Weakly Supervised Multi-Label Classification of Full-Text Scientific Papers

Yu Zhang
University of Illinois at Urbana-Champaign
yuz9@illinois.edu

Bowen Jin
University of Illinois at Urbana-Champaign
bowenj4@illinois.edu

Xiusi Chen
University of California, Los Angeles
xchen@cs.ucla.edu

Yanzhen Shen
University of Illinois at Urbana-Champaign
yanzhen4@illinois.edu

Yunyi Zhang
University of Illinois at Urbana-Champaign
yzhan238@illinois.edu

Yu Meng
University of Illinois at Urbana-Champaign
yumeng5@illinois.edu

Jiawei Han
University of Illinois at Urbana-Champaign
hanj@illinois.edu

ABSTRACT

Instead of relying on human-annotated training samples to build a classifier, weakly supervised scientific paper classification aims to classify papers only using category descriptions (e.g., category names, category-indicative keywords). Existing studies on weakly supervised paper classification are less concerned with two challenges: (1) Papers should be classified into not only coarse-grained research topics but also fine-grained themes, and potentially into multiple themes, given a large and fine-grained label space; and (2) full text should be utilized to complement the paper title and abstract for classification. Moreover, instead of viewing the entire paper as a long linear sequence, one should exploit the structural information such as citation links across papers and the hierarchy of sections and paragraphs in each paper. To tackle these challenges, in this study, we propose FuTex, a framework that uses the cross-paper network structure and the in-paper hierarchy structure to classify full-text scientific papers under weak supervision. A network-aware contrastive fine-tuning module and a hierarchy-aware aggregation module are designed to leverage the two types of structural signals, respectively. Experiments on two benchmark datasets demonstrate that FuTex significantly outperforms competitive baselines and is on par with fully supervised classifiers that use 1,000 to 60,000 ground-truth training samples.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → **Classification and regression trees**.

KEYWORDS

multi-label text classification; weak supervision; scientific paper; full text

[†]Code and Datasets are available at https://github.com/yuzhimanhua/FUTEX.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599544

ACM Reference Format:
Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. 2023. Weakly Supervised Multi-Label Classification of Full-Text Scientific Papers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599544

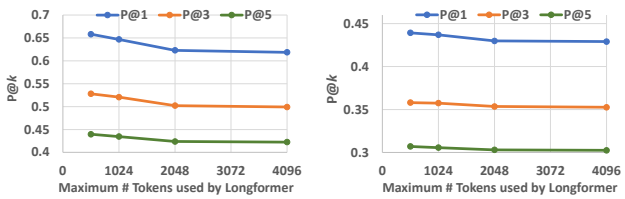
1 INTRODUCTION

Weakly supervised text classification [29, 33, 50] aims to classify text documents into a set of pre-defined categories without relying on any human-labeled training documents. Instead, the classifier seeks help from various formats of weak supervision such as category names [31, 33, 42, 50], a few category-indicative keywords [29, 63, 70], and category descriptions [69]. This setting significantly alleviates the burden of manual annotations, which is particularly helpful in some real applications such as scientific paper classification, where annotations need to be acquired from domain experts.

Although existing studies on weakly supervised text classification have applied their proposed methods to scientific paper datasets such as arXiv [32, 64] and DBLP [30, 66], they are less concerned with the following two challenges in practice.

A Large and Fine-Grained Label Space. One major goal of scientific paper classification is to help researchers track and analyze academic information and resources. To facilitate this goal, papers should be classified into not only coarse-grained research fields (e.g., "Machine Learning" and "Public Health") but also fine-grained themes (e.g., "Large Language Models" and "Deltacoronavirus"). Note that in a large and fine-grained label space, most papers are naturally relevant to multiple themes. However, most existing studies under the weakly supervised setting focus on classifying papers at a coarse level with 5 to 50 categories and assume each document is relevant to only one category (or a single path from the root to a leaf if categories form a hierarchy). As far as we know, MICoL [69] is a pioneering work that considers weakly supervised multi-label classification with more than 10,000 categories. Nevertheless, its accuracy is hampered by using limited information such as paper titles and abstracts only, which will be discussed below.

The Usage of Paper Full Texts. A paper's title and abstract, although summarized to cover its major topics, cannot capture all fine-grained aspects. For example, technique-related labels may be introduced in more detail in the "Method" section; downstream tasks may be explained in the "Experiments" section. To find more



(a) MAG-CS (b) PubMed

Figure 1: Weakly supervised classification performance of Longformer [2] on two datasets of scientific papers, MAG-CS and PubMed, used in [68]. When Longformer allows a larger input sequence length (i.e., it can take more tokens from paper full texts), the classification precision drops.

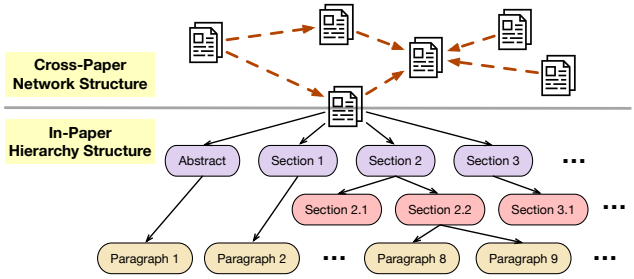


Figure 2: The cross-paper network structure and in-paper hierarchy structure associated with scientific papers.

labels relevant to a paper, it becomes necessary to leverage its full text. Intuitively, when we aim to utilize paper full texts, the first challenge is to deal with long text. Indeed, if we check the two datasets, MAG-CS and PubMed, used by MICoL [69], the average full-paper length exceeds 4,000 words. In comparison, most pre-trained language models (PLMs) such as BERT [12] and SciBERT [2] can take an input sequence with at most 512 tokens. Therefore, those PLM-based weakly supervised text classifiers [29, 33, 50, 69] cannot be directly applied to full-text scientific papers.

More importantly, we would like to argue that classifying full-text papers is beyond the problem of dealing with long text. As shown in Figure 1, we adopt Longformer [3], which can take at most 4,096 tokens, to classify papers in MAG-CS and PubMed under the weakly supervised setting. (For more experiment details such as how Longformer is used, please refer to Section 4.1.) We control the maximum input sequence length of Longformer from 512 to 4,096. Surprisingly, the more tokens Longformer takes (i.e., the more information from the full text that Longformer considers), the lower the classification precision is. This observation has two implications: First, full texts are noisy and should not be treated the same as abstracts. Abstracts should still play a leading role in classification, while full texts provide auxiliary information [11, 17]. Second, to better exploit full text, we need to consider the structures in scientific papers. The major design that enables Longformer to take a longer input sequence is to sparsify the fully connected attention in Transformer [47], where each input token only interacts with its neighbor tokens and the first several tokens in the linear sequence. However, the rich structural information prevalently available inside the full text is not fully captured by Longformer. Figure 2 shows two types of such structural information: the *cross-paper network structure* and the *in-paper hierarchy structure*. The hierarchy structure organizes sections, subsections, and paragraphs into a tree, where the parent-child relation indicates that a finer text unit is included in a coarser one. Such a structure implies which paragraphs should be jointly considered when aggregating paragraph semantics to the entire paper. Moreover, by parsing the bibliographic entries in paper full texts, one can obtain each paper's references and construct a citation network. This cross-paper network structure indicates the semantic proximity between two papers. To summarize, these two types of structures provide additional semantic signals that are not reflected in a linear text sequence.

Contributions. Being aware of the two aforementioned challenges, in this paper, we study weakly supervised multi-label text classification of full-text scientific papers. We propose FuTex, the design of which is centered around how to use the cross-paper network structure and the in-paper hierarchy structure to classify scientific papers in a large and fine-grained label space. FuTex has three major modules: (1) *Network-aware contrastive fine-tuning* aims to leverage the cross-paper network structure to fine-tune a pre-trained language model (PLM) so that it can probe fine-grained label semantics and distinguish among similar categories. (2) *Hierarchy-aware aggregation* aims to exploit the in-paper hierarchy structure to obtain the entire paper representation by aggregating from its paragraphs. With this aggregation process, a PLM does not need to deal with the full-text sequence at once. (3) *Self-training* aims to take the initial prediction (i.e., top-ranked categories according to the first two modules) as pseudo labels to train a full-text paper classifier. Then the prediction of the trained classifier can complement the initial prediction to improve the final classification results.

We conduct experiments on two datasets (both with >10,000 categories) commonly used in previous studies [56, 68, 69]. Results show that FuTex outperforms competitive baselines including scientific PLMs [2, 9, 24], weakly supervised text classifiers [57, 69], and structure-enhanced PLMs [54, 55]. Notably, on the MAG-CS dataset, our FuTex model, without any ground-truth training data, is on par with a supervised classifier [40] trained on 60,000 labeled papers. To summarize, this work makes the following contributions.

- We study the problem of weakly supervised multi-label classification of full-text scientific papers. Different from most previous studies on weakly supervised text classification, it considers a large, fine-grained label space and paper full texts.
- We propose the FuTex framework which utilizes the cross-paper network structure and the in-paper hierarchy structure associated with scientific papers to improve classification performance.
- We conduct experiments on two datasets and demonstrate the effectiveness of FuTex in comparison with competitive baselines, including those using abstracts only and those using full text.

2 PRELIMINARIES

2.1 Cross-Paper and In-Paper Structures

Cross-Paper Network Structure. Given a collection of scientific papers, we can obtain the references of each paper by parsing the bibliographic entries in its full text. If we view papers as nodes and references as directed edges, a network can be constructed. Formally, we have the definition below.

Definition 2.1. (Cross-Paper Network Structure) A scientific paper corpus $\mathcal{D}$ has a cross-paper network structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

where each node $d \in \mathcal{V}$ is a paper, and $(d_i, d_j) \in \mathcal{E}$ if and only if d_i cites d_j (i.e., d_j is a bibliographic entry in d_i).

To describe the relationship between two papers in $\mathcal{G}$, we adopt the notation of meta-paths [45, 46]. To be specific, if d_i cites d_j, we can say the two papers are connected via the meta-path "Paper $\rightarrow$ Paper" (or its abbreviation $P \rightarrow P$). Similarly, if two papers d_i and d_j share a common reference d_k (i.e., $(d_i, d_k) \in \mathcal{E}$ and $(d_j, d_k) \in \mathcal{E}$), we can say d_i and d_j are connected via the meta-path $P \rightarrow P \leftarrow P$. Intuitively, if two papers are connected via a certain meta-path, their relevant topics are more likely to overlap. Following [69], given a meta-path $\mathcal{M}$, we use $d_i \rightarrow^{\mathcal{M}} d_j$ to denote that d_i is connected to d_j via $\mathcal{M}$, and the meta-path-based neighborhood $\mathcal{N}_{\mathcal{M}}(d_i)$ is defined as $\{d_j | d_i \rightarrow^{\mathcal{M}} d_j \text{ AND } d_j \neq d_i\}$.

In-Paper Hierarchy Structure. One unique challenge we are facing in full-text paper classification is that each paper is beyond a plain text sequence (i.e., title+abstract) and contains its internal hierarchical structure of paragraphs. As shown in Figure 2, nodes representing paragraphs, subsections, sections, and the entire paper form a tree, in which the parent-child relation implies a finer text unit is entailed by a coarser one. Formally, we have the definition below.

Definition 2.2. (In-Paper Hierarchy Structure) A full-text paper d contains a hierarchical tree structure $\mathcal{T}_d$. The root of $\mathcal{T}_d$ represents the entire paper; the leaves of $\mathcal{T}_d$ are d's paragraphs $\mathcal{P}_d = \{p_{d1}, ..., p_{dM}\}$. The tree can be characterized by a mapping $\text{Child}(\cdot)$, where $\text{Child}(x)$ is the set of text units that are one level finer than x and contained in x.

Putting the two structures together, we would like to classify the nodes in a network $\mathcal{G}$. Meanwhile, each node d contains its own subcomponents that form a hierarchy $\mathcal{T}_d$. Given a paper, we need to jointly consider its subcomponents and its proximity with neighbors to infer its categories.

2.2 Problem Definition

In this paper, we study weakly supervised multi-label text classification. By "weakly supervised", we imply that we do not have any annotated training samples for any label, and the only available supervision to characterize a label is its name and several descriptive sentences [4, 69]. Figure 3 shows the name and description of the label "Deltacoronavirus" as an example. This setting is more challenging than zero-shot multi-label text classification [14, 35, 41, 52, 62] which assumes annotated documents are given for some *seen* classes and the trained classifier should be generalized to predict *unseen* classes. Under the weakly supervised setting, all classes are unseen. This setting is also called "*wild* zero-shot" in some previous studies [57, 69].

By "multi-label", we mean that each paper can be relevant to more than one label. This is a natural assumption when the label space is fine-grained and multi-faceted. For example, a COVID-19 paper can be labeled as "Infections", "Lung Diseases", "Coronavirus", and "Public Health" at the same time. This assumption makes our task more challenging than weakly supervised single-label classification [29, 33, 38, 50, 63].

To summarize, our task can be defined as follows.

Definition 2.3. (Problem Definition) Given (1) a collection of scientific papers $\mathcal{D}$ with a cross-paper network structure $\mathcal{G}$, where each paper d has its full text and hierarchy structure $\mathcal{T}_d$, and (2) a label space $\mathcal{L}$ where each label l has its name and description, our task is to predict the relevant labels $\mathcal{L}_d \subseteq \mathcal{L}$ for each $d \in \mathcal{D}$.

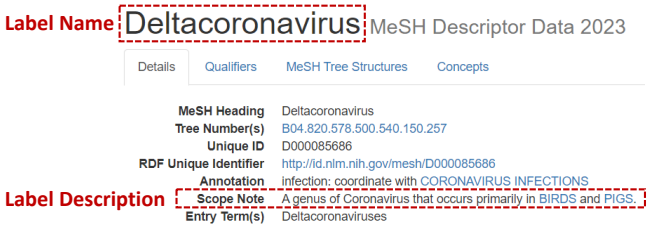


Figure 3: Name and description of the label "Deltacoronavirus" from PubMed (https://meshb.nlm.nih.gov/record/ui?ui=D000085686).

3 MODEL

One straightforward solution to our task is to pick a pre-trained language model (e.g., SciBERT [2]), use it to encode each paper's content and each label's name/description to get their embeddings, and then perform the nearest neighbor search in the embedding space. However, such an approach suffers from two drawbacks: First, unfine-tuned PLMs may not be powerful enough to detect the subtle semantic differences between two papers or two label descriptions, but fine-grained text classification, to a great extent, requires the classifier to distinguish among labels that are close to each other. Second, the entire paper is long (e.g., with ~4,000 words on average in the Semantic Scholar Open Research Corpus (S2ORC) [26]), which exceeds the maximum sequence length (e.g., 512 tokens) that a PLM can handle in most cases.

To overcome the aforementioned two drawbacks, we propose to exploit the cross-paper network structure and the in-paper hierarchy structure, which will be introduced in Sections 3.1 and 3.2, respectively. Then, in Section 3.3, we present a self-training strategy, that is, how we use initial predictions as pseudo labels to train a classifier that complements the predictions. The overview of our proposed FuTex framework is shown in Figure 4.

3.1 Network-Aware Contrastive Fine-Tuning

The first module in FuTex aims to utilize the cross-paper network structure to improve the PLM's ability to distinguish among fine-grained labels. We follow the intuition of LinkBERT [55] that if two papers are connected via certain citation-based relationships (e.g., $d_i \rightarrow^{\mathcal{M}} d_j$), then a paragraph $p_s \in d_i$ and a paragraph $p_t \in d_j$ are more likely to share fine-grained topics than two randomly picked paragraphs. In LinkBERT [55], Yasunaga et al. propose to concatenate the two "linked" paragraphs together (i.e., `[CLS]` p_s `[SEP]` p_t `[SEP]`) to perform masked token prediction and document relation prediction for language model pre-training [12]. However, in this paper, we are not aiming at training a general-purpose PLM. Instead, our model only needs to judge whether two text units are relevant to similar topics or not. Being able to do this, during inference, the model can take a paragraph and a label description as input to predict whether the paragraph is relevant to the label. To achieve this goal, following [69], we adopt a contrastive fine-tuning objective to replace the language model pre-training objectives in LinkBERT.

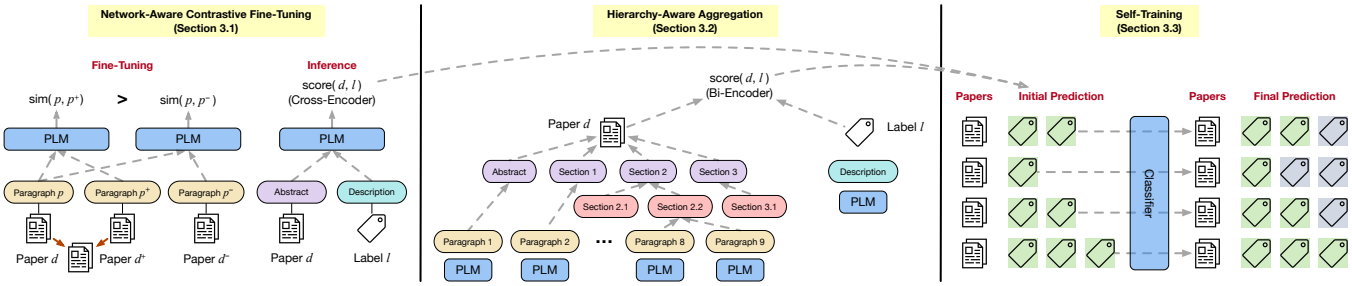


Figure 4: Overview of the FuTex framework.

To be specific, the PLM should be fine-tuned to distinguish between "linked" and "unlinked" paragraph pairs. As shown in Figure 4 (left), given three papers d, d^+, and d^-, where $d^+ \in \mathcal{N}_{\mathcal{M}}(d)$ and $d^- \notin \mathcal{N}_{\mathcal{M}}(d)$. We randomly sample three paragraphs p, p^+, and p^- from d, d^+, and d^-, respectively. The PLM aims to predict the similarity between p and p^+ as well as that between p and p^-.

$$\begin{aligned} \boldsymbol{e}_{\text{pos}} &= \text{PLM}(\texttt{[CLS]}\, p\, \texttt{[SEP]}\, p^+\, \texttt{[SEP]}), \\ \boldsymbol{e}_{\text{neg}} &= \text{PLM}(\texttt{[CLS]}\, p\, \texttt{[SEP]}\, p^-\, \texttt{[SEP]}). \end{aligned} \tag{1}$$

Here, $\boldsymbol{e}_{\text{pos}}$ and $\boldsymbol{e}_{\text{neg}}$ are the output representations of the `[CLS]` token after PLM encoding.

A linear layer is then trained to predict the scores given the output representations.

$$\text{sim}(p, p^+) = \boldsymbol{w}^\top \boldsymbol{e}_{\text{pos}}, \quad \text{sim}(p, p^-) = \boldsymbol{w}^\top \boldsymbol{e}_{\text{neg}}, \tag{2}$$

where $\boldsymbol{w}$ is a learnable vector. The PLM is fine-tuned to make $\text{sim}(p, p^+)$ larger than $\text{sim}(p, p^-)$, in which case we can adopt the contrastive loss [7].

$$\mathcal{J} = \mathop{\mathbb{E}}_{\substack{p \in d \\ p^+ \in d^+ \in \mathcal{N}_{\mathcal{M}}(d) \\ p^- \in d^- \notin \mathcal{N}_{\mathcal{M}}(d)}} \left[-\log \frac{\exp(\text{sim}(p, p^+))}{\exp(\text{sim}(p, p^+)) + \exp(\text{sim}(p, p^-))} \right]. \tag{3}$$

After contrastive fine-tuning, the PLM is used to predict the score between a paragraph and a label. For example, given a paper $d \in \mathcal{D}$ and a label $l \in \mathcal{L}$, we use a_d to denote the title+abstract of d, and we use t_l to denote the name+description of l. Then, the score can be calculated as

$$\text{sim}(a_d, t_l) = \boldsymbol{w}^\top \text{PLM}(\texttt{[CLS]}\, a_d\, \texttt{[SEP]}\, t_l\, \texttt{[SEP]}). \tag{4}$$

However, the strategy that concatenates each paragraph and each label and feeds them into one PLM (i.e., the Cross-Encoder architecture [36]) makes the inference computationally expensive because the representation of each text unit cannot be pre-computed. For example, suppose there are D papers (each of which has P paragraphs) and L labels, then we need to call the PLM $O(DPL)$ times during inference. Such a cost will prohibit us from applying the model to a large corpus and a large label space. For example, if $D = L = 10^4$ and $P = 30$, then $DPL = 3 \times 10^9$. To alleviate the cost, we adopt the following two strategies.

Adding a retrieval stage. Following [69], given a paper d, we first adopt exact name matching to retrieve a small set of candidate labels $C(d)$ from the entire label space $\mathcal{L}$. To be specific, if a label's name appears in the paper's content, it will be added as a candidate. The fine-tuned PLM is then applied as a reranker to score labels from the retrieved candidate pool.

Using Bi-Encoder for non-abstract paragraphs. As mentioned in the Introduction, the abstract and other paragraphs should not be treated equally in text classification. The abstract is highly summarized to cover the major topics of the paper, while a paragraph in the paper body may capture only one aspect and provides auxiliary topic signals. To use the more powerful tool in the most important case, we only adopt the Cross-Encoder architecture when inferring the labels of paper abstracts (i.e., Eq. (4)). For other paragraphs, we adopt the Bi-Encoder architecture [19]. To be specific, we encode each paragraph and each label separately.

$$\boldsymbol{h}_p = \text{PLM}(\texttt{[CLS]}\, p\, \texttt{[SEP]}), \quad \boldsymbol{h}_l = \text{PLM}(\texttt{[CLS]}\, t_l\, \texttt{[SEP]}). \tag{5}$$

Here, $\boldsymbol{h}_p$ and $\boldsymbol{h}_l$ are the output representations of the `[CLS]` token after PLM encoding. Then, the similarity between p and l can be computed as $\cos(\boldsymbol{h}_p, \boldsymbol{h}_l)$. One drawback of this strategy is that the paragraph and the label text cannot serve as each other's context during PLM encoding. However, the efficiency is significantly improved because we can pre-compute $\boldsymbol{h}_p$ and $\boldsymbol{h}_l$ for all paragraphs and labels. In fact, if we combine the two proposed strategies, the inference complexity will be reduced to $O(D\lambda + DP + L)$, where λ is the average number of candidate labels picked for each paper in the retrieval stage. For example, if we assume $D = L = 10^4$ and $P = \lambda = 30$, then $D\lambda + DP + L = 6.1 \times 10^5$, which is orders of magnitude smaller than $DPL = 3 \times 10^9$.

3.2 Hierarchy-Aware Aggregation

Although the similarity between each paragraph p and each label l can be computed efficiently now, we have not figured out how to calculate the score between an entire paper d and a label l. Intuitively, simply averaging all paragraph embeddings may not work well because important signals from the paper abstract and conclusion will then be buried under the great amount of content from other sections. Previously, FullMeSH [11] proposed to check 5 sections – abstract, introduction, method, result, and summary – in each full paper so as to probe relevant topics from different aspects. Inspired by their idea, we utilize the in-paper hierarchy structure $\mathcal{T}_d$ to perform embedding aggregation from paragraphs to sections, and then to the entire paper.

Given a non-leaf text unit $x \in \mathcal{T}_d$ (e.g., subsections, sections, or the entire paper), we obtain the embedding of x by aggregating the embeddings from x's children. Formally,

$$\boldsymbol{h}_x = \frac{1}{|\text{Child}(x)|} \sum_{y \in \text{Child}(x)} \boldsymbol{h}_y. \tag{6}$$

After the bottom-up aggregation, the score between the entire paper

d and the label l can be computed as

$$\text{score}_B(d, l) = \cos(\boldsymbol{h}_d, \boldsymbol{h}_l). \tag{7}$$

Here, the subscript "B" means the score is based on the Bi-Encoder architecture. Recall that we can calculate the score between d and l using Eq. (4) based on the Cross-Encoder architecture:

$$\text{score}_X(d, l) = \boldsymbol{w}^\top \text{PLM}(\texttt{[CLS]}\, a_d\, \texttt{[SEP]}\, t_l\, \texttt{[SEP]}), \tag{8}$$

where "X" stands for Cross-Encoder.

Now we adopt an ensemble ranking step to jointly consider the two scores. Given a document d, we first rank all candidate labels from $C(d)$ in descending order according to $\text{score}_B(d, l)$ and $\text{score}_X(d, l)$, respectively. In this way, each candidate label l will have two rank positions $r_B(l|d)$ and $r_X(l|d)$. Then, we calculate the mean reciprocal rank (MRR) of l.

$$\text{MRR}(l|d) = \frac{1}{r_B(l|d)} + \frac{1}{r_X(l|d)}. \tag{9}$$

Finally, candidate labels are sorted according to $\text{MRR}(l|d)$ as the reranking result.

3.3 Self-Training

The retrieval stage proposed in Section 3.1 significantly improves the efficiency of FuTex. However, it may filter out labels that do not explicitly appear in a paper but are implicitly relevant to the paper in the latent semantic space. To mitigate this issue, we present a self-training strategy to utilize paper full texts and confident predictions (based on MRR) to train a text classifier $f_{\text{class}}(\cdot)$. The classifier is then used to predict the probability that a paper is relevant to a label (not necessarily selected in the retrieval stage), and those top-ranked labels will complement our initial MRR-based predictions.

Since the major goal of this paper is not to invent a new fully supervised text classifier, we propose to use an off-the-shelf model. To leverage paper full texts and meanwhile bypass the sequence length problem, we choose a bag-of-words multi-label classifier – Parabel [40], which, according to [59], has competitive performance even compared with deep learning classifiers on benchmark datasets. Parabel represents each training document d as a $|\mathcal{W}_\mathcal{D}|$-dimensional feature vector $\boldsymbol{x}_d$, where $\mathcal{W}_\mathcal{D}$ is the vocabulary of $\mathcal{D}$. Given a word $w \in \mathcal{W}_\mathcal{D}$, its corresponding entry in $\boldsymbol{x}_d$ is the following tf–idf score:

$$x_{d,w} = \text{tf}(w, d) \cdot \text{idf}(w, \mathcal{D}), \tag{10}$$

where $\text{tf}(w, d)$ is the term frequency of w in d, and $\text{idf}(w, \mathcal{D}) = \log \frac{|\mathcal{D}|}{|\{d' \in \mathcal{D} | w \in d'\}|}$ is the inverse document frequency of w.

For each paper d, according to MRR calculated in Eq. (9), we use the top-N predicted labels as pseudo labels to train the Parabel classifier. (If d has less than N candidate labels selected in the retrieval stage, we use all of them.) Formally, the pseudo labels of d is represented as an $|\mathcal{L}|$-dimensional vector $\boldsymbol{y}_d$, where

$$y_{d,l} = \begin{cases} 1, & r_{\text{MRR}}(l|d) \le N, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Here, $r_{\text{MRR}}(l|d)$ is the rank position of l according to $\text{MRR}(l|d)$.

Given $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$ of each paper, Parabel learns a tree-based discriminative classifier $\Pr(\hat{\boldsymbol{y}}_d|\boldsymbol{x}_d)$. (For more technical details, one can refer to [40].) Based on the trained classifier, we relabel each paper d to keep those initial confident predictions while incorporating new highly-ranked labels. To be specific, the top-N predictions according to MRR remain at top-N in our final predictions; the other labels will be sorted according to Parabel's output $\Pr(\hat{y}_{dl} = 1|\boldsymbol{x}_d)$ and ranked after top-N. We use the following example to explain this process.

Table 1: Dataset Statistics

Dataset	#Papers	#Labels	#Words/ Paper	#Paragraphs/ Paper	#Labels/ Paper
MAG-CS [49]	96,718	10,909	4071.69	45.91	5.84
PubMed [28]	251,573	16,070	4901.42	33.38	8.69

Example 3.1. (Final Prediction after Self-Training) Suppose there are 5 labels $\mathcal{L} = \{A, B, C, D, E\}$ and $N = 2$. Given a paper d, assume 3 labels A, B, and C are selected in the retrieval stage and their MRR scores are 2.00, 1.00, and 0.67, respectively. Then, A and B will be the two pseudo labels of d used for training Parabel. The trained Parabel is used to classify d again and get the following scores:

l	A	B	C	D	E
$\Pr(\hat{y}_{dl} = 1\|\boldsymbol{x}_d)$	0.80	0.85	0.30	0.60	0.90

In our final prediction, A and B will still be the top 2, and the other labels will be ranked according to Parabel's prediction. Therefore, the final rank will be (A, B, E, D, C). In practice, we find this strategy achieves better classification performance than reranking all labels purely based on $\Pr(\hat{y}_{dl} = 1|\boldsymbol{x}_d)$, the result of which is (E, B, A, D, C).

The entire procedure of FuTex is summarized in Appendix A.1.

4 EXPERIMENTS

4.1 Setup

Datasets. We use two datasets, **MAG-CS** [49] and **PubMed** [28], that are widely adopted in previous studies on scientific paper classification [56, 68, 69]. Originally, MAG-CS contains ~705K papers published at 105 top computer science venues, where each paper is labeled with its related fields-of-study [43]; PubMed consists of ~899K papers published in 150 top medicine journals, where each paper is labeled with its related MeSH terms [10]. However, since the two original datasets do not have paper full texts, we try to extract full texts from S2ORC [26]. S2ORC has segmented each paper into paragraphs, marked the section that each paragraph belongs to, and parsed the bibliographic entries. We remove the paragraphs with less than 10 words from each paper. Note that not all papers in these two datasets can be found in S2ORC, and we finally obtain 96,718 full-text MAG-CS papers and 251,573 full-text PubMed papers. Because FuTex does not require any annotated training data, all these papers are used for testing. More statistics of these two datasets can be found in Table 1.

Compared Methods. We compare our FuTex model with the following baselines including scientific PLMs, structure-enhanced PLMs, and zero-shot multi-label text classification methods.

- **SciBERT [2]**[1] is a scientific PLM trained on 1.14M scientific papers from Semantic Scholar [1] using masked language modeling and next sentence prediction tasks.
- **OAG-BERT [24]**[2] is a scientific PLM trained on 120M scientific papers from the Open Academic Graph [61]. It proposes heterogeneous entity type embedding, span-aware entity masking, and

[1] https://huggingface.co/allenai/scibert_scivocab_uncased
[2] https://github.com/THUDM/OAG-BERT

Table 2: P@k and NDCG@k scores of compared methods on MAG-CS and PubMed. Bold: the highest score. *: FuTex is significantly better than this method with p-value < 0.05. **: FuTex is significantly better than this method with p-value < 0.01.

	Method	MAG-CS [49]					PubMed [28]				
		P@1	P@3	P@5	NDCG@3	NDCG@5	P@1	P@3	P@5	NDCG@3	NDCG@5
Abstract	SciBERT [2]	0.6825**	0.5525**	0.4542**	0.5990**	0.5555**	0.4503**	0.3712**	0.3193**	0.3916**	0.3595**
	OAG-BERT [24]	0.5960**	0.4860**	0.4168**	0.5258**	0.5018**	0.4142**	0.3340**	0.2902**	0.3539**	0.3265**
	LinkBERT [55]	0.6515**	0.5288**	0.4402**	0.5729**	0.5363**	0.3689**	0.3331**	0.3014**	0.3434**	0.3270**
	SPECTER [9]	0.7599**	0.5958**	0.4765**	0.6510**	0.5923**	0.5419**	0.4124**	0.3388**	0.4441**	0.3946**
	0SHOT-TC [25, 57]	0.6489**	0.5003**	0.4164**	0.5490**	0.5128**	0.5504**	0.4280**	0.3532**	0.4583**	0.4085**
	MICoL [69]	0.7658	0.6005**	0.4797**	0.6562**	0.5965**	0.5775*	0.4329**	0.3528**	0.4680**	0.4133**
Full Text	SciBERT [2]	0.6881**	0.5573**	0.4574**	0.6041**	0.5594**	0.4885**	0.4176**	0.3517**	0.4375**	0.3965**
	OAG-BERT [24]	0.5775**	0.4724**	0.4097**	0.5108**	0.4913**	0.4245**	0.3867**	0.3296**	0.4002**	0.3660**
	LinkBERT [55]	0.6654**	0.5365**	0.4447**	0.5820**	0.5429**	0.4241**	0.3919**	0.3380**	0.4041**	0.3727**
	SPECTER [9]	0.7582**	0.5965**	0.4773**	0.6512**	0.5927**	0.5400**	0.4335**	0.3575**	0.4610**	0.4112**
	Longformer [3]	0.6582**	0.5280**	0.4396**	0.5740**	0.5366**	0.4392**	0.3580**	0.3070**	0.3786**	0.3466**
	PLM+GAT [48, 54]	0.7306**	0.5760**	0.4663**	0.6285**	0.5765**	0.5140**	0.4000**	0.3374**	0.4279**	0.3870**
	GraphFormers [54]	0.7261**	0.5720**	0.4637**	0.6242**	0.5730**	0.5213**	0.4046**	0.3380**	0.4334**	0.3895**
	FuTex	**0.7692**	**0.6089**	**0.4914**	**0.6648**	**0.6099**	**0.5900**	**0.4533**	**0.3798**	**0.4869**	**0.4388**

entity-aware two-dimensional position embedding to leverage paper metadata information (e.g., venues and authors).

- **LinkBERT [55]**[3] is a structure-enhanced PLM. It uses linked Wikipedia paragraphs as context to perform masked language modeling and document relation prediction. On the PubMed dataset, we use BioLinkBERT[4], which is pre-trained on PubMed papers and performs better than LinkBERT.
- **SPECTER [9]**[5] is a structure-enhanced scientific PLM. It continues pre-training SciBERT using a citation prediction objective with 684K pairs of linked papers.

The four baselines above can be used to classify either abstracts or full-text papers. When applying them to abstracts, we directly encode each abstract and each label description to calculate the cosine similarity between the two embeddings. When applying them to full text, we follow the hierarchy-aware aggregation process in Section 3.2 and ensemble the results of using abstract only and using full text.

- **0SHOT-TC [25, 57]** is a zero-shot text classification method. It is a natural language inference (NLI) model that predicts to what extent a paper (as the premise) entails the sentence "*this document is about {label_name}.*" (as the hypothesis). Following [42], we use **RoBERTa-large-mnli** [25][6] as the NLI model.
- **MICoL [69]**[7] is a zero-shot text classification method. It proposes a metadata-induced contrastive learning technique to fine-tune a PLM. MICoL has various configurations with the model architecture and the used meta-path. According to the experimental results in [69], we choose the best-performing configuration: (Cross-Encoder, $P \rightarrow P \leftarrow P$).

The two methods above adopt the Cross-Encoder architecture. As mentioned in Section 3.1, it will be too computationally expensive to apply them to all paragraphs. Therefore, we keep their original usage on paper abstracts only.

- **Longformer [3]**[8] is a PLM dealing with long documents. It sparsifies the fully connected attention and can take 4,096 tokens at most. We adopt it to encode full-text papers by setting the maximum number of tokens as 512, 1,024, 2,048, and 4,096. As shown in the Introduction, the 512-token version performs the best, so we use it for performance comparison.
- **PLM+GAT [48, 54]** is a PLM stacked with a Graph Attention Network (GAT) layer. Each paragraph $p \in d$ is first encoded by a PLM. Then we use GAT to obtain paragraph embeddings by aggregating PLM representations of its neighbor paragraphs $p' \in d^+ \in \mathcal{N}_{\mathcal{M}}(d)$. A link prediction objective (i.e., judging whether two paragraphs are connected via meta-path $\mathcal{M}$) is then adopted to train the PLM and GAT in an end-to-end manner.
- **GraphFormers [54]**[9] is a GNN-nested PLM architecture, in which GNN layers and Transformer layers are alternately stacked. Similar to PLM+GAT, a link prediction objective is adopted to train the model so that the PLM can be enhanced by the cross-paper network structure.

The three methods above are naturally suitable for classifying paper full texts.

According to our experiments, SPECTER performs better than SciBERT, OAG-BERT, and LinkBERT. Therefore, for MICoL, PLM+GAT, GraphFormers, and FuTex, we all use SPECTER as the base PLM. Also, the meta-path $\mathcal{M}$ is set as $P \rightarrow P \leftarrow P$ for all these models for a fair comparison.

Implementation and Hyperparameters. Each label from the PubMed dataset may have multiple label names, including one canonical name and 0, 1, or several synonyms (i.e., "entry terms"). Following [69], we include l as a candidate label of d if *any* of its names appears in d's content. On both MAG-CS and PubMed, during the retrieval stage, we use each paper's title and abstract, instead of the full text, for label name matching because this yields better classification performance. During network-aware contrastive fine-tuning, when feeding the two paragraphs into a Cross-Encoder, the maximum length of each paragraph is 256 tokens. The training batch size is 8. We use the AdamW optimizer [27], warm up the

[3] https://huggingface.co/michiyasunaga/LinkBERT-base
[4] https://huggingface.co/michiyasunaga/BioLinkBERT-base
[5] https://huggingface.co/allenai/specter
[6] https://huggingface.co/roberta-large-mnli
[7] https://github.com/yuzhimanhua/MICoL
[8] https://huggingface.co/allenai/longformer-base-4096
[9] https://github.com/microsoft/GraphFormers

learning rate for the first 100 steps and then linearly decay it. The learning rate is 5e-5, the weight decay is 0.01, and ϵ = 1e-8. We sample 45,000 and 150,000 tuples of (p, p^+, p^-) from MAG-CS and PubMed, respectively, for PLM fine-tuning. During self-training, we set $N = 5$ to get pseudo labels of each paper. For the Parabel classifier[10], all parameters are set by default. Specifically, the number of trees is 3; the maximum number of labels in a leaf node is 100; the beam search width in prediction is 10. We remove words appearing in less than 5 papers when training Parabel.

Evaluation Metrics. We use two commonly adopted metrics in multi-label text classification: P@k and NDCG@k, where k = 1, 3, and 5. Given a paper d, we use $z_d \in \{0,1\}^{|\mathcal{L}|}$ to denote its ground-truth labels and rank(i) to denote the i-th ranked label in the final prediction. P@k and NDCG@k are then defined as:

$$\text{P@}k = \frac{1}{k}\sum_{i=1}^{k} z_{d,\text{rank}(i)}.$$
$$\text{DCG@}k = \sum_{i=1}^{k}\frac{z_{d,\text{rank}(i)}}{\log(i+1)},\quad \text{NDCG@}k = \frac{\text{DCG@}k}{\sum_{i=1}^{\min(k,\|z_d\|_0)}\frac{1}{\log(i+1)}}. \tag{12}$$

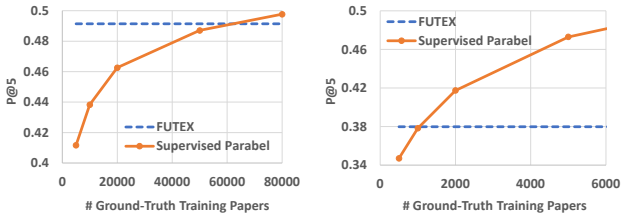
4.2 Performance Comparison

Table 2 shows P@k and NDCG@k scores of compared methods on MAG-CS and PubMed. For models with randomness (i.e., MICoL, PLM+GAT, GraphFormers, and FuTex), we run each of them 5 times with the average performance reported. Other models (i.e., SciBERT, OAG-BERT, LinkBERT, SPECTER, 0SHOT-TC, and Longformer) are deterministic according to our usage. To show statistical significance, we conduct a two-tailed t-test to compare FuTex with each baseline if the baseline has randomness, and we conduct a two-tailed Z-test to compare FuTex with each deterministic baseline. The significance level is also marked in Table 2.

From Table 2, we find that: (1) FuTex consistently and significantly outperforms all baselines. On both datasets, MICoL is a competitive baseline, possibly because it also uses citation links across papers. However, since MICoL only considers text content from paper titles and abstracts, it is not as powerful as FuTex. (2) There are four baselines (i.e., SciBERT, OAG-BERT, LinkBERT, and SPECTER) that can be used in both abstract-only and full-text settings. In most cases, a full-text variant can outperform its abstract-only counterpart. This observation validates our claim that considering the full text is beneficial to paper classification. Besides, unlike Longformer (as shown in Figure 1), our proposed hierarchy-aware aggregation strategy, which is also used by the full-text variants of the four baselines, can effectively use paper full texts. Furthermore, in a few cases, a full-text variant underperforms its abstract-only counterpart in terms of P@1 but achieves higher P@3 and P@5. This finding implies that signals extracted by hierarchy-aware aggregation from paper full texts can better help lower-ranked predictions.

Comparison with a Supervised Model. We further compare FuTex with a fully supervised multi-label text classifier. In accordance with our choice in Section 3.3, we report the performance of Parabel [40] with ground-truth training data. To be specific, for each dataset, we select 10,000 papers as testing samples and pick different numbers of training samples from the remaining papers. Figure 5 shows the P@5 score of Parabel with different numbers of ground-truth training samples in comparison with FuTex. We can observe that our FuTex model, without relying on any annotated training data, is on par Parabel that uses 60,000 and 1,000 ground-truth training samples on MAG-CS and PubMed, respectively.

[10] http://manikvarma.org/code/Parabel/download.html



(a) MAG-CS (b) PubMed

Figure 5: The P@5 score of supervised Parabel with different numbers of ground-truth training papers. Our FuTex model, without any ground-truth training samples, is on par with Parabel that uses 60,000 and 1,000 training samples on MAG-CS and PubMed, respectively.

4.3 Ablation Study

There are three major modules in FuTex: network-aware contrastive fine-tuning, hierarchy-aware aggregation, and self-training. Now we conduct an ablation study to check the contribution of each module. To facilitate this, we create three ablation versions of FuTex:

- **FuTex-NoNetwork** does not have the network-aware contrastive fine-tuning module. It directly uses SPECTER in the Bi-Encoder architecture, ensembles the results of using abstract only and using full text, and then performs self-training.
- **FuTex-NoHierarchy** does not have the hierarchy-aware aggregation module. After contrastive fine-tuning, it applies the Cross-Encoder to paper titles/abstracts only and then performs self-training.
- **FuTex-NoSelfTrain** does not have the self-training module. It uses the MRR-based ranking list obtained in Section 3.2 as the final prediction.

Table 3 demonstrates the performance of the full FuTex model and the three ablation versions. We can observe that: (1) The full FuTex model consistently and significantly outperforms the three ablation versions, indicating that all three major modules have a positive contribution to the classification performance. (2) Among the three ablation versions, FuTex-NoNetwork performs the worst in terms of P@1. This finding indicates that the cross-paper network structure is more beneficial to top-ranked predictions. By contrast, FuTex-NoSelfTrain has the lowest P@5 score on both datasets, which means that the self-training module contributes the most to lower-ranked predictions. This observation validates our claim that self-training can find more labels that are semantically relevant to each paper so as to complement the initial top-ranked categories.

4.4 Case Study

We now perform a case study to qualitatively demonstrate the effect of considering full text in paper classification. Table 4 shows two cases, one of which is from MAG-CS and the other from PubMed. For both papers, we show their text information including the title,

Table 3: P@k and NDCG@k scores of the full FuTex model and three ablation versions on MAG-CS and PubMed. Bold, *, and **: the same meaning as in Table 2.

Method	MAG-CS [49]					PubMed [28]				
	P@1	P@3	P@5	NDCG@3	NDCG@5	P@1	P@3	P@5	NDCG@3	NDCG@5
FuTex-NoNetwork	0.7601**	0.6013**	0.4859**	0.6567**	0.6031**	0.5422**	0.4410**	0.3722**	0.4676**	0.4244**
FuTex-NoHierarchy	0.7655**	0.6048**	0.4879**	0.6607**	0.6060**	0.5802**	0.4421**	0.3681**	0.4760**	0.4272**
FuTex-NoSelfTrain	0.7673**	0.6040**	0.4827**	0.6591**	0.5993**	0.5877**	0.4456**	0.3648**	0.4801**	0.4254**
FuTex	**0.7692**	**0.6089**	**0.4914**	**0.6648**	**0.6099**	**0.5900**	**0.4533**	**0.3798**	**0.4869**	**0.4388**

Table 4: Case study on MAG-CS and PubMed. Blue: Labels indicated by the paper title or abstract. Orange: Labels indicated by the paper's full text but not mentioned in the title or abstract.

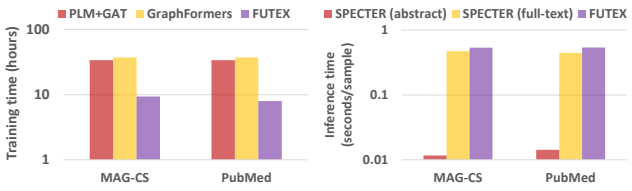
MAG-CS [49]	PubMed [28]
Title: compact modeling technique for outdoor navigation	**Title**: serum **calprotectin**: a novel diagnostic and prognostic marker in **inflammatory bowel diseases**
Abstract: abstract-in this paper, a new methodology to build compact local maps in real time for outdoor **robot** navigation is presented. the environment information is obtained from a 3-d **scanner** laser. the navigation model, which is called traversable region model, is based on a **voronoi diagram** technique ...	**Abstract**: there is an unmet need for novel blood-based biomarkers that offer timely and accurate diagnostic and prognostic testing in **inflammatory bowel diseases** (ibd). we aimed to investigate the diagnostic and prognostic utility of serum **calprotectin** (sc) in ibd ...
Full Text: this new challenge has prompted a change in robotics navigation philosophy, where **path planning** and modeling were always obtained a priori ... in general, in **mobile robot navigation**, the occupancy-based approach is one of the most commonly used methods ... **path planning** in large and outdoor environments is a complex task because there are a lot of parameters that define the traversability, for example, as follows ... when the 3-d model is defined (as the one in figs. 5 and 6 , where the terrain considered ta is represented in blue and the nta is represented in red), the free space can be extracted to build a trm for the robot navigation and **path planning** ...	**Full Text**: inflammatory bowel diseases (ibd), including **crohn's disease** (cd) and **ulcerative colitis** (uc), are chronic, debilitating inflammatory disorders of the gastrointestinal tract affecting adults and children ... a recent meta-analysis of 13 studies and 1,041 patients found that fc had a pooled **sensitivity and specificity** of 0.93 ... to determine the accuracy of blood parameter measurements as a prognostic test capable of diagnosing ibd, receiver operating characteristic (roc) analyses were performed by plotting **sensitivity** against **specificity** ...
Ground-Truth Labels: robot, voronoi diagram, motion planning, computer vision, mobile robot navigation, mobile robot, scanner, computational geometry	**Ground-Truth Labels**: leukocyte l1 antigen complex, inflammatory bowel diseases, multivariate analysis, colitis ulcerative, prognosis, kaplan meier estimate, sensitivity and specificity, humans, crohn disease, logistic models, proportional hazards models, odds ratio, area under curve
MICoL Prediction: voronoi diagram (✓), robot (✓), scanner (✓)	**MICoL Prediction**: inflammatory bowel diseases (✓), leukocyte l1 antigen complex (✓)
FuTex Prediction: voronoi diagram (✓), robot (✓), scanner (✓), motion planning (✓), mobile robot navigation (✓)	**FuTex Prediction**: inflammatory bowel diseases (✓), leukocyte l1 antigen complex (✓), crohn disease (✓), colitis ulcerative (✓), sensitivity and specificity (✓)

abstract, and excerpts from full text. We also show their ground-truth labels, labels predicted by FuTex, and labels predicted by MICoL (which is the most competitive baseline using titles and abstracts only). We mark a label as blue if it (or a semantically similar term) appears in the paper title or abstract; we mark a label as orange if it does not appear in the title/abstract but is mentioned in full text.

In the MAG-CS case, three labels "voronoi diagram", "robot", and "scanner" explicitly appear in the paper abstract, and they are correctly predicted by both MICoL and FuTex. However, MICoL misses labels such as "motion planning" and "mobile robot navigation", which are not mentioned in the title/abstract. In fact, the term "outdoor robot navigation" in the abstract may imply the paper's relevance to "mobile robot navigation", but MICoL does not build the connection between them. After the paper's full text is exploited, "mobile robot navigation" completely appears in the content, and the term "path planning", which is semantically close to the label "motion planning", is repeatedly mentioned. As a result, both labels are accurately captured by FuTex.

In the PubMed case, MICoL successfully predicts the ground-truth labels "inflammatory bowel diseases" and "leukocyte l1 antigen complex" (whose synonym is "calprotectin"[11]) indicated by the title/abstract. However, MICoL fails to predict labels such as "crohn disease", "colitis ulcerative", and "sensitivity and specificity". As shown in Table 4, hints to these labels can be found in the full text. Note that these labels are indeed relevant to the paper rather than just being mentioned: Crohn's disease and ulcerative colitis are two types of inflammatory bowel diseases studied in the paper, and the paper extensively discusses the sensitivity and specificity of predicting these diseases. By leveraging the paper's full text, FuTex accurately picks these labels.

[11] According to https://meshb-prev.nlm.nih.gov/record/ui?ui=D039841, "Calprotectin" is an entry term of "Leukocyte L1 Antigen Complex".



(a) Training time (b) Inference time

Figure 6: Training and inference time of FuTex and representative baselines on MAG-CS and PubMed.

4.5 Efficiency

We now analyze the training and inference time of FuTex on MAG-CS and PubMed. To be specific, we compare the training time of FuTex (the network-aware contrastive learning step) with that of PLM+GAT and GraphFormers, which also use network signals for training. For a fair comparison, we run each model on an NVIDIA

RTX A6000 GPU, and we train the three models for the same number of epochs on each dataset (i.e., 20 epochs on MAG-CS and 5 epochs on PubMed). As for the inference stage, we notice that the major factor which affects each model's inference efficiency is whether it is used for abstracts only or full texts. If we fix this factor, different BERT-based baselines will have similar inference efficiency because they have similar model sizes and architectures. Therefore, we choose SPECTER as a representative for comparison and report its inference time (per testing sample) when used for abstracts only and full texts. The results are demonstrated in Figure 6.

From Figure 6(a), we observe that FuTex has much less training time than PLM+GAT and GraphFormers. From Figure 6(b), we find that the inference efficiency of FuTex is on par with SPECTER (full-text). In comparison with SPECTER (abstract), FuTex and SPECTER (full-text) need significantly more time to run because about 30 to 45 times more paragraphs are considered.

5 RELATED WORK

Weakly Supervised Text Classification. Weakly supervised text classification aims to assign relevant label(s) to each document without any human-annotated training samples provided. The common formats of weak supervision include label names [33], a small set of category-indicative keywords [29], and label descriptions [69]. Technically, earlier methods mainly utilize Explicit Semantic Analysis [6, 44], Latent Dirichlet Allocation [8, 21, 22], and context-free word embeddings [31, 32]. Inspired by the success of BERT [12] in a wide spectrum of text mining tasks, recent studies start to exploit the power of PLMs in weakly supervised text classification. For example, ConWea [29] uses BERT to disambiguate the provided keywords and retrieve more category-indicative words for pseudo training data collection; LOTClass [33] leverages one BERT encoder to perform masked language modeling for finding more indicative words and another BERT to perform classification; X-Class [50] uses BERT representations of words to perform category-aware clustering and then aligns documents to categories; LIME [38] adopts BART-large-MNLI [20] and prompts to predict pseudo labels of each document and then uses BERT for self-training. However, all the aforementioned methods focus on a relatively small label space (≤ 50 categories in most cases) and assume each document is relevant to one category only (or a single path from the root to a leaf category in the hierarchical classification setting). In contrast, FuTex studies larger and more fine-grained label spaces (e.g., $> 10,000$ categories) where each document is relevant to multiple labels in most cases.

Zero-Shot Multi-Label Text Classification. In general, similar to "weakly supervised", the term "zero-shot" also implies that the classifier does not need any annotated samples. However, in large-scale or extreme multi-label text classification, "zero-shot" is interpreted in different ways in the existing literature. According to [57], the *restrictive* zero-shot setting assumes training samples are given for some seen classes and the classifier aims to predict unseen classes [5, 14, 34, 35, 41, 52, 62], which is different from our "weakly supervised" setting; the *wild* zero-shot setting does not assume any seen classes and the classifier needs to make predictions without relying on any annotations [42, 57, 69]. For example, TaxoClass [42] uses RoBERT-large-MNLI [25] to convert text classification to an entailment task; MICoL [69] proposes a metadata-induced contrastive learning method to fine-tune SciBERT [2]. However, existing *wild* zero-shot classifiers still view each document as a linear sequence of paragraphs, thus cannot be directly applied to full-text paper classification due to the maximum length limit of PLMs. In comparison, FuTex exploits the cross-paper and in-paper structures of scientific literature.

Scientific Paper Classification. Classifying scientific papers is a common evaluation task in both text mining (e.g., [2, 9, 24, 65]) and network mining (e.g., [13, 15, 18, 60]) studies. However, most studies consider coarse-grained paper classification only (e.g., ≤ 50 categories). To satisfy users' fine-grained interests, Zhang et al. [67–69] and Ye et al. [56] propose to use paper metadata to perform large-scale multi-label paper classification. In the biomedical domain, MeSH indexing [23, 39, 53] can also be cast as a multi-label text classification task to tag PubMed papers with fine-grained medical subject headings. Nevertheless, these studies focus on using the paper title and abstract only. To the best of our knowledge, FullMeSH [11] and BERTMeSH [58] are two representative studies making use of paper full texts. However, they adopt a fully supervised setting and are not directly applicable to our task.

6 CONCLUSIONS AND FUTURE WORK

We present FuTex, a multi-label scientific paper classifier that relies on label names and descriptions as the only supervision and does not require any human-annotated training data. FuTex exploits paper full texts and consists of three modules: the contrastive learning module leverages the cross-paper citation network structure; the semantic aggregation module uses the in-paper hierarchy structure of sections, subsections, and paragraphs; the self-training module trains a full-text classifier using pseudo labels to complement the initial predictions. Experiments on two datasets demonstrate the superiority of FuTex over competitive weakly supervised baselines and show that FuTex is on par with fully supervised classifiers with thousands of ground-truth training samples. An ablation study validates the usefulness of all three proposed modules. A case study shows that FuTex can effectively extract signals from full text to predict labels not indicated by the paper title or abstract.

Interesting future directions include (1) how to leverage other in-paper structural signals, such as the relationship between paragraphs and figures/tables, to further improve the classification performance and (2) how to leverage large language models (e.g., GPT-4 [37]) for weakly supervised fine-grained paper classification, where one needs to tackle the maximum input length limit given the paper full text and tens of thousands of label names.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their valuable and insightful feedback. Research was supported in part by the IBM-Illinois Discovery Accelerator Institute, US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] 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. In *NAACL-HLT'18*. 84–91.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP'19*. 3615–3620.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[4] Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *ICML'20*. 1371–1382.

[5] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-Scale Multi-Label Text Classification on EU Legislation. In *ACL'19*. 6314–6322.

[6] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *AAAI'08*. 830–835.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML'20*. 1597–1607.

[8] Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless text classification with descriptive LDA. In *AAAI'15*.

[9] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL'20*. 2270–2282.

[10] Margaret H Coletti and Howard L Bleich. 2001. Medical subject headings used to search the biomedical literature. *JAMIA* 8, 4 (2001), 317–323.

[11] Suyang Dai, Ronghui You, Zhiyong Lu, Xiaodi Huang, Hiroshi Mamitsuka, and Shanfeng Zhu. 2020. FullMeSH: improving large-scale MeSH indexing with full text. *Bioinformatics* 36, 5 (2020), 1533–1541.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT'19*. 4171–4186.

[13] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD'17*. 135–144.

[14] Nilesh Gupta, Sakina Bohra, Yashoteja Prabhu, Saurabh Purohit, and Manik Varma. 2021. Generalized Zero-Shot Extreme Multi-label Learning. In *KDD'21*. 527–535.

[15] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW'20*. 2704–2710.

[16] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *KDD'16*. 935–944.

[17] Antonio J Jimeno-Yepes, Laura Plaza, James G Mork, Alan R Aronson, and Alberto Díaz. 2013. MeSH indexing based on automatically generated summaries. *BMC Bioinformatics* 14, 1 (2013), 1–12.

[18] Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. 2023. Patton: Language Model Pretraining on Text-Rich Networks. *arXiv preprint arXiv:2305.12268* (2023).

[19] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP'20*. 6769–6781.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL'20*. 7871–7880.

[21] Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective document labeling with very few seed words: A topic model approach. In *CIKM'16*. 85–94.

[22] Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. 2018. Dataless text classification: A topic modeling approach with document manifold. In *CIKM'18*. 973–982.

[23] Ke Liu, Shengwen Peng, Junqiu Wu, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2015. MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. *Bioinformatics* 31, 12 (2015), i339–i347.

[24] 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 *KDD'22*. 3418–3428.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[26] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *ACL'20*. 4969–4983.

[27] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR'19*.

[28] Zhiyong Lu. 2011. PubMed and beyond: a survey of web tools for searching biomedical literature. *Database* 2011 (2011).

[29] Dheeraj Mekala and Jingbo Shang. 2020. Contextualized Weak Supervision for Text Classification. In *ACL'20*. 323–333.

[30] Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. META: Metadata-Empowered Weak Supervision for Text Classification. In *EMNLP'20*. 8351–8361.

[31] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *CIKM'18*. 983–992.

[32] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *AAAI'19*. 6826–6833.

[33] Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text Classification Using Label Names Only: A Language Model Self-Training Approach. In *EMNLP'20*. 9006–9017.

[34] Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in text: Learning document, label, and word representations jointly. In *AAAI'16*. 1948–1954.

[35] Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2015. Predicting unseen labels using label hierarchies in large-scale multi-label learning. In *ECML-PKDD'15*. 102–118.

[36] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424* (2019).

[37] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[38] Seongmin Park and Jihwa Lee. 2022. LIME: Weakly-Supervised Text Classification without Seeds. In *COLING'22*. 1083–1088.

[39] Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. DeepMeSH: deep semantic representation for improving large-scale MeSH indexing. *Bioinformatics* 32, 12 (2016), i70–i79.

[40] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *WWW'18*. 993–1002.

[41] Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *EMNLP'18*, Vol. 2018. 3132.

[42] Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical Multi-Label Text Classification Using Only Class Names. In *NAACL-HLT'21*. 4239–4249.

[43] Zhihong Shen, Hao Ma, and Kuansan Wang. 2018. A Web-scale system for scientific knowledge exploration. In *ACL'18 System Demonstrations*. 87–92.

[44] Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *AAAI'14*. 1579–1585.

[45] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. 2011. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM'11*. 121–128.

[46] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB* 4, 11 (2011), 992–1003.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS'17*. 5998–6008.

[48] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR'18*.

[49] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.

[50] Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-Class: Text Classification with Extremely Weak Supervision. In *NAACL-HLT'21*. 3043–3053.

[51] Tong Wei, Wei-Wei Tu, Yu-Feng Li, and Guo-Ping Yang. 2021. Towards Robust Prediction on Tail Labels. In *KDD'21*. 1812–1820.

[52] Yuanhao Xiong, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit Dhillon. 2022. Extreme Zero-Shot Learning for Extreme Text Classification. In *NAACL'22*. 5455–5468.

[53] Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, and Aidong Zhang. 2019. MeSHProbeNet: a self-attentive probe net for MeSH indexing. *Bioinformatics* 35, 19 (2019), 3794–3802.

[54] 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. In *NeurIPS'21*. 28798–28810.

[55] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining Language Models with Document Links. In *ACL'22*. 8003–8016.

[56] Chenchen Ye, Linhai Zhang, Yulan He, Deyu Zhou, and Jie Wu. 2021. Beyond Text: Incorporating Metadata and Label Structure for Multi-Label Document Classification using Heterogeneous Graphs. In *EMNLP'21*. 3162–3171.

[57] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *EMNLP'19*. 3905–3914.

[58] Ronghui You, Yuxuan Liu, Hiroshi Mamitsuka, and Shanfeng Zhu. 2021. BERTMeSH: deep contextual representation learning for large-scale high-performance MeSH indexing with full text. *Bioinformatics* 37, 5 (2021), 684–692.

[59] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *NeurIPS'19* (2019), 5820–5830.

[60] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD'19*. 793–803.

[61] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019. Oag: Toward linking large-scale heterogeneous entity graphs. In *KDD'19*. 2585–2595.
[62] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating Semantic Knowledge to Tackle Zero-shot Text Classification. In *NAACL-HLT'19*. 1031–1040.
[63] Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised Text Classification Based on Keyword Graph. In *EMNLP'21*. 2803–2813.
[64] Yu Zhang, Xiusi Chen, Yu Meng, and Jiawei Han. 2021. Hierarchical Metadata-Aware Document Categorization under Weak Supervision. In *WSDM'21*. 770–778.
[65] Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023. Pre-training Multi-task Contrastive Learning Models for Scientific Literature Understanding. *arXiv preprint arXiv:2305.14232* (2023).
[66] Yu Zhang, Shweta Garg, Yu Meng, Xiusi Chen, and Jiawei Han. 2022. Motifclass: Weakly supervised text classification with higher-order metadata information. In *WSDM'22*. 1357–1367.
[67] Yu Zhang, Bowen Jin, Qi Zhu, Yu Meng, and Jiawei Han. 2023. The Effect of Metadata on Scientific Literature Tagging: A Cross-Field Cross-Model Study. In *WWW'23*. 1626–1637.
[68] Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. MATCH: Metadata-Aware Text Classification in A Large Hierarchy. In *WWW'21*. 3246–3257.
[69] Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, Junheng Hao, Ye-Yi Wang, Kuansan Wang, and Jiawei Han. 2022. Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification. In *WWW'22*. 3162–3173.
[70] Yu Zhang, Frank F. Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. 2019. HiGitClass: Keyword-Driven Hierarchical Classification of GitHub Repositories. In *ICDM'19*. 876–885.

A APPENDIX

A.1 The Entire Procedure of FuTex

We summarize the entire procedure of FuTex in Algorithm 1.

Algorithm 1: FuTex

Input: A collection of unlabeled scientific papers $\mathcal{D}$; the cross-paper network structure $\mathcal{G}$; each paper d's full text and its in-paper hierarchy structure $\mathcal{T}_d$; a label space $\mathcal{L}$; each label l's name and description.
Output: The relevant labels $\mathcal{L}_d \subseteq \mathcal{L}$ of each paper $d \in \mathcal{D}$.

Retrieve a small set of candidate labels $C(d) \subseteq \mathcal{L}$ for each paper using lexical matching;
// Network-Aware Contrastive Fine-Tuning;
Fine-tune a PLM using the contrastive loss in Eq. (3);
for $d \in \mathcal{D}$ **do**
  for $l \in C(d)$ **do**
    $\text{score}_X(d, l) \leftarrow$ Eq. (8);
// Hierarchy-Aware Aggregation;
for $d \in \mathcal{D}$ **do**
  for $p \in d$ **do**
    $\boldsymbol{h}_p \leftarrow$ Eq. (5);
  $\boldsymbol{h}_d \leftarrow$ Eq. (6) according to the hierarchy $\mathcal{T}_d$;
for $l \in C(d)$ **do**
  $\boldsymbol{h}_l \leftarrow$ Eq. (5);
for $d \in \mathcal{D}$ **do**
  for $l \in C(d)$ **do**
    $\text{score}_B(d, l) \leftarrow$ Eq. (7);
$\text{MRR}(l|d) \leftarrow$ Eq. (9);
// Self-Training;
for $d \in \mathcal{D}$ **do**
  $\boldsymbol{x}_d \leftarrow$ Eq. (10);
  $\boldsymbol{y}_d \leftarrow$ Eq. (11);
Train a Parabel classifier $\Pr(\hat{\boldsymbol{y}}_d|\boldsymbol{x}_d)$ using $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$;
Get the final ranking list following Example 3.1;
Return $\mathcal{L}_d = \{\text{top-}k \text{ ranked labels of } d\}$;

A.2 Performance on Infrequent Labels

A large and fine-grained label space typically implies a long-tailed label distribution, where most categories are associated with only a few documents. In many real applications, it is desirable to predict more tail labels. For example, in scientific paper classification, predicting a paper is relevant to "Lagrangian Support Vector Machine" is more informative than saying the paper the relevant to "Machine Learning". To promote prediction of tail labels, recent studies [14, 16, 51, 59, 69] propose to use propensity-based P@k (i.e., PSP@k) and propensity-based NDCG@k (i.e., PSN@k) as evaluation metrics. PSP@k and PSN@k are formally defined as follows.

$$\frac{1}{p_l} = 1 + C(N_l + B)^{-A}, \quad \text{PSP@}k = \frac{1}{k}\sum_{i=1}^{k}\frac{z_{d,\text{rank}(i)}}{p_{\text{rank}(i)}}.$$

$$\text{PSDCG@}k = \sum_{i=1}^{k}\frac{z_{d,\text{rank}(i)}}{p_{\text{rank}(i)}\log(i+1)}, \quad \text{PSN@}k = \frac{\text{PSDCG@}k}{\sum_{i=1}^{\min(k,||z_d||_0)}\frac{1}{\log(i+1)}}. \tag{13}$$

The intuition behind these two metrics is to give a higher reward to a model if it predicts an infrequent label correctly. In Eq. (13), $\frac{1}{p_l}$ is such a reward; N_l is number of papers relevant to l in the whole dataset $\mathcal{D}$; $A, B, C > 0$ are constants. In this way, the less frequent a label is, the higher reward a model can get when predicting it correctly. PSP@k and PSN@k scores can be viewed as a reward-weighted version of P@k and NDCG@k, respectively. Following previous studies [16, 51, 59, 69], we set $A = 0.55$, $B = 1.5$, and $C = (\log|\mathcal{D}| - 1)(B+1)^A$. By definition, we have PSP@1 ≡ PSN@1 if each paper has at least one ground-truth label.

Table 5 shows PSP@k and PSN@k scores of FuTex and competitive baselines on MAG-CS and PubMed. From Table 5, we can observe that: (1) FuTex consistently and significantly outperforms all baselines except MICoL. (2) When comparing with MICoL, FuTex has lower PSP@1 and PSN@3 but higher PSP@3, PSP@5, and PSN@5. The only statistically significant gap between FuTex and MICoL is the gap of PSP@5. This echos our finding from Table 2 that considering paper full texts is more beneficial to lower-ranked predictions. One possible reason why FuTex underperforms MICoL in terms of PSP@1 and PSN@3 is that FuTex ensembles the predictions of a Cross-Encoder (Eq. (8)) and a Bi-Encoder (Eq. (7)) while MICoL is solely based on a Cross-Encoder according to our usage. In fact, as shown in [69], labels predicted by the Cross-Encoder architecture are more infrequent than those by the Bi-Encoder.

A.3 Performance on a Small Dataset

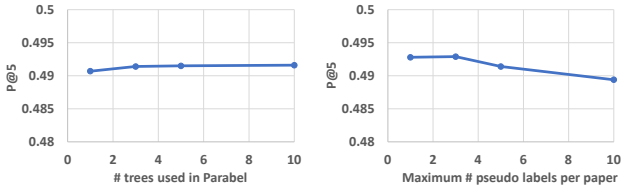
MAG-CS and PubMed have nearly 97K and 252K papers, respectively, which can provide rich self-supervision during contrastive learning and self-training. We now examine the performance of FuTex on a small dataset and check if it can still outperform competitive baselines. To facilitate this, we adopt the Art dataset from the MAPLE benchmark [67]. These Art papers are labeled with 1,990 categories at different granularities (e.g., "classics", "popular music", and "rhetorical criticism"), and we manage to find 328 of them from S2ORC [26] to obtain full texts. The performance of FuTex and competitive baselines on these Art papers are demonstrated in Table 6. We can observe that FuTex performs the best in terms of P@3, P@5, NDCG@3, and NDCG@5. For P@1, FuTex

Table 5: PSP@k and PSN@k scores of compared methods on MAG-CS and PubMed. Bold, *, and **: the same meaning as in Table 2.

Method	MAG-CS [49]					PubMed [28]				
	PSP@1	PSP@3	PSP@5	PSN@3	PSN@5	PSP@1	PSP@3	PSP@5	PSN@3	PSN@5
SPECTER [9] (abstract)	0.5222**	0.5529**	0.5511**	0.5448**	0.5450**	0.3712**	0.3577**	0.3427**	0.3617**	0.3519**
0SHOT-TC [25, 57]	0.4417**	0.4648**	0.4824**	0.4584**	0.4696**	0.3410**	0.3395**	0.3324**	0.3401**	0.3357**
MICoL [69]	**0.5419**	0.5675	0.5604**	**0.5610**	0.5579	**0.4196**	0.3872	0.3625**	**0.3964**	0.3798
SPECTER [9] (full text)	0.5180**	0.5521**	0.5510**	0.5431**	0.5438**	0.3622**	0.3657**	0.3530**	0.3654**	0.3577**
PLM+GAT [48, 54]	0.4920**	0.5335**	0.5398**	0.5225**	0.5278**	0.3337**	0.3364**	0.3324**	0.3359**	0.3335**
GraphFormers [54]	0.4937**	0.5324**	0.5380**	0.5222**	0.5270**	0.3460**	0.3454**	0.3374**	0.3459**	0.3409**
FuTex	0.5320	**0.5682**	**0.5678**	0.5588	**0.5600**	0.4071	**0.3882**	**0.3764**	0.3933	**0.3853**

Table 6: P@k and NDCG@k scores of compared methods on the Art dataset. Bold, *, and **: the same meaning as in Table 2.

Method	Art [67]				
	P@1	P@3	P@5	NDCG@3	NDCG@5
SPECTER [9] (abstract)	0.4146**	0.2693**	0.1890**	0.3452**	0.3349**
0SHOT-TC [25, 57]	**0.4573**	0.2724**	0.1927**	0.3559**	0.3479**
MICoL [69]	0.4305	0.2699**	0.1933**	0.3510**	0.3447**
SPECTER [9] (full text)	0.4238**	0.2713**	0.1921**	0.3510**	0.3432**
FuTex	0.4347	**0.3049**	**0.2211**	**0.3852**	**0.3820**



(a) (b)

Figure 7: Parameter sensitivity analysis on MAG-CS. (a) Effect of the number of trees used in the Parabel classifier. (b) Effect of the maximum number of pseudo labels per paper used for self-training.

is second to 0SHOT-TC. This observation implies that even if the dataset is small (which may limit the power of contrastive learning and self-training), FuTex still works effectively.

A.4 Hyperparameter Study

We study the effect of two major hyperparameters in FuTex: the number of trees used in the Parabel classifier and the maximum number of pseudo labels per paper used for self-training (i.e., N). The P@5 scores of FuTex on MAG-CS with different hyperparameter values in $\{1, 3, 5, 10\}$ are plotted in Figure 7. We can find that the performance of FuTex is not quite sensitive to the two hyperparameters. Indeed, all P@5 scores shown in Figure 7 outperform those of all baselines in Table 2.