Author Name Disambiguation via Graph-Enhanced Language Model Fine-Tuning

Anonymous ACL submission

Abstract

Author name disambiguation (AND) serves as a core component of modern academic search systems to curate author profiles and bibliometrics. Recently, language models (LMs) and graph neural networks (GNNs) have significantly pushed the frontier of modeling textual and relational information. However, their representation powers are not fully exploited to improve the accuracy of AND. In this work, we propose a unified model – graph-enhanced language model (*i.e.* GAND) that enables *joint* modeling of the text information and relations between documents. Compared to the traditional contrastive loss, we develop a multi-task fine-tuning objective. This not only mitigates potential distribution shifts in testing data but also improves the efficiency of fine-tuning language models for AND. Experiments on two real datasets for name disambiguation demonstrate the superior performance of GAND over embedding-based approaches, fine-tuning LMs and OpenAI's text embeddings.

1 Introduction

The rise of the academic search engines (Ammar et al., 2018; Lu, 2011; Sinha et al., 2015; Tang et al., 2008) greatly facilitates modern research activities, which lets researchers efficiently retrieve relevant papers and scholars from the bibliographic database. Platforms like Google Scholar, Semantic Scholar (Ammar et al., 2018) and PubMed (Lu, 2011) curate massive amounts of research publications and profiles. Meanwhile, there are thousands of researchers who share the same or similar names. According to Google Scholar[1], there are at least 35 "James White"s and 14 "Michael Jordan"s in its database. When a new article is published under a common name, how to accurately perform author name disambiguation (AND) remains an open but challenging problem.

[1] https://scholar.google.com/

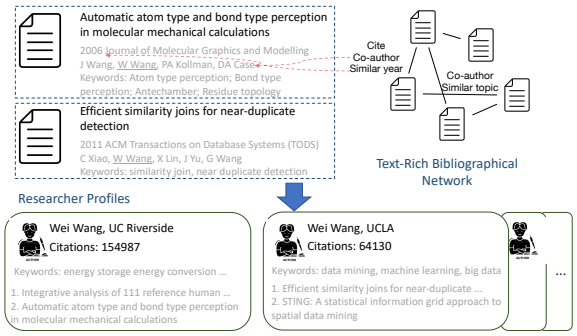


Figure 1: An author name disambiguation system.

An author name disambiguation (Han et al., 2004; Tang et al., 2011) system aims to group publications of the same real-world person into distinct profiles in cases where multiple authors share the same name (*e.g.* "Wei Wang" in Figure 1). To determine the real-world person of each mention in publications, the common industrial pipeline (Zhang et al., 2018; Subramanian et al., 2021) includes three steps: (1) creating a document pool (also known as a block) based on the same name string; (2) transforming different features of each document (e.g., content and co-authors) into a dense vector; and (3) performing a hierarchical linkage-based clustering considering the pairwise similarity between documents under the same name.

In these steps, representation learning (Levin et al., 2012; Subramanian et al., 2021) of documents is identified as the most fundamental task of AND. There are two modalities to be considered when modeling the similarity between two documents: (1) *textual information*: the similarity of the content and (2) *relational information*: the relevance of the meta information such as co-authors, citations, venues, *etc.* Despite the fact that joint modeling of both modalities have been extensively studied (Wang et al., 2020; Zhang et al., 2021), existing models suffer from two disadvantages based on our observation.

Lack of Advanced Unified Modeling. Exist-

ing approaches focus either on modeling textual similarity or on capturing relations between documents. For example, graph-based methods (Fan et al., 2011; Tang et al., 2011) leverage relations between documents like *co-author*[2] and employ affinity propagation techniques for clustering. On the other hand neural network-based approaches learn the encoding of text embeddings through contrastive learning (Zhang et al., 2018) or pair-wise classification (Subramanian et al., 2021). However, the potential for a unified modeling approach that integrates both textual and relational features is only explored at the word embedding level (Wang et al., 2020; Zhang et al., 2021). The unified modeling of pre-trained language models and graph neural networks remains underexplored.

Limited Out-of-Distribution Generalization. Recently, fine-tuning a pre-trained language model such as Sentence-BERT (Reimers and Gurevych, 2019) greatly advances the accuracy of measuring document similarities. Nevertheless, the task of author name disambiguation (AND) often involves dealing with out-of-distribution testing data. In such cases, the topic distribution or the criteria of "written by the same author" may vary significantly from the training corpus. In our experiment (see also Figure 3), we find traditional triplet loss (Cohan et al., 2020) cannot improve the AND accuracy on testing documents, where our observation is consistent with the previous findings on fine-tuning PLMs with OOD issues (Kumar et al., 2022).

To cope with the aforementioned two challenges, we propose a novel framework GAND for author name disambiguation. In our framework, relational (*i.e.* relation with other papers) and textual attributes of articles are transformed into a text-rich bibliographical network $\mathcal{G}$ as shown on the left side of Figure 1. Specifically, there are two major differences between GAND and existing AND algorithms: (1) A *unified* LM-based representation learning module jointly embeds the textual and relational attributes of each document. It consists of a pre-trained language model (PLM) encoder followed by a graph attention layer (Velickovic et al., 2017), which learns the importance scores of neighbors regarding embeddings derived from the PLM. In this way, the neighbors with low content similarity would not contribute to the final representation of the target document. We train such a graph-enhanced language model (*i.e.* GNN-LM) with annotated AND data. (2) A *multi-task* training objective - **Multi-FT** is devised to overcome the overfitting issue of the triplet loss. In our approach, we treat each distinct surface name in the training data as a separate AND task. Our model optimizes a separate classification head for each task, interpreting them as proxies that represent different individuals who share the same name. Consequently, it does not enforce the embeddings of every positive document pair being close in the latent space. Instead, the representations of documents are updated less if they are classified to the correct author already. **Multi-FT** eliminates the need for negative samples and reduces computational cost by approximately 50% when compared to traditional triplet loss. In our experiments, we compare GAND with other AND algorithms and state-of-the-art language model fine-tuning methods on two name disambiguation datasets. Our framework outperforms baselines on four different clustering metrics by a clear margin (*i.e.* Table 2). In addition, we observe consistent improvements on other baselines by employing the **Multi-FT** loss (see also Figure 4).

Our contributions could be summarized as: (1) we propose a graph-enhanced PLM framework unifying textual and relational information for author name disambiguation; (2) a multi-task objective is introduced to learn the embedding function, offering better generalization for out-of-distribution data; (3) extensive experiments on two different AND datasets demonstrate the effectiveness and efficiency of GAND.

2 Problem Definition

In this paper, we followed the terminology in (Zhang et al., 2018). Given a collection of academic publications $\mathcal{D}$, a *reference* or surface name refers to a set of authors $\mathcal{A}$, where each $a \in \mathcal{A}$ denotes a real-person with the same string name. A *block* $\mathcal{D}_{\mathcal{A}} \subset \mathcal{D}$ contains all publications from every author in *reference* $\mathcal{A}$. Feature x_i of paper $d_i \in \mathcal{D}_{\mathcal{A}}$ consists of textual attributes such as its title and relational attributes such as author names, venue names, *etc.* We use $\mathbb{I}(d_i) = a_i$[3] to represent the real author a_i of paper i within reference $\mathcal{A}$. Therefore, $\mathbb{I}(d_i) = \mathbb{I}(d_j)$ if two papers d_i, d_j are authored by the same identity (*i.e.*

[2]If two documents have a common author, they will be connected via the relation *co-author*.

[3]If there are multiple authors need disambiguation in d_i, we can duplicate the document to resolve this problem.

real-world person). Given a set of training references $\mathcal{R}_{\text{train}} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$, the corresponding blocks are $\mathcal{B}_{\text{train}} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n\}$. Each training document d_i is a tuple of (r_i, x_i, a_i), where r_i denotes the specific reference and block of d_i, x_i is the feature and $a_i \in \mathcal{A}_{r_i}$ is the real author id. We define the task of AND as follows.

Definition 2.1 (Representation Learning for Author Name Disambiguation)**.** *Given documents from training blocks $\mathcal{B}_{train}$, the task of author name disambiguation aims to learn an embedding function $\mathbf{f} : (r_i, x_i) \rightarrow \mathbb{R}^d$ that maps document of the same author close in the latent space.*

For each testing reference r'_i, AND performs a hierarchical clustering on document embeddings $\mathbf{f}(x'_i)$ to obain the cluster membership c'_i. The quality of AND is measured between clusters (r'_i, c'_i) and ground truth (r'_i, a'_i). Our formulation of AND as embedding learning shares the same setting with most of the related studies (Zhang et al., 2018; Subramanian et al., 2021).

3 Method

In this section, we first describe how to construct the document graph using textual and relational attributes from candidate documents. Then we introduce a graph-enhanced language model encoder that unifies GNN and PLM architectures to represent each document in the graph. Finally, we design a multi-task fine-tuning objective to alleviate the generalization issue of traditional PLM fine-tuning methods for AND.

3.1 Representing Textual and Relational Attributes

In the problem definition, we categorize the features of documents into textual and relational attributes. Specifically, each document d_i has text information including title w and abstract v. We concatenate w and v as a joint sequence of tokens "[CLS] $w_1, ..., w_m$ [SEP] $v_1, ..., v_n$". Most relational attributes can be represented as a unique identifier, for example, paper d_i has two authors "author_2023", "author_2025"[4] and cites "paper_156".

Graph Construction. In order to model relations between documents, we construct a document graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ as follows: (1) each node is a document, and an edge e_{ij} exists if document d_i and d_j has relation r_{ij}. In this paper, we mainly consider first-order and second-order relations. For example, *cite* is a first-order relation and $e_{ij} \in \mathcal{E}$ if d_i cites "paper_j" or vice versa. *co-author* is a second-order relation such that $e_{ij} \in \mathcal{E}$ if both document d_i and d_j shares an (unambiguous) author "author_x". The textual attributes and relational attributes are transformed into node features $\mathcal{X}$ and graph edges $\mathcal{E}$, respectively.

[4]we only use the co-author names without ambiguity.

3.2 Graph-Enhanced PLM Encoder

The purpose of representation learning for author name disambiguation is to embed documents from the same author closely such that off-the-shelf clustering methods can distinguish among different identities easily. As shown in Figure 2, we feed both the target document d_i and its neighbors $\mathcal{N}_i$ in the constructed graph $\mathcal{G}$ into the joint PLM-GNN model. To be specific, given the token sequence $\{w_1, ..., w_t\}$ of the target and neighbor documents, we adopt mean pooling of token embeddings from the last layer of a pre-trained PLM as the intermediate document embedding $\mathbf{h}_i = \text{MEAN}\{h_1^{(l)}, ..., h_t^{(l)}\}$,

$$\{h_1^{(l)}, ..., h_t^{(l)}\} = \textbf{PLM}\{w_1, ..., w_t\}, \quad (1)$$

The neighbor documents are those documents connected to the target document in $\mathcal{G}$. In the Introduction, we have discussed that not every neighbor can help with disambiguation. To the light of this, our model encourages the final representation to have relevant neighbors by learning an attention weight α on text representations h between the target and neighbors. We calculate attention weights $\{\mathbf{a}, \mathbf{\Theta}\}$ following graph attention networks (Velickovic et al., 2017),

$$\alpha_{i,j} = \frac{\exp\left(\sigma\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{h}_i \,\|\, \mathbf{\Theta}\mathbf{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp\left(\sigma\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{h}_i \,\|\, \mathbf{\Theta}\mathbf{h}_k]\right)\right)}, \quad (2)$$

where the activation function σ is LeakyReLU; $\mathbf{a} \in \mathbb{R}^{1\times d}$. The final representation of document d_i is,

$$\mathbf{z}_i = \alpha_{i,i}\mathbf{\Theta}\mathbf{h}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}\mathbf{h}_j. \quad (3)$$

3.3 Multi-task Fine-Tuning

In the previous section, our proposed PLM-GNN encoder turns textual and structural attributes into a unified representation $\mathbf{z}_i$, while how to learn generalizable attention parameters $\{\Theta, \mathbf{a}\}$ and how to fine-tune the PLM remain challenging.

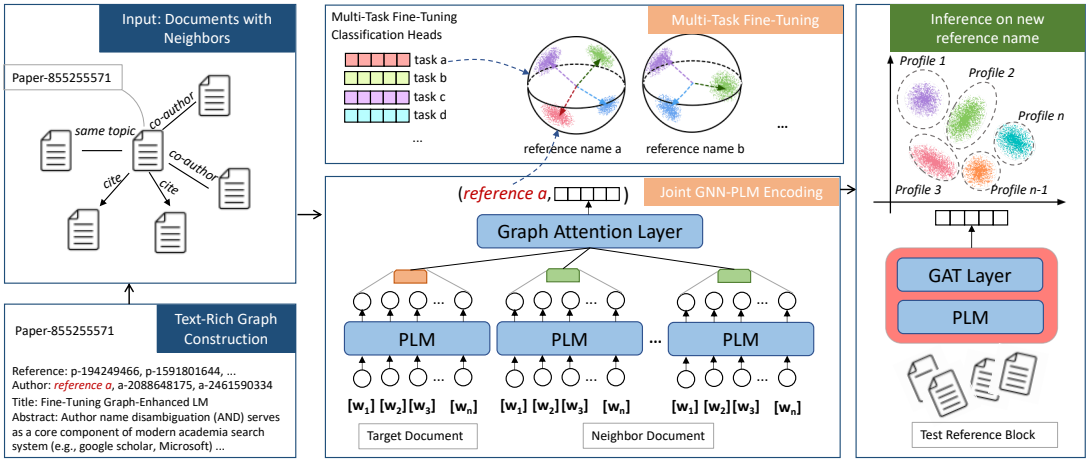
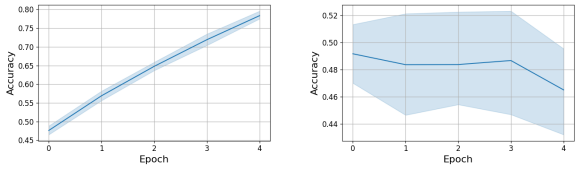


Figure 2: Framework Overview. In each AND task $\mathcal{U}_i$, the number of classes is determined by the number of different individuals in $\mathcal{A}_i$. Classification heads of training references are not used during inference.



(a) Train Accuracy (b) Test Accuracy

Figure 3: Overfitting Issue of Fine-Tuning PLM.

The triplet margin loss is widely used in contrastive PLM pre-training. For example, SPECTER (Cohan et al., 2020) maximizes the margin between the similarity of positive pairs and that of negative pairs. In AMiner-AND (Zhang et al., 2018), authors also adopt a similar idea on each reference r:

$$\mathcal{L}_{\textbf{FT}} = \sum_{i=1}^{N} \max\Big\{d(\mathbf{z}_i, \mathbf{z}_{i^+}) - d(\mathbf{z}_i, \mathbf{z}_{i^-}) + m, 0\Big\} \tag{4}$$

where d is a distance metric such as $L2$-norm; $(r, \mathbf{z}_{i^+}, a_{i^+})$ is a positive document if they are authored by the same real-world person $a_i = a_{i^+}$ and $(r, \mathbf{z}_{i^-}, a_{i^-})$ is a negative document if $a_i \neq a_{i^-}$.

However, if we fine-tune PLMs using the same loss for AND, we observe the pairwise accuracy drops as the training accuracy increases in Figure 3. In the literature, Kumar et al. (2022) recently observed that the results of fine-tuning PLMs are worse than linear probing when the distribution shift between training and testing corpora is large. In AND, the number of testing references is often much larger than training references. It is possible that documents of testing blocks can vary a lot from training documents on topics or domains.

To alleviate the out-of-distribution challenge, we formulate the training of AND as a multi-task fine-tuning (**Multi-FT**) problem, where each task corresponds to a training reference name. Given a small amount of training tasks $\{\mathcal{U}_1, ..., \mathcal{U}_M\}$, each task disambiguates documents from block $\mathcal{D}_i$ of references $\mathcal{A}_i$. Through mini-batch training, given each training sample as a triple of (document, ground-truth author, task) as $\{\mathbf{z}_i, a_i, \mathcal{U}_i\}$, we propose the following multi-task learning loss,

$$\mathcal{L}_{\textbf{Multi-FT}} = \sum_{i=1}^{N} \left(\frac{\exp(\mathbf{W}_{a_i}^{\mathcal{U}_i} \mathbf{z}_i)}{\sum_{j=1}^{|\mathcal{A}_i|} \exp(\mathbf{W}_{a_j}^{\mathcal{U}_i} \mathbf{z}_i)} \right) \tag{5}$$

where $\mathbf{W}^{\mathcal{U}_i} \in \mathbb{R}^{|\mathcal{A}_i| \times d}$ is the task-specific classification heads and $\mathbf{z}_i$ is the normalized document embeddings in Equation 3. In the section "Multi-Task Fine-Tuning" of Figure 2, we visualize the classification of each task. For each training reference name $\mathcal{A}_i$, each row of the classification heads $\mathbf{W}_{a_i}^{\mathcal{U}_i}$ can be interpreted as the embedding of the ground-truth author a_i, which is also called proxy in the literature of metric learning (Movshovitz-Attias et al., 2017). Compared to $\mathcal{L}_{\textbf{FT}}$, this approach only requires that documents from the same author a_i are closer to the same proxy embedding $\mathbf{W}_{a_i}^{\mathcal{U}_i}$, offering two advantages: (1) it reduces the computational complexity involved in encoding negative data points, and (2) it addresses out-of-distribution (OOD) generalization by imposing a provable relaxed constraint (Movshovitz-Attias et al., 2017). In Algorithm 1, we summarize the training and inference procedure of GAND.

Model Analysis. There are four hyperparameters introduced by GAND, they are the maximum size

K of the neighborhood $\mathcal{N}$, the number of GNN layers L, the maximum length of a document N, and the batch size B. The space complexity of GAND is $\mathcal{O}(B \cdot N \cdot K^L)$. We denote the time complexity of PLM to encode a document of length N as α_{PLM}. The graph attention layer takes α_Θ time to compute embeddings of the document. For training references with a total of $|\mathcal{D}|$ documents, the total time complexity of our approach is $\mathcal{O}(|\mathcal{D}| \cdot K^L(\alpha_\Theta + \alpha_{\text{PLM}}))$. We set $L = 1$ in our experiment with ablation study in Section 4.3.

Algorithm 1: Pseudo code for GAND optimization

Input: a set of training references $\mathcal{R}_{\text{train}}$,
PLM, testing references $\mathcal{R}_{\text{test}}$,
Graph Neighborhood Sampler **SAMPLE**.
Output: test document embedding $\mathbf{z}'$.
// Construct Graph $\mathcal{G}_s$, $\mathcal{G}_t$;
$\mathcal{G}_s \leftarrow \mathcal{A}_{\text{train}}, \mathcal{G}_t \leftarrow \mathcal{A}_{\text{test}}$
// Training;
for *each batch of* $\{x_i, a_i, \mathcal{U}_i\}$ *from* **SAMPLE**$(\mathcal{G}_s)$ **do**
  compute $\mathbf{z}_i \leftarrow$ Eq.(3),
  $\mathcal{L} \leftarrow$ Eq.(5);
  update $\{\mathbf{a}, \Theta, \text{PLM}_\Theta\}$;
end
// Testing;
for *each document* $\mathbf{x}'_\mathbf{i}$ *from* **SAMPLE**$(\mathcal{G}_t)$ **do**
  compute $\mathbf{z}'_\mathbf{i} \leftarrow$ Eq.(3)
end
return $\mathbf{z}'$

4 Experiments

4.1 Evaluation Setup

Datasets. We evaluate the performance of author name disambiguation on two public datasets: (1) AMiner (Zhang et al., 2018) collects 600 ambiguous name references, and about 203K documents are published by these authors. (2) MAG-CS is a subset of Microsoft Academic Graph (MAG) (Sinha et al., 2015) in the computer science domain.[5] In order to create a challenging AND task, we choose ambiguous references with at least three different real identities and each of them has at least five publications between 2000 and 2020. Each paper in both datasets contains its authors and bibliographies. We preprocess the documents into graph as described in Section 3.1, where *cite* is a first-order relation and *co-author* is a second-order relation. The total number of documents and graph statistics are summarized in Table 4.

[5]We select papers from top 105 venues in the field of computer science from MAG.

Table 1: Overall Dataset Statistics

	AMiner	MAG-CS
# Documents	203078	6895
# Authors	6228	454
# Total Edges	4.99M	63.6K
# Co-author Edges	4.99M	62.2K
# Cite Edges	N/A	1463
# Total References	600	125

Compared Methods. We utilize MPNet (Song et al., 2020) as the backbone PLM for our approach. The performance of other SciBERT variants can be found in the Appendix § A.3. We compare GAND with existing AND algorithms, fine-tuned language models, and graph neural networks. These methods are: (1) AMiner-AND (Zhang et al., 2018) learns global document embedding through contrastive learning and local graph embeddings in the document graph under each reference. (2) S2AND (Subramanian et al., 2021) constructs pairwise linkage features and then trains a gradient boosted trees (GBT) classifier to estimate the similarity between document pairs. For pre-trained language models, we continue fine-tuning their checkpoints using a triplet margin loss as of (Cohan et al., 2020). (3) SPECTER (Cohan et al., 2020) is a citation-informed language model pretraining method utilizing the citations between scientific documents. (4) MPNet (Song et al., 2020) is a state-of-the-art masked language model that unifies masked and permuted pre-training for language understanding tasks. We also include two representative graph neural networks using MPNet encoded representation as node features. (5) MPNet+SGC (Wu et al., 2019) simplifies the consecutive nonlinearities and weight matrices of traditional graph convolution networks (Kipf and Welling, 2017) with better scalability. (6) MPNet+GAT (Velickovic et al., 2017) employs attention between target and neighbor nodes in the graph and we also use this architecture in GAND. Similar to experiments in SPECTER (Cohan et al., 2020), We freeze the language model embeddings in these two methods because of the OOM issue of fine-tuning PLMs when

Table 2: Author name disambiguation results on two datasets. We report the mean_{std} of five runs for all the methods. Scores marked with ** (resp., *) pass the t-test with $p < 0.05$ (resp., $p < 0.1$) in comparison with the second best.

Method	MAG-CS				AMiner			
	Micro-F1	Macro-F1	B^3 F1	NMI	Micro-F1	Macro-F1	B^3 F1	NMI
AMiner-AND	$61.98_{2.49}$	$65.14_{1.39}$	$67.20_{1.43}$	$52.63_{2.61}$	$71.91_{0.33}$	$67.26_{0.29}$	$71.62_{0.25}$	$\mathbf{65.13}^{**}_{0.36}$
S2AND	$73.92_{2.81}$	$\underline{75.29}_{1.32}$	$\underline{77.32}_{1.35}$	$\underline{66.03}_{2.16}$	$73.08_{0.08}$	$69.40_{0.20}$	$72.32_{1.35}$	$\underline{63.84}_{0.15}$
SPECTER	$70.58_{2.48}$	$69.12_{1.78}$	$72.68_{1.54}$	$53.55_{1.87}$	$69.17_{0.10}$	$62.20_{0.20}$	$66.01_{0.01}$	$55.53_{0.05}$
MPNet	$66.92_{2.33}$	$67.49_{1.53}$	$70.88_{1.34}$	$52.23_{2.19}$	$69.62_{0.03}$	$63.08_{0.22}$	$66.74_{0.31}$	$56.15_{0.29}$
MPNet+SGC	$72.60_{2.75}$	$73.99_{1.40}$	$76.31_{1.41}$	$64.18_{1.92}$	$74.16_{0.33}$	$68.26_{0.54}$	$73.23_{0.28}$	$64.22_{0.43}$
MPNet+GAT	$73.25_{2.35}$	$73.92_{1.47}$	$76.52_{1.39}$	$63.68_{2.14}$	$73.09_{0.24}$	$67.57_{0.07}$	$72.10_{0.09}$	$62.45_{0.23}$
OAG-BERT	$69.43_{2.02}$	$68.38_{1.24}$	$71.79_{0.91}$	$52.44_{1.90}$	$69.56_{0.07}$	$62.21_{0.34}$	$66.19_{0.20}$	$55.41_{0.33}$
OpenAI-embeddings	$72.29_{2.18}$	$70.99_{0.62}$	$74.58_{0.69}$	$57.12_{0.9}$	$74.67_{0.00}$	$68.04_{0.00}$	$72.43_{0.00}$	$62.61_{0.00}$
GAND **w.o.** GNN	$70.35_{3.90}$	$69.97_{1.63}$	$72.83_{1.79}$	$57.80_{1.84}$	$74.94_{0.16}$	$68.82_{0.16}$	$72.35_{0.07}$	$63.51_{0.63}$
GAND **freeze** PLM	$\underline{74.31}_{2.97}$	$73.59_{1.57}$	$77.16_{1.43}$	$62.78_{2.09}$	$\underline{75.28}_{0.05}$	$\underline{69.51}_{0.14}$	$\underline{74.26}_{0.06}$	$62.99_{0.91}$
GAND	$\mathbf{75.02}^{*}_{2.68}$	$\mathbf{75.94}^{*}_{1.63}$	$\mathbf{77.98}^{*}_{1.48}$	$\mathbf{67.23}^{**}_{1.96}$	$\mathbf{75.80}^{**}_{0.46}$	$\mathbf{70.53}^{**}_{0.60}$	$\mathbf{74.81}^{*}_{0.42}$	$63.75_{0.48}$

neighborhood size grows exponentially in GNNs. (7) OAG-BERT (Liu et al., 2022) jointly encodes scientific text and venue/author information with an entity-augmented academic language model. (8) OpenAI-embeddings uses the same augmented text with OAG-BERT and generate embeddings through API calls.[6] (9) GAND **w.o.** GNN is a variant of our approach, in which we remove all neighbors, that is, $\mathcal{N} = \varnothing$ in Equation 2. (10) GAND **freeze** PLM freezes the parameter of PLM in Equation 1.

Evaluation Metrics. We evaluate the clustering results of different methods via four metrics: (1) Pairwise Micro-F1 is the harmonic mean of precision and recall between all predicted pairs. (2) Pairwise Macro-F1 computes the average F1-score of testing references r'_i. (3) B^3 F1 (Subramanian et al., 2021) is a co-reference resolution metric that computes the precision and recall based intersection between ground truth cluster $\mathcal{A}$ and predicted clusters $\mathcal{C}$. (4) Normalized Mutual Information (NMI) is a symmetric metric to measure the quality of clustering results. We calculate the average NMI score of all testing references.

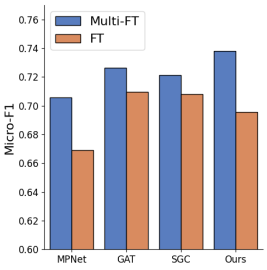
Experiment Settings. The inputs of the compared algorithms are the same documents and text-rich networks. On training references, we process the data following the public implementation of each method. On test references, we evaluate the quality of document embeddings by performing the same hierarchical clustering algorithm. On AMiner, we run all methods under the default training (500 references) and testing (100 references) five times. In addition, we separate 100 references from the training set as validation. On MAG-CS, we perform a 5-fold cross-validation under five different random seeds. In each round, 20% and 20% of data are used for training and validation sets, respectively. Following existing studies (Zhang et al., 2018), we use Macro-F1 on the validation set to select the best checkpoint for evaluation. All models are trained for 10 epochs in each run on a single Nvidia A6000 GPU with the batch size as 16. Configuration of all the hyperparameters can be found in Appendix § A.1. The source code can be found in the supplementary material. We will release the data used in the experiment upon acceptance.

4.2 Experimental Result

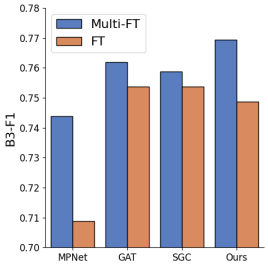
In Table 2, we show the performance of all compared algorithms. We report the significance level of the best result under each metric against the second runner through a two-tailed t-test.

First, we observe that document representations from pre-trained language models do not outperform the feature-based method (S2AND) and contrastive representation learning (AMiner-AND). These methods also surpass the entity-augmented LM (OAG-BERT) and LLMs (OpenAI-embeddings), demonstrating that relational attributes significantly enhance the performance of author name disambiguation. Third, GAND reports the best performance on 7 out of 8 metrics across two datasets, indicating our design of joint GNN-PLM encoding successfully benefits from PLMs

[6] We use text-embedding-3-small in our experiments and we do not see significant improvements for text-embedding-3-large. https://platform.openai.com/docs/guides/embeddings

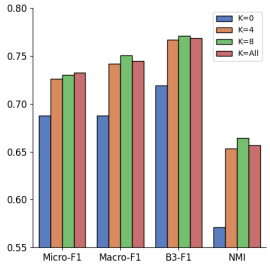


(a) Micro-F1

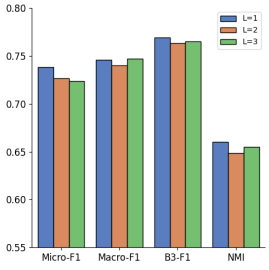


(b) B^3 F1

Figure 4: Effect of **Multi-FT** on GAND and baselines on MAG-CS.



(a) Effect of #neighbors



(b) Effect of #layers

Figure 5: Parameter study on the effect of attention and number of neighbors.

and graph neural networks. At last, when compared with our own variants, we find both graph structure and fine-tuning PLMs improve the accuracy, especially on MAG-CS. We notice the performance gap on AMiner is smaller. Because documents in the same blocks from AMiner are collected from multiple domains (*e.g.* chemistry and computer science), either textual or relational information can already distinguish documents from different authors in this case. Nevertheless, our full model still reports statistically significant performance improvements.

Effect of Multi-Task Fine-Tuning. In Section 3.3, we discussed the out-of-domain challenge of fine-tuning PLMs for author name disambiguation and proposed a multi-task training objective. Besides the performance improvements we observed in Table 2, we apply the same objective on our backbone PLM (*i.e.* MPNet) and two GNN baselines. We compare the performance of original fine-tuning (**FT**) via triplet loss (*i.e.* Equation 4) and multi-task fine-tuning (**Multi-FT**) in Figure 4. Full results on both datasets can be found in Appendix §A.5. We have three observations: (1) multi-task fine-tuning can improve the performance of various baselines in most cases, but still worse than GAND. (2) Multi-FT notably improves the test performance of MPNet and GAND, which confirms that out-of-distribution (OOD) testing documents adversely affect contrastive fine-tuning PLMs. (3) The marginal improvement seen in baselines (*e.g.* GAT, SGC) that freeze PLM embeddings reaffirms that the extensive parameter space of PLMs becomes a challenge for OOD generalization when training data is limited. Multi-FT demonstrates promising potential as a training objective for applications that have limited supervision available.

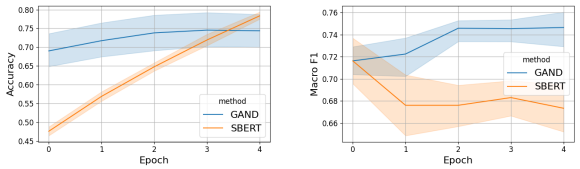
4.3 Model Study

Hyperparameter Sensitivity. Compared with PLMs, the additional complexity of GAND comes from the graph attention layer in Equation 2. In our analysis, the time complexity of GAND increases linearly with the number of neighbors K and exponentially with the number of GNN layers L. Figure 5 demonstrates the result of GAND by varying the number of neighbors and number of GNN layers. We observe that GAND performs significantly better than our variant without GNN (K=0 in Figure 5a), highlighting the importance of *relational information* encoded in the constructed graph for achieving better performance. Meanwhile, we observe that increasing the number of neighbors (*e.g.* K=4, 8, All) or the number of layers in GAND does not lead to additional performance improvements. We believe a small number of *co-authored* or *cited* neighbor documents is enough for GAND. Hence, our model does not require much more computations than fine-tuning PLMs. According to this result, we believe setting $K = 5$ and $L = 1$ in our main experiment is reasonable.

Analysis of $\mathcal{L}_{\text{Multi-FT}}$. In Figure 3, we present the generalization issue observed when fine-tuning PLM using triplet loss. Now we compare the training and testing performance of MPNet and GAND on MAG-CS for five runs. Figure 6 illustrates that both models achieve high pairwise Micro-F1 scores by clustering the training data. However, only GAND demonstrates an improved Macro-F1 on the testing data, whereas the performance of MPNet declines throughout the training process. In addition, GAND does not require encoding negative samples, which effectively reduces the computation cost by approximately half, assuming one negative sample per positive pair. In Table 3, GAND (**Multi-FT**) exhibits approximately 50% reduced training

Table 3: Training time per epoch comparison between **FT** and **Multi-FT**.

Model Arch.	MAG-CS	AMiner
MPNet (**FT**)	1min0s	95min37s
GAND (**FT**)	3min06s	246min44s
GAND (**Multi-FT**)	2min24s	121min47s



(a) Training Micro-F1 (b) Testing Macro-F1

Figure 6: Comparison of Fine-Tuning PLMs and GAND.

time compared to GAND (**FT**), particularly on the large dataset AMiner.

5 Related Work

Author Name Disambiguation. There have been lots of efforts in the field to perform author name disambiguation in bibliographic databases. Early approaches (Han et al., 2004, 2005) define various similarity metrics between pairs of articles and apply unsupervised clustering algorithms to correspond each cluster to a real-world author. Ever since feature engineering became the most critical step towards successful disambiguation. Lots of pairwise features (Louppe et al., 2016; Song et al., 2015; Treeratpituk and Giles, 2009) are proposed to train pairwise classifiers such as co-authors, affiliations, ethnicity *etc.* People also collected multiple AND datasets (*e.g.* AMiner (Zhang et al., 2018), INSPIRE (Louppe et al., 2016)) and trained pairwise classifier. The notable ones are random forests (Jhawar et al., 2020; Subramanian et al., 2021) and deep neural networks (Kim et al., 2019; Zhang et al., 2018). Similar to GAND, (Fan et al., 2011; Tang et al., 2011; Zhang et al., 2021) constructed a document graph with relations between documents and conducted clustering on the graph. In this work, we focus on learning the document representations for author name disambiguation. To this end, both AMiner-AND (Zhang et al., 2018) and AND-GAT (Zhang et al., 2021) learn a neural network encoder on word embeddings. However, none of these algorithms simultaneously model deep contextualized text embeddings and relational information.

Pre-trained Language Models for Scientific Text. The first well-known pre-trained masked language model - BERT (Devlin et al., 2018) propose to train a deep bidirectional transformer using masked token and next sentence prediction tasks. SciBERT (Beltagy et al., 2019) pre-trains a BERT model on multiple scientific publication corpus for downstream applications. BioBERT (Lee et al., 2020) is another similar approach but pre-trains BERT model on biomedical corpora. Sentence-BERT (Reimers and Gurevych, 2019) introduces the triplet objective to derive the sentence embeddings that can be compared using cosine similarity. SPECTER (Cohan et al., 2020) applies fine-tuning to SciBERT using a triplet loss, aiming to maximize the margin between the similarity of a query paper and its citations compared to randomly sampled papers. SciNCL(Ostendorff et al., 2022) further extends the document similarity learning via neighborhood sampling on citation graphs by controlling the sampling margin between hard-to-learn positives and negatives. OAG-BERT (Liu et al., 2022) is an entity-augmented academic language model pre-trained with the task of masked entity prediction. Graph-empowered language models (GNN-LMs) are introduced to incorporate the relational information into the language model. For example, GraphFormers (Yang et al., 2021) is a GNN-nested Transformer architecture that insert graph neural networks between transformer layers. PATTON (Jin et al., 2023) proposes to pre-train the GraphFormers on a text-rich graph using the masked token and node prediction objectives. However, the computation cost of these GNN-LMs are significantly higher than LMs. In this work, we propose an efficient graph-enhanced PLM fine-tuning framework for author name disambiguation.

6 Conclusions and Future Work

In this paper, we study the problem of author name disambiguation with both textual and relational attributes in the documents. We propose a graph-enhanced language model – GAND to encode both information and further improve the performance of AND with a novel multi-task fine-tuning loss. Experimental result shows GAND outperforms various existing AND algorithms. Interesting future work can be extending the multi-task fine-tuning objective to more pairwise language understanding tasks with structured knowledge like question answering.

7 Limitations

High-order Interactions between Documents. In this work, our proposed framework mainly utilizes one-hop of structural information in the constructed graph. Traditional graph neural networks deal with multi-hop message passing. As we discuss in the main paper, the time and space complexity of updating the parameters of multiple hops of documents grows exponentially. One possible strategy is separating approximation of high-order propagation and feature transformation such as SGC (Wu et al., 2019) and PPRGO (Bojchevski et al., 2020). It is also possible to conduct parallel computing across multiple GPUs. While we demonstrate that a deeper graph neural networks does not provide benefits for the AND task, we leave the systematic exploration of this direction as future work.

Risks. The proposed PLM-GNN architecture and multi-task fine-tuning objective are evaluated for author name disambiguation only. Although it may be effective for other similarity modeling tasks, we do not expect this approach yields superior performance on other NLU tasks. When the corpus is a lot larger than the ones used in the paper (*e.g.*, $>$ 10M documents), the training time of our algorithm will be longer.

8 Ethics Statement

We carefully anonymized actual author information in both datasets with the unique identifier as mentioned in graph construction. For the AND data used for train and evaluation, we do not have any intentions other than studying the proposed problem. Ethical information such as gender and nationality of the authors are not used in this work.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*.

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

Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. 2020. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2464–2473.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiaoming Fan, Jianyong Wang, Xu Pu, Lizhu Zhou, and Bing Lv. 2011. On graph-based name disambiguation. *Journal of Data and Information Quality (JDIQ)*, 2(2):1–23.

Hui Han, Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. 2004. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, pages 296–305. IEEE.

Hui Han, Hongyuan Zha, and C Lee Giles. 2005. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 334–343.

Kaushal Jhawar, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. Author name disambiguation in pubmed using ensemble-based classification algorithms. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 469–470.

Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. 2023. Patton: Language model pretraining on text-rich networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Kunho Kim, Shaurya Rohatgi, and C Lee Giles. 2019. Hybrid deep pairwise classification for author name disambiguation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2369–2372.

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

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language

representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Michael Levin, Stefan Krawczyk, Steven Bethard, and Dan Jurafsky. 2012. Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology*, 63(5):1030–1047.

Xiao Liu, Da Yin, Jingnan Zheng, Xingjian Zhang, Peng Zhang, Hongxia Yang, Yuxiao Dong, and Jie Tang. 2022. Oag-bert: Towards a unified backbone language model for academic knowledge services. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3418–3428.

Gilles Louppe, Hussein T Al-Natsheh, Mateusz Susik, and Eamonn James Maguire. 2016. Ethnicity sensitive author disambiguation using semi-supervised learning. In *international conference on knowledge engineering and the semantic web*, pages 272–287. Springer.

Zhiyong Lu. 2011. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision*, pages 360–368.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. *arXiv preprint arXiv:2202.06671*.

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

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Min Song, Erin Hea-Jin Kim, and Ha Jin Kim. 2015. Exploring author name disambiguation on pubmed-scale. *Journal of informetrics*, 9(4):924–941.

Shivashankar Subramanian, Daniel King, Doug Downey, and Sergey Feldman. 2021. S2and: A benchmark and evaluation system for author name disambiguation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 170–179. IEEE.

Jie Tang, Alvis CM Fong, Bo Wang, and Jing Zhang. 2011. A unified probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):975–987.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998.

Pucktada Treeratpituk and C Lee Giles. 2009. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 39–48.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Haiwen Wang, Ruijie Wan, Chuan Wen, Shuhao Li, Yuting Jia, Weinan Zhang, and Xinbing Wang. 2020. Author name disambiguation on heterogeneous information network with adversarial representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 238–245.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810.

Yutao Zhang, Fanjin Zhang, Peiran Yao, and Jie Tang. 2018. Name disambiguation in aminer: Clustering, maintenance, and human in the loop. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1002–1011.

Zhiqiang Zhang, Chunqi Wu, Zhao Li, Juanjuan Peng, Haiyan Wu, Haiyu Song, Shengchun Deng, and Biao Wang. 2021. Author name disambiguation using multiple graph attention networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

A Appendix

A.1 Implementation Details

We implement GAND as well as all PLM baselines using the Hugging Face library (Wolf et al., 2019). The graph attention layer and neighbor sampler used in our model are implemented using torch-geometric[7]. We use AdamW as our optimizer and hyperparameter configurations are shown in Table 4.

Table 4: Hyperparameters of all compared algorithms.

parameter name	value
adam ϵ	1e-8
learning rate lr	5e-5
weight decay	1e-2
batch size B	16
hidden dimension d	768
maximal document length N	256
maximal gradient norm	1.0
neighborhood size K	5
# GNN layers L	1
# epochs	10
# random seeds	5

A.2 Evaluation Metric Details

Here, we provide the detailed calculation of Pairwise Micro-F1 and Normalized Mutual Information (NMI) in our experiments. Given a test document $d_i \in \mathcal{D}$, we denote the prediction $\hat{y}_i$ as a pair of reference identifier and cluster membership (r_i, c_i), and similarly the ground truth y_i as a pair of reference identifier and real author (r_i, a_i).

(1) Pairwise Micro-F1 is the harmonic mean of precision and recall between all predicted pairs.

$$\text{Prec} = \frac{\sum_{(i,j)\in\mathcal{S}} \mathbb{I}(r_i = r_j \wedge c_i = c_j \wedge a_i = a_j)}{\sum_{(i,j)\in\mathcal{S}} \mathbb{I}(r_i = r_j \wedge c_i = c_j)}$$
$$\text{Rec} = \frac{\sum_{(i,j)\in\mathcal{S}} \mathbb{I}(r_i = r_j \wedge c_i = c_j \wedge a_i = a_j)}{\sum_{(i,j)\in\mathcal{S}} \mathbb{I}(r_i = r_j \wedge a_i = a_j)} \quad (6)$$

where $\mathcal{S}$ is the Cartesian product of all test documents $\mathcal{S} = \mathcal{D} \times \mathcal{D}$.

(2) Normalized Mutual Information (NMI) For reference r, the NMI score is computed between ground truth and predicted clusters.

$$\text{NMI}(r) = \frac{2 \times I(Y_r; \hat{Y}_r)}{[H(Y_r) + H(\hat{Y}_r)]} \quad (7)$$

where $Y_r = \{y_i | r_i = r\}$ is the set of ground truth pairs under reference r, I is the mutual information and H is the entropy.

[7] https://pytorch-geometric.readthedocs.io/en/latest/index.html

Table 5: Performance of GAND with different backbone PLMs on MAG-CS.

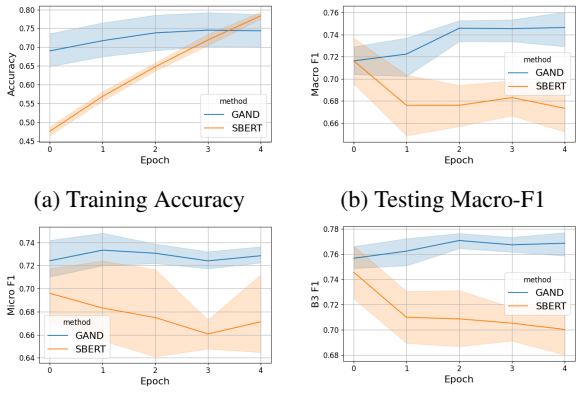
Type of PLMs.	Micro-F1	Macro-F1	B3-F1	NMI
MPNet	73.80	74.61	76.93	66.04
SPECTER	72.91	73.83	76.13	64.89
SciBERT	72.07	73.49	75.73	63.55

A.3 GAND with different backbone PLMs

In our main experiments, we use MPNET (Song et al., 2020) as our backbone PLM. Here we provide the ablation study using different PLMs using one of the random seeds in Table 5. We observe that MPNet exhibits slightly better performance compared to other standalone PLMs and when used as the backbone for our models, as shown in both this section and Table 2 of the main paper. As a result, we choose MPNet as our default backbone model.

A.4 Full Results of Training Dynamics Comparison

In Section 4.3, we compare the training dynamics of GAND and traditional contrastive fine-tuning on Macro-F1. In Table 7, we provide the performance curves on other metrics. The result is consistent with the main paper, that is, multi-task fine-tuning can simultaneously improve the training accuracy and clustering performance on test data.



(a) Training Accuracy (b) Testing Macro-F1

(c) Testing Micro-F1 (d) Testing B^3 F1

Figure 7: Comparison of Fine-Tuning PLMs and GAND.

Table 6: Result of applying Multi-FT on multiple baselines.

Method	MAG-CS				AMiner			
	Micro-F1	Macro-F1	B^3 F1	NMI	Micro-F1	Macro-F1	B^3 F1	NMI
MPNet	66.92	67.49	70.88	52.23	69.59	62.86	66.43	55.86
MPNet+Multi-FT	70.56	71.09	74.38	57.77	74.15	68.30	72.22	62.27
SGC	72.60	73.99	76.31	64.18	73.83	67.72	72.95	63.79
SGC+Multi-FT	73.72	74.21	77.05	63.20	75.52	69.84	74.65	**64.08**
GAT	73.25	73.92	76.52	63.68	72.85	67.50	72.19	62.23
GAT+Multi-FT	74.08	73.66	77.00	62.49	74.99	69.71	74.50	64.05
GAND	**75.02**	**75.94**	**77.98**	**67.23**	**75.80**	**70.53**	**74.81**	63.75

A.5 Additional Results on Applying Multi-FT

In Figure 4, we show the effectiveness of Multi-FT on MAG-CS dataset using bar plot. We also conduct the performance on AMiner and observe Multi-FT can improve the performance of PLMs significantly across two datasets and four metrics. On GNN baselines, the improvements are smaller but consistent at most times. The detailed numbers can be found in Table 6.