@5, MAP, NDCG@5, MRR, and R@5, respectively. These results indicate that while neural bi-encoders capture rich semantic similarities, incorporating relational structure via graph learning further enhances alignment between scientific papers and datasets, leading to superior recommendation performance.

4.4 Ablation Study

To assess the contribution of each component in our model, we conducted an ablation study by removing one component at a time, with results shown in Table 3. In this ablation study we pick **TeG-DReC**(GAT) as our original results. Removing the

textual component, SciBERT, results in a substantial drop across all metrics, particularly in R@5 and MRR, which decrease by 0.374 and 0.304, respectively. This performance drop is also reflected in other metrics, such as NDCG@5, MAP, and P@5, which decrease by 0.295, 0.248, and 0.101, respectively. This underscores the critical role of textual features in capturing semantic alignment between papers and datasets for accurate recommendations.

Similarly, removing the GNN component also reduces performance across all metrics, with notable decreases in R@5 and MRR of 0.366 and 0.287, respectively. Other metrics, including NDCG@5, MAP, and P@5, also show decreases of 0.283, 0.237, and 0.096, respectively. These results indicate that while semantic representations of publications and datasets significantly improve model performance, integrating graph-structured information further enhances recommendation quality, highlighting the complementary benefits of combining textual and structural components in **TeG-DRec**.

4.5 Error Analysis

We conducted an error analysis on the **TeG-DRec** recommendation output. There are two main types of errors in the recommended results:

Biased towards certain dataset **TeG-DRec** shows a bias toward certain datasets, such as TreQA, which appears most frequently in recommendations even though it occurs only once in the ground truth, as shown in Figure 6. Similar trends are observed for other over-recommended datasets absent from the actual ground truth. A debiasing technique can be implemented to solve the problems.

Textual bias in dataset query Textual bias in the training data may affect the recommendations. For example, as shown in Figure 6, SQuAD, a question-answering dataset, appears 312 times in positive training interactions. This high frequency can bias the model toward recommending TreQA, another question-answering dataset, even when it is absent from the ground truth. Incorporating content-aware attention could help mitigate this issue.

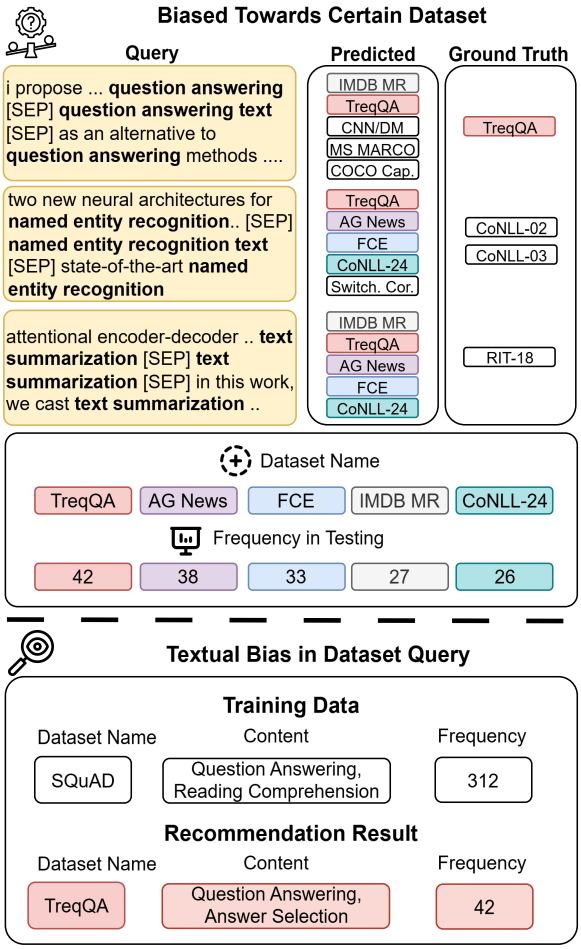

Figure 6: We present two examples for error analysis: the top illustrates a case where the **TeG-DRec** is biased towards a certain dataset, while the bottom highlights textual bias in the dataset query.

5 Conclusion

This research introduced **TeG-DRec**, a framework for scientific dataset recommendation that unifies GNNs with textual content via a subgraph module, ensuring that textual content and graph structures are correctly aligned and passed to the inductive graph and loss components. This integration enables the model to better generalise towards unseen data. The framework leverages textual representations from SciBERT and incorporates inductive GNNs, which are adaptable to various types of inductive graph models. Experimental results on the Datafinder dataset with truly unseen nodes show that our method outperforms previous baselines, including both text-based IR and graph-based approaches. Future work should incorporate a debiasing technique for recommendations to reduce popularity bias. This can be done by re-weighting the training loss based on dataset frequency, which means less frequent datasets get a higher weight. Aside from that, using content-aware attention rather than simply aggregating the textual embedding reduces bias from frequent, irrelevant words or phrases.

Limitations

Our major limitation is that our method relies heavily on the quality and availability of textual information (e.g, paper abstracts and dataset descriptions). In cases where the text is noisy, incomplete, or missing, the recommendation performance may degrade. Another limitation is that the availability of datasets for dataset recommendation systems is very low compared to other datasets which make use heavily rely on Datafinder Dataset alone.

Ethical Statement

This work adheres to the ethical standards outlined in the ACL Code of Ethics and the general principles of responsible AI research. All data used in this study are publicly available and used strictly for research purposes under their respective licenses. No personally identifiable information (PII) or sensitive content was collected or processed. We also took care to examine potential sources of bias and ensure that model outputs do not propagate harmful or discriminatory associations.

Acknowledgements

We would like to thank anonymous reviewers for their helpful comments and suggestions. This work is supported by the JKA (2024M-557). Ammar Qayyum is funded by the MEXT scholarship, Grant Number 233206.

References

Basmah Altaf, Uchenna Akujuobi, Lu Yu, and Xiangliang Zhang. 2019. Dataset recommendation via variational graph autoencoder. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 11–20. IEEE.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Tora Fahrudin and Dedy Rahman Wijaya. 2024. New custom rating for improving recommendation system performance. *Journal of Big Data*, 11(1):91.

Michael Färber and Ann-Kathrin Leisinger. 2021. Recommending datasets for scientific problem descriptions. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 3014–3018, New York, NY, USA. Association for Computing Machinery.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. *Preprint*, arXiv:1903.02428.

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

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Sarthak Jain, Madeleine Van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction. *arXiv preprint arXiv:2005.00512*.

Jüri Keller and Leon Paul Mondrian Munz. 2022. Evaluating research dataset recommendations in a living lab. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 135–148. Springer.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. 2025. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 4061–4065, New York, NY, USA. Association for Computing Machinery.

Özlem Özgöbek, Nafiseh Shabib, and Jon Atle Gulla. 2014. Data sets and news recommendation. In *UMAP Workshops*.

Shreya Patankar, Hitesh Prajapati, Jeet Shah, and Ankit Upadhyay. 2023. Automl-learning, understanding and applying machine learning to datasets. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 919–922. IEEE.

Ammar Qayyum, Bassamtiano Irnawan, Sheng Xu, Zihao Hu, Fumiyo Fukumoto, and Yoshimi Suzuki. 2025. Textual graph contrastive learning for enhanced dataset recommendation. In *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–4. IEEE.

Ziheng Qin, Zhaopan Xu, Yukun Zhou, Zangwei Zheng, Zebang Cheng, Hao Tang, Lei Shang, Baigui Sun, Xiaojiang Peng, Radu Timofte, and 1 others. 2024. Dataset growth. In *European Conference on Computer Vision*, pages 58–75. Springer.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S Zimmermann, and Wieland Brendel. 2024. Infonce: Identifying the gap between theory and practice. *arXiv preprint arXiv:2407.00143*.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *International conference on machine learning*, pages 9448–9457. PMLR.

Hung-Nghiep Tran, Akiko Aizawa, and Atsuhiro Takasu. 2024. An encoding–searching separation perspective on bi-encoder neural search. *arXiv preprint arXiv:2408.01094*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Preprint*, arXiv:1710.10903.

Katrien Verbert, Hendrik Drachsler, Nikos Manouselis, Martin Wolpers, Riina Vuorikari, and Erik Duval. 2011. Dataset-driven research for improving recommender systems for learning. In *Proceedings of the 1st international conference on learning analytics and knowledge*, pages 44–53.

Vijay Viswanathan, Luyu Gao, Tongshuang Wu, Pengfei Liu, and Graham Neubig. 2023. Datafinder: Scientific dataset recommendation from natural language descriptions. *arXiv preprint arXiv:2305.16636*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 317–324, New York, NY, USA. Association for Computing Machinery.

Jiaren Xiao, Quanyu Dai, Xiaochen Xie, James Lam, and Ka-Wai Kwok. 2023. Adversarially regularized graph attention networks for inductive learning on partially labeled graphs. *Knowledge-Based Systems*, 268:110456.

Krishan Kant Yadav, Hemant Kumar Soni, and Nikhlesh Pathik. 2023. Recommendation system based on double ensemble models using knn-mf. *International Journal of Advanced Computer Science and Applications*, 14(5).

Zitong Zhang and Yaseen Ashraf. 2023. A content-based dataset recommendation system for biomedical datasets. In *2023 6th International Conference on Information and Computer Technologies (ICICT)*, pages 198–202. IEEE.

Appendices

A Datafinder Dataset Content

Table 4 presents the structure of the Datafinder Dataset. The dataset is organised into three main components: training scientific paper metadata, test scientific paper metadata, and dataset metadata. Each component is further divided into several categories, including content descriptions of papers and datasets, datasets referenced by scientific papers, publication metadata, unique identifiers, citation details, and additional information related to papers and datasets. The highlighted fields in Table 4 indicate the features utilised as node attributes for each corresponding entity in our model.

Training Scientific Paper		
Paper Content	**Paper ID**	**Paper Information**
title	paper_id	has_pdf_body_text
abstract	arxiv_id	mag_field_of_study
query	acl_id	has_inbound_citations
keyphrase_query	pmc_id	has_outbound_citations
Dataset	pubmed_id	has_pdf_sparse
positives	mag_id	has_pdf_sparse_abstract
negatives	**Citation Information**	has_pdf_parse_bib_entries
Paper Publication	author	has_pdf_parse_text
journal	outbound_citations	has_pdf_parse_body_text
venue	inbound_citations	has_pdf_parse_entries
doi		s2_url
year		

Test Scientific Paper		
Paper Content	**Paper ID**	**Paper Information**
abstract	-	task
query	**Citation Information**	domain
keyphrase_query	-	modality
Dataset		language
documents		training_style
Paper Publication		text_length
year		

Dataset		
Dataset Content	**Dataset ID**	**Dataset Information**
title	id	variants
content	**Citation Information**	
structured_info	-	
Dataset Publciation		
year		
date		

Table 4: Datafinder Dataset content, the highlighted box is the features which is used for node features

B Hyperparameters value

Table 5 shows the hyperparameter setting for the parameters that are used in **TeG-DRec**. The hyperparameters include the maximum length of the textual encoder, the hidden dimension, the optimiser and its learning rate, the number of epochs, the loss rate, the loss temperature, and the seed number.

Variable	**Value**
SciBERT Dimension	512
Hidden Dimension	256
Optimizer	Adamw
Learning Rate	GraphSAGE : 1e-3, R-GCN: 5e-3, GAT: 5e-3
Epoch	40 with early stopping after 5 epoch of no improvement
Warmup Epoch's Scheduler	5
InfoNCE Temperature	0.08
Margin Value	1
Contrastive Loss Rate	0.8
Margin Loss Rate	0.8
L2 Regression Loss	1e-4
Seed Number	1

Table 5: Hyperparameters variable and its value for the reproducibility purpose