LaFiCMIL: Rethinking Large File Classification from the Perspective of Correlated Multiple Instance Learning

Anonymous ACL submission

Abstract

Transfomer-based models have significantly advanced natural language processing, in particular the performance in text classification tasks. Nevertheless, these models face challenges in processing large files, primarily due to their input constraints, which are generally restricted to hundreds or thousands of tokens. Attempts to address this issue in existing models usually consist in extracting only a fraction of the essential information from lengthy inputs, while often incurring high computational costs due to their complex architectures. In this work, we address the challenge of classifying large files from the perspective of correlated multiple instance learning. We introduce LaFiCMIL, a method specifically designed for large file classification. LaFiCMIL is optimized for efficient operation on a single GPU, making it a versatile solution for binary, multi-class, and multi-label classification tasks. We conducted extensive experiments using seven diverse and comprehensive benchmark datasets to assess LaFiCMIL's effectiveness. By integrating BERT for feature extraction, LaFiCMIL demonstrates exceptional performance, setting new benchmarks across all datasets. A notable achievement of our approach is its ability to scale BERT to handle nearly 20 000 tokens while operating on a single GPU with 32GB of memory. This efficiency, coupled with its state-of-the-art performance, highlights LaFiCMIL's potential as a groundbreaking approach in the field of large file classification.

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP), entailing the assignment of suitable label(s) to specific input texts (Kowsari et al., 2019; Premasiri et al., 2023). This process is crucial across various domains, including sentiment analysis (Dang et al., 2020), fake news detection (Kumar et al., 2020), and offensive language identification (Ranasinghe and Zampieri, 2020), among others. Recent years have seen the emergence of self-attention-based models like Transformer (Vaswani et al., 2017), GPT (Radford et al., 2018, 2019), and the BERT family (Devlin et al., 2018; Feng et al., 2020; Sun et al., 2023), which have established state-of-the-art benchmarks in text classification tasks. However, the challenge of processing very long documents remains a significant obstacle, largely due to the high computational requirements of these models when facing extremely large number of tokens.

There are mainly two types of solutions in the literature to address long token sequences: ① extending the input length limit using a sparse attention mechanism, such as Longformer (Beltagy et al., 2020), and ② dividing long documents into segments and recurrently processing the Transformer-based segment representations, such as RMT (Bulatov et al., 2022, 2023). Nevertheless, they either struggle with handling extremely long sequences, as with Longformer, or suffer from information loss across the recurrent processing of segments, as with RMT.

In this work, we leverage Multiple Instance Learning (MIL) to tackle the problem of large file classification. MIL deals with a bag of instances for which only a single bag-level label is assigned, while instance-level labels remain unknown. Furthermore, the number of instances in each bag is undetermined, necessitating a flexible MIL model capable of accommodating input bags with varying instance counts. Recently, MIL has been successfully applied to computer vision problems, particularly whole slide image classification (Shao et al., 2021; Zhang et al., 2022). This inspired us to leverage MIL to address the large file classification problem.

We introduce **LaFiCMIL**, a simple yet effective **La**rge **Fi**le **C**lassification approach based on correlated **M**ultiple **I**nstance **L**earning. On the one hand, as proven in Theorem 1 (cf., Section 3), a

MIL score function for a bag classification task can be approximated by a series of sub-functions of the instances. This inspires us to split a large file into smaller chunks and extract their features separately using BERT. On the other hand, we aim to guide the model to learn high-level overall features from all instances, rather than deriving the final bag prediction from instance predictions based on a simplistic learned projection matrix. In addition, in contrast to the basic version of MIL (Ilse et al., 2018), where instances within the same bag exhibit neither dependency nor ordering among one another, we claim that the small chunks from the same large file are correlated in some way (e.g., semantic dependencies in paragraphs). This implies that the presence or absence of a positive instance in a bag can be influenced by the other instances contained within the same bag. As a result, relying on our computationally efficient LaFiAttention layer, our approach is capable of **efficiently** extracting **correlations** among **all chunks** as additional information to boost classification performance.

In our evaluation, LaFiCMIL consistently achieved new state-of-the-art performance across all seven benchmark datasets, especially when tested with long documents in the evaluation sets. A notable highlight is LaFiCMIL's performance on the full test set of the Paired Book Summary dataset, where it demonstrated a significant 4.41 percentage point improvement. This dataset is especially challenging as it contains the highest proportion of long documents, exceeding 75%. Furthermore, LaFiCMIL also distinguished itself by having the fastest training process compared to other baseline models.

The contributions of our study are as follows:

- We introduce, LaFiCMIL, a novel approach for large file classification from the perspective of correlated multiple instance learning.
- The training of LaFiCMIL is super efficient, which requires only 1.86× training time than the original BERT, but is able to handle 39× longer sequence on a single GPU.
- We perform a comprehensive evaluation, illustrating that LaFiCMIL achieves new state-of-the-art performance across all seven benchmark datasets.
- We share the datasets and source code to the community at: https://anonymous.4open.science/r/LaFiCMIL-ARR-666P

2 Related Work

2.1 Large File Classification

In recent years, significant efforts have been made to alleviate the input limit of Transformer-based models to handle different types of large files. One notable example is Longformer (Beltagy et al., 2020), which extends the limit to 4096 tokens using a sparse attention mechanism (Zaheer et al., 2020). CogLTX (Ding et al., 2020) chooses to identify key sentences through a trained judge model. Alternatively, ToBERT (Pappagari et al., 2019) and RMT (Bulatov et al., 2022, 2023) segment long documents into fragments and then aggregate or recurrently process their BERT-based representations. Recently, two simple BERT-based methods proposed in (Park et al., 2022) achieved state-of-the-art performance on several datasets for long document classification. Specifically, BERT+Random selects random sentences up to 512 tokens to augment the first 512 tokens. BERT+TextRank augments the first 512 tokens with a second set of 512 tokens obtained via TextRank (Mihalcea and Tarau, 2004). They also provide a comprehensive evaluation to compare the relative efficacy of various baselines on diverse datasets, which revealed that no single approach consistently outperforms others across all six benchmark datasets, encompassing different classification tasks such as binary (Kiesel et al., 2019), multi-class (Lang, 1995), and multi-label classification (Bamman and Smith, 2013; Chalkidis et al., 2019).

One potential reason for the limited performance of existing approaches is that they do not fully leverage the information available in large files, resulting in only partial essential information being captured. In this paper, we explore the possibility of utilizing the complete information from large files to improve the performance of various classification tasks.

2.2 Multiple Instance Learning

Multiple Instance Learning (MIL) has attracted increasing research interests and applications in recent years. The application scenarios of MIL span across various domains (Ji et al., 2020; Song et al., 2019; Hebbar et al., 2021), but the most prominent one is Medical Imaging and Diagnosis. Particularly, there has been a growing trend towards developing MIL algorithms for medical whole slide image analysis (Kanavati et al., 2020; Xu et al., 2019).

These MIL models can generally be categorized

into two groups based on whether the final bag predictions are derived directly from instance predictions (Feng and Zhou, 2017; Lerousseau et al., 2020; Lu et al., 2021a; Sharma et al., 2021) or from aggregated instance features (Li et al., 2021; Shao et al., 2021; Zhang et al., 2022). For the first group, bag predictions are typically achieved through either average pooling or maximum pooling. In contrast, the second group learns a high-level representation of a bag and constructs a classifier on top of this bag representation for bag-level predictions. Although instance-level probability pooling is simple and straightforward, empirical evidence has demonstrated that it is less effective than its bag embedding counterpart (Wang et al., 2018; Shao et al., 2021).

Furthermore, the fundamental assumption of Multiple Instance Learning (MIL) postulates that instances within a bag are independent of each other, a supposition that may not hold true in practical applications. Consequently, some researchers have endeavored to explore scenarios wherein instances within a bag exhibit correlations or dependencies, a concept referred to as Correlated Multiple Instance Learning (c-MIL) (Zhou et al., 2009; Zhang, 2021; Shao et al., 2021). This suggests that the presence or absence of a positive instance in a bag could be affected by other instances in the same bag. Nevertheless, applying c-MIL to solve large file classification problems beyond whole slide image classification remains under-explored.

3 Technical Preliminaries

In this section, we describe several essential technical preliminaries, which underpin and inform the development of LaFiCMIL. We first present a pair of theorems that substantiate the foundation of our approach, the fundamental principles of Correlated Multiple Instance Learning (c-MIL).

Theorem 1. *Suppose $S : \chi \to \mathbb{R}$ is a continuous set function w.r.t Hausdorff distance*[1] *$d_H(.,.)$. $\forall \varepsilon > 0$, for any invertible map $P : \chi \to \mathbb{R}^n$, $\exists$ function σ and g, such that for any set $X \in \chi$:*

$$|S(X) - g(P_{X\in\chi}\{\sigma(x) : x \in X\})| < \varepsilon \tag{1}$$

The proof of Theorem 1 can be found in (Shao et al., 2021). From this theorem, we can conclude that a Hausdorff continuous **set function** $S(X)$ can be arbitrarily approximated by a function in the form $g(P_{X\in\chi}\{\sigma(x) : x \in X\})$. This insight can be applied to MIL, as the mathematical definition of **sets** in the theorem is equivalent to that of **bags** in MIL framework. Consequently, the theorem provides a foundation for approximating bag-level predictions in MIL using instance-level features.

Theorem 2. *The instances in the bag are represented by random variables $\theta_1, \theta_2, ..., \theta_n$, the information entropy of the bag under the correlation assumption can be expressed as $H(\theta_1, \theta_2, ..., \theta_n)$, and the information entropy of the bag under the i.i.d. (independent and identical distribution) assumption can be expressed as $\sum_{t=1}^{n} H(\theta_t)$, then we have:*

$$\begin{aligned} H(\theta_1, \theta_2, ..., \theta_n) &= \sum_{t=2}^{n} H(\theta_t|\theta_1, \theta_2, ..., \theta_{t-1}) + H(\theta_1) \\ &\leq \sum_{t=1}^{n} H(\theta_t) \end{aligned} \tag{2}$$

The proof of Theorem 2 can be found in (Shao et al., 2021). This theorem demonstrates that the information entropy of a bag under the correlation assumption is smaller than the information entropy of a bag under the i.i.d. assumption. The lower information entropy in Correlated Multiple Instance Learning (c-MIL) suggests reduced uncertainty and the potential to provide more valuable information for bag classification tasks. In Section 4.1, we introduce c-MIL, and in Section 4.2, we derive the efficient LaFiCMIL, which learns and exploits correlations among instances to address large file classification problem.

In the remainder of this section, we present the necessary preliminaries for our efficient attention layer inspired by the Nyströmformer (Xiong et al., 2021), referred to as LaFiAttention in the following discussion, which performs as a sub-function within our proposed LaFiCMIL.

In the original Transformer (Vaswani et al., 2017), an input sequence of n tokens of dimensions d, $X \in \mathbf{R}^{n\times d}$, is projected using three matrices $W_Q \in \mathbf{R}^{n\times d_q}$, $W_K \in \mathbf{R}^{n\times d_k}$, and $W_V \in \mathbf{R}^{n\times d_v}$, referred as query, key, and value respectively with $d_k = d_q$. The outputs Q, K, V are calculated as

$$Q = XW_Q,\ K = XW_K,\ V = XW_V \tag{3}$$

Therefore, the self-attention can be written as:

$$D(Q, K, V) = SV = softmax(\frac{QK^T}{\sqrt{d_q}})V \tag{4}$$

Then, the softmax matrix S used in self-attention can be written as

[1] https://en.wikipedia.org/wiki/Hausdorff_distance

$$S = softmax(\frac{QK^T}{\sqrt{d_q}}) = \begin{bmatrix} A_S & B_S \\ F_S & C_S \end{bmatrix} \quad (5)$$

where $A_S \in \mathbf{R}^{m\times m}$, $B_S \in \mathbf{R}^{m\times(n-m)}$, $F_S \in \mathbf{R}^{(n-m)\times m}$, $C_S \in \mathbf{R}^{(n-m)\times(n-m)}$, and $m < n$.

In order to **reduce** the memory and time **complexity** from $O(n^2)$ to $O(n)$, LaFiAttention approximates S by

$$\hat{S} = softmax(\frac{Q\widetilde{K}^T}{\sqrt{d_q}})A_S^+ softmax(\frac{\widetilde{Q}K^T}{\sqrt{d_q}}), \quad (6)$$

where $\widetilde{Q} = [\widetilde{q_1}; ...; \widetilde{q_m}] \in \mathbf{R}^{m\times d_q}$ and $\widetilde{K} = [\widetilde{k_1}; ...; \widetilde{k_m}] \in \mathbf{R}^{m\times d_q}$ are the selected landmarks for inputs $Q = [q_1; ...; q_n]$ and $K = [k_1; ...; k_n]$, A_S^+ is the Moore-Penrose inverse[2] of A_S.

Lemma 1. *For $A_S \in \mathbf{R}^{m\times m}$, the sequence $\{Z_j\}_{j=0}^{j=\infty}$ generated by (Razavi et al., 2014),*

$$Z_{j+1} = \frac{1}{4}Z_j(13I - A_S Z_j(15I - A_S Z_j(7I - A_S Z_j))) \quad (7)$$

converges to Moore-Penrose inverse A_S^+ in the third-order with initial approximation Z_0 satisfying $\|A_S A_S^+ - A_S Z_0\| < 1$.

LaFiAttention approximates A_S^+ by Z^* with Lemma 1. Following the empirical choice from (Xiong et al., 2021), we run 6 iterations in order to achieve a good approximation of the pseudoinverse. Then, the softmax matrix S used in self-attention is approximated by

$$\hat{S} = softmax(\frac{Q\widetilde{K}^T}{\sqrt{d_q}})Z^* softmax(\frac{\widetilde{Q}K^T}{\sqrt{d_q}}). \quad (8)$$

4 Approach

In this section, we first introduce customized c-MIL for large file classification and then provide technical details about our LaFiCMIL approach.

4.1 Correlated Multiple Instance Learning

Unlike traditional supervised classification, which predicts labels for individual instances, Multiple Instance Learning (MIL) predicts bag-level labels for bags of instances. Typically, individual instance labels within each bag exist but inaccessible, and the number of instances in different bags may vary.

In the basic MIL concept (Ilse et al., 2018), instances in a bag are independent and unordered. However, correlations may exist among instances within a bag, where the presence or absence of a positive instance can be influenced by other instances. In fact, when formulating large file classification as a MIL problem, correlations among instances can be found due to the presence of semantic dependencies between paragraphs. According to Theorem 2, these correlations can be exploited to reduce uncertainty in prediction. In other words, this relationship can be leveraged as additional information to boost the performance of long document classification tasks. We provide the mathematical definition of the Correlated Multiple Instance Learning (c-MIL) below.

c-MIL Formulation

Here, we consider a binary classification task of c-MIL as an example. Given a bag (i.e., a large file) X_i composed of instances (i.e., small chunks) $\{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$, for $i = 1, ..., N$, that exhibit dependency or ordering among each other. The bag-level label is Y_i, yet the instance-level labels $\{y_{i,1}, y_{i,2}, ..., y_{i,n}\}$ are not accessible. Then, a binary classification of c-MIL can be defined as:

$$Y_i = \begin{cases} 0, & if \sum y_{i,j} = 0 \quad y_{i,j} \in \{0,1\}, j = 1, ..., n \\ 1, & otherwise \end{cases} \quad (9)$$

$$\hat{Y}_i = S(X_i), \quad (10)$$

where S is a scoring function, and $\hat{Y}$ is the predicted score. N is the total number of bags, and n is the number of instances in the ith bag. The number n generally varies for different bags.

4.2 LaFiCMIL

According to Theorem 1, we leverage Multi-layer Perceptron (Rumelhart et al., 1986), BERT (Devlin et al., 2018), LaFiAttention Layer and Layer Normalization (Ba et al., 2016) as **sub-functions to approximate** the c-MIL score function S defined in Equation 10.

Given a set of bags $\{X_1, ..., X_N\}$, where each bag X_i contains multiple instances $\{x_{i,1}, ..., x_{i,n}\}$, a bag label Y_i, and a randomly initialized category vector $x_{i,category}$. The goal is to learn the maps: $\mathbb{X} \to \mathbb{T} \to \gamma$, where $\mathbb{X}$ is the bag space, $\mathbb{T}$ is the transformer space and γ is the label space. The map of $\mathbb{X} \to \mathbb{T}$ can be defined as:

$$\begin{aligned} X_i^0 =& [x_{i,category}; f(x_{i,1}); ...; f(x_{i,n})] \\ &+ E_{pos}, \quad X_i^0, E_{pos} \in \mathbb{R}^{(n+1)\times d} \end{aligned} \quad (11)$$

$$Q^l = X_i^{l-1}W_Q, \quad K^l = X_i^{l-1}W_K, \quad V^l = X_i^{l-1}W_V, \quad l = 1, ..., L \quad (12)$$

[2] https://en.wikipedia.org/wiki/Moore-Penrose_inverse

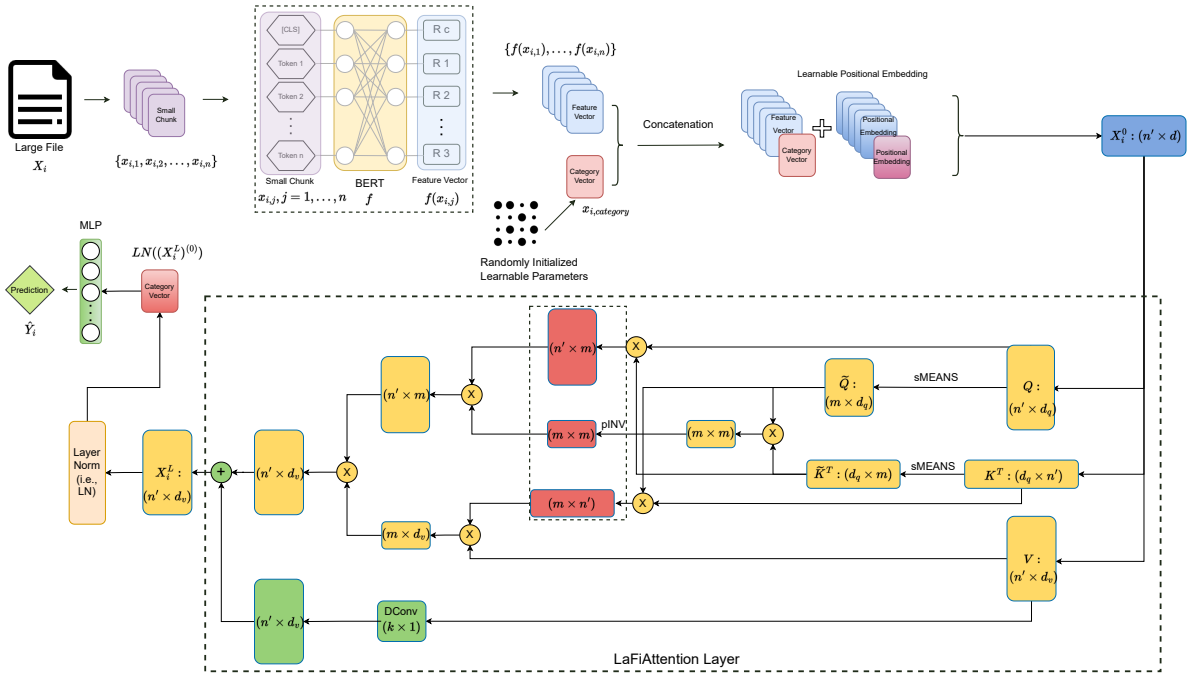

Figure 1: The LaFiCMIL framework processes large files for classification. Initially, document chunks are transformed into embedding vectors using BERT. A learnable category vector is then concatenated to these embeddings to form an augmented bag X_i^0 with $n' = n + 1$ instances. The LaFiAttention layer captures the inter-instance correlations within X_i^0. Operations within this layer, such as matrix multiplication ($\times$) and addition ($+$), are specified alongside the variable names and matrix dimensions. Key processes include sMEANS for landmark selections similar to (Shen et al., 2018), pINV for pseudoinverse approximation, and DConv for depth-wise convolution. Classification is completed by passing the learned category vector through a fully connected layer.

where function f is approximated by a pre-trained BERT model, E_{pos} is the Positional Embedding , and L is the number of MSA block.

$$head = LaFiSA(Q^l, K^l, V^l) = softmax(\frac{Q^l(\widetilde{K}^l)^T}{\sqrt{d_q}})Z^{*l}softmax(\frac{\widetilde{Q}^l(K^l)^T}{\sqrt{d_q}})V^l, \quad (13)$$

$$MSA(Q^l, K^l, V^l) = Concat(head_1, ..., head_h)W_O, \quad (14)$$

$$X_i^l = MSA(LN(X_i^{l-1})) + X_i^{l-1}, \quad l = 1, ..., L \quad (15)$$

where $W_O \in \mathbb{R}^{hd_v \times d}$, $head \in \mathbb{R}^{(n+1) \times d_v}$, $LaFiSA$ denotes the approximated Self-attention layer by Nyström method (Baker, 1977) according to Equation 8, h is the number of head in each MSA block, and Layer Normalization(LN) is applied before each MSA block.

The map of $\mathbb{T} \to \gamma$ can be simply defined as:

$$Y_i = MLP(LN((X_i^L)^{(0)})), \quad (16)$$

where $(X_i^L)^{(0)}$ represents the learned category vector, and MLP means Multi-layer Perceptron (i.e., fully connected layer).

From the above formulation, we can find that the most important part is to efficiently learn the map from bag space $\mathbb{X}$ to Transformer space $\mathbb{T}$. As illustrated in Figure 1, this map is approximated by a series of sub-functions which are approximated by various neural layers. The overall process is pre-

Algorithm 1: LaFiCMIL processing flow

Input: A set of bags (i.e., long documents) $\{X_1, ..., X_N\}$, a feature extraction function f (i.e., BERT), a randomly initialized learnable category vector $x_{i,category}$.

Output:

for X_i in $\{X_1, ..., X_N\}$ **do**
  divide the i^{th} bag into instances: $\{x_{i,1}; ...; x_{i,n}\} \leftarrow X_i$
  $X_i^0 \leftarrow [x_{i,category}; f(x_{i,1}); ...; f(x_{i,n})]$
  $E_{pos} \leftarrow Positional\_Embedding(X_i^o)$
  $X_i^o \leftarrow X_i^o + E_{pos}$
  parallelly computing MSA blocks: $l = 1, ..., L$
    $Q^l \leftarrow X_i^{l-1}W_Q, K^l \leftarrow X_i^{l-1}W_K, V^l \leftarrow X_i^{l-1}W_V$
    Compute landmarks from input Q^l and landmarks
    from input K^l, $\widetilde{Q}^l$ and $\widetilde{K}^l$ as the matrix form;
    Compute $\widetilde{F} \leftarrow softmax(\frac{Q^l(\widetilde{K}^l)^T}{\sqrt{d_q}})$;
    Compute $\widetilde{B} \leftarrow softmax(\frac{\widetilde{Q}^l(K^l)^T}{\sqrt{d_q}})$;
    Compute $\widetilde{A} \leftarrow softmax(\frac{\widetilde{Q}^l(\widetilde{K}^l)^T}{\sqrt{d_q}})$+;
    $\hat{S} \leftarrow \widetilde{F} \times \widetilde{A} \times \widetilde{B}$
    $head \leftarrow \hat{S}V^l$
    $MSA^l \leftarrow Concat(head_1, ..., head_h)W^o$
  for l in $\{1, ..., L\}$ **do**
    $X_i^l \leftarrow MSA(LN(X_i^{l-1})) + X_i^{l-1}$
  end for
  $\hat{Y}_i \leftarrow MLP(LN((X_i^L)^{(0)}))$,
end for
Final predictions for documents $\{X_1, ..., X_N\}$ are $\{\hat{Y}_1, ..., \hat{Y}_N\}$

Table 1: Statistics on the datasets. # BERT Tokens indicates the average token number obtained via the BERT tokenizer. % Long Docs means the proportion of documents exceeding 512 BERT tokens.

Dataset	Type	# Total	# Train	# Val	# Test	# Labels	# BERT Tokens	% Long Docs
Hyperpartisan	binary	645	516	64	65	2	744.18±677.87	53.49
20NewsGroups	multi-class	18 846	10 182	1132	7532	20	368.83±783.84	14.71
Book Summary	multi-label	12 788	10 230	1279	1279	227	574.31±659.56	38.46
-Paired	multi-label	6393	5115	639	639	227	1148.62±933.97	75.54
EURLEX-57K	multi-label	57 000	45 000	6000	6000	4271	707.99±538.69	51.3
-Inverted	multi-label	57 000	45 000	6000	6000	4271	707.99±538.69	51.3
Devign	binary	27 318	21 854	2732	2732	2	615.46±41 917.54	39.76

sented in Algorithm 1 and can be summarized as follows: given a large file, we use a BERT model to generate the representations of the divided chunks (i.e., instances in the concept of c-MIL). Then, we initialize a **learnable** category vector that follows a normal distribution and has the same shape as each instance. By considering the category vector as an additional instance, we learn the correlation between each instance using LaFiAttention layer. With the help of the attention mechanism, the category vector exchanges information with each chunk and extracts necessary features for large file classification. Finally, the category vector is fed into a fully connected layer to finalize the classification task.

5 Experiments

5.1 Experimental Setup

Datasets. To ensure a fair comparison with baselines, we adopt the same benchmark datasets utilized in the state-of-the-arts for long document classification (Park et al., 2022). We first evaluate LaFiCMIL on these six benchmark datasets: ① Hyperpartisan (Kiesel et al., 2019), a compact dataset encompassing 645 documents, designed for *binary classification*. ② 20NewsGroups (Lang, 1995), comprising 20 balanced categories and 11 846 documents.③ CMU Book Summary (Bamman and Smith, 2013), tailored for *multi-label classification*, contains 12 788 documents and 227 genre labels. ④ Paired Book Summary (Park et al., 2022), formulated by combining pairs of documents from the CMU Book Summary dataset, features longer documents. ⑤ EURLEX-57K (Chalkidis et al., 2019), a substantial *multi-label classification* dataset consisting of 57 000 EU legal documents and 4271 available labels. ⑥ Inverted EURLEX-57K (Park et al., 2022), a modified version of EURLEX-57K dataset in which the order of sections is inverted, ensuring that core information appears towards the end of the document. To better assess our method's capability on handling extremely longer sequences, we also include a C programming language dataset Devign (Zhou et al., 2019) for code defect detection, in which the long documents have **extremely more tokens** than the other six datasets.Table 1 offers detailed insights into the datasets, including metrics like average, maximum, and minimum token counts, along with the percentage of large documents, among other aspects. For a detailed description of each dataset, please refer to the comprehensive explanations in the Appendix A.1.

Implementation Details. We split a long text document into chunks (i.e., c-MIL instances), and follow the standard BERT input length (i.e., 512 tokens) for each chunk. To ensure a fair comparison, in line with the baseline models in (Park et al., 2022) and RMT (Bulatov et al., 2022), we employ an uncased base BERT (Devlin et al., 2018) as the feature extractor. Note that the batch size is set to 1 since we treat all instances in each single long document as a mini-batch to efficiently generate the feature vectors in parallel. Therefore, the actual batch size varies depending on the number of instances in the long document. We construct LaFiAttention layer with eight attention heads. With these settings, 100% of large documents from all six benchmark datasets can be fully processed using a single Tesla V-100 GPU with 32GB of memory on an NVIDIA DGX Station. As a result, the average inference time (0.026s) of each mini-batch is almost the same as BERT (0.022s). During training, the Adam optimizer (Kingma and Ba, 2014) is leveraged. As for the loss function, it varies depending on specific classification task. Following the baseline work (Park et al., 2022), we use sigmoid and binary cross entropy for binary and multi-label classification, and softmax and cross entropy loss for multi-class classification. For Hyperpartisan (Kiesel et al., 2019), Book Summary (Bamman

and Smith, 2013) and EURLEX-57K (Chalkidis et al., 2019), the learning rate `5e-6` is adopted. We find that the learning rate `5e-7` is more suitable for 20NewsGroups (Lang, 1995). We fine-tune the model for 20, 40, 60, and 100 epochs on Hyperpartisan, 20NewsGroups, EURLEX-57K, and Book Summary, respectively. We provide the experimental setup for code defect detection in the Appendix A.3.

Table 2: Performance metrics on only long documents in test set. The highest score in each column is bolded and underlined, while the second highest score is only bolded. The subsequent tables of this task are organized in a consistent manner.

Model	Hyperpartisan	20News	EURLEX	-Inverted	Book	-Paired
BERT	88.00	86.09	66.76	62.88	60.56	52.23
-TextRank	85.63	85.55	66.56	64.22	61.76	56.24
-Random	83.50	**86.18**	**67.03**	**64.31**	**62.34**	56.77
Longformer	**93.17**	85.50	44.66	47.00	59.66	**58.85**
ToBERT	86.50	-	61.85	59.50	61.38	58.17
CogLTX	91.91	86.07	61.95	63.00	60.71	55.74
RMT	90.04	83.62	64.16	63.21	60.62	58.27
LaFiCMIL	**95.00**	**87.49**	**67.28**	**65.04**	**65.41**	**63.03**

Table 3: Performance metrics on full test set.

Model	Hyperpartisan	20News	EURLEX	-Inverted	Book	-Paired
BERT	92.00	84.79	73.09	70.53	58.18	52.24
-TextRank	91.15	84.99	72.87	71.30	58.94	55.99
-Random	89.23	84.65	**73.22**	**71.47**	**59.36**	56.58
Longformer	**95.69**	83.39	54.53	56.47	56.53	**57.76**
ToBERT	89.54	**85.52**	67.57	67.31	58.16	57.08
CogLTX	94.77	84.62	70.13	70.80	58.27	55.91
RMT	94.34	82.87	71.46	70.99	57.30	56.95
LaFiCMIL	**96.92**	**85.07**	**73.72**	**72.03**	**61.34**	**62.17**

Evaluation Metrics. We evaluate the performance of LaFiCMIL using the same metrics as those employed in the baseline works (Park et al., 2022; Hanif and Maffeis, 2022). We report the accuracy (%) for both binary and multi-class classification. We use micro-F1 (%) for multi-label classification, which is based on summing up the individual true positives, false positives, and false negatives for each category. We report the detection accuracy (%) for code defect detection.

6 Experimental Results

In this section, we present and analyze the performance of the proposed LaFiCMIL in long document classification. We first discuss the overall performance, followed by an computational efficiency analysis and an ablation study on the core concepts of LaFiCMIL.

6.1 Overall Performance

Our experimental results reveal a phenomenon similar to (Park et al., 2022) in that no existing approach consistently outperforms the others across all benchmark datasets. However, as shown in Table 2, our LaFiCMIL establishes new state-of-the-art performance on all benchmark datasets when considering only long documents in the test set. Here, we define a long document as one containing at least two chunks (i.e., exceeding 512 BERT tokens). As shown in Table 3, we also achieve new state-of-the-art performance on five out of six benchmark datasets when considering the full data (i.e., a mix of long and short documents) in the test set. Particularly, we significantly improve the state-of-the-art score from 57.76% to 62.17% on the Paired Book Summary dataset, which contains the highest proportion of long documents (i.e., more than 75%). In contrast, we fail to achieve the best performance on 20NewsGroups, as the proportion of long documents in this dataset is very small (only 14.71%); thus, our improvement on **long** documents (as shown in Table 2) cannot dominate the overall performance on the entire dataset. This phenomenon is consistent with our motivation that the more large files (containing at least two chunks) present in the dataset, the more correlations LaFiCMIL can extract to boost classification performance.

Given that 100% of long documents from the six NLP benchmark datasets can be fully processed, we conduct an additional evaluation of LaFiCMIL's ability to process **extremely long** sequences, based on the code defect detection dataset Devign. Our findings reveal that LaFiCMIL is capable of handling inputs of up to nearly 20 000 tokens when utilizing CodeBERT (Feng et al., 2020) and VulBERTa (Hanif and Maffeis, 2022) as feature extractors on a **single GPU** setup. This capability allows for 99.92% of the code files in the Devign dataset to be processed in their entirety. Concurrently, as demonstrated in Table 4, LaFiCMIL enhances the performance of both CodeBERT and VulBERTa, establishing a new state-of-the-art in accuracy over the evaluated baselines. Please find a detailed analysis on this task in the Appendix A.3.

6.2 Computational Efficiency Analysis

In this section, we provide a comprehensive analysis of computational efficiency outlined in Table 5. All models were evaluated on a single GPU

Table 4: Accuracy (%) comparison of different models on the C programming language dataset for code defect detection. The highest accuracy score is bolded and underlined and the base model results are only bolded.

RoBERTa	CodeBERT	Code2vec	PLBART	VulBERTa	CodeBERT+ LaFiCMIL	VulBERTa+ LaFiCMIL
61.05	**62.08**	62.48	63.18	**64.27**	63.43	**64.53**

Table 5: Runtime and memory requirements of each model, **relative to BERT**, based on experiments on the Hyperpartisan dataset. Training and inference time were measured and compared in seconds per epoch. GPU memory requirement is in GB.

Model	Train Time	Inference Time	GPU Memory
BERT	1.00	1.00	<16
-TextRank	1.96	1.96	16
-Random	1.98	2.00	16
Longformer	12.05	11.92	32
ToBERT	1.19	1.70	32
CogLTX	104.52	12.53	<16
RMT	2.95	2.87	32
LaFiCMIL	1.86	1.18	32

with 32GB of memory using the Hyperpartisan dataset. LaFiCMIL performs distinctly in this context, demonstrating a runtime nearly on par with BERT. The balance between high computational efficiency and advanced classification capability illustrates LaFiCMIL's exceptional capability to efficiently process long documents without significant computational overhead.

6.3 Ablation Study

To gain a comprehensive understanding of the efficacy of each core concept in our approach (namely, BERT, LaFiAttention, and c-MIL), we conduct an ablation study. This study aims to evaluate the classification performance of LaFiCMIL when each concept is systematically removed, allowing us to evaluate their individual contributions.

Without a feature extractor, any approach would be ineffective. Thus, when BERT is removed, the LaFiAttention layer must assume the role of feature extractor instead of c-MIL. This would result in the disappearance of the c-MIL mechanism, and the approach can now only take the first chunk as input, transforming it into a basic attention-based classifier. As might be expected, the absence of the BERT concept leads to the worst performance across all datasets among the three variants, as shown in the first row of Table 6. If excluding the LaFiAttention concept, c-MIL devolves into a standard MIL, for which we employ the widely accepted Attention-MIL (Ilse et al., 2018). The results of this setting are presented in the second row of Table 6. Given that this variant can still process all chunks of a lengthy document, it performs best among all three variants on the four datasets with the largest number and longest length of documents. When the c-MIL concept is removed, the LaFiAttention layer will also be absent as it executes c-MIL which is no longer needed, leaving only BERT. Due to its restriction to process only the first chunk as input, this variant fails to achieve the best results on the four datasets with a preponderance of long documents. Finally, upon comparing the three variants with the full LaFiCMIL, shown in the fourth row of Table 6, it becomes evident that the exclusion of any concept significantly weakens performances across all datasets.

Table 6: Concept ablation study on long documents in test set. "wo" means "LaFiCMIL **w**ith**o**ut".

Model	Hyperpartisan	20News	EURLEX	-Inverted	Book	-Paired
wo BERT	85.00	53.92	60.54	54.14	50.11	46.61
wo LaFiAttn	87.50	84.97	**66.82**	**64.89**	**62.50**	**60.13**
wo c-MIL	**88.00**	**86.09**	66.76	62.88	60.56	52.23
LaFiCMIL	**95.00**	**87.22**	**67.28**	**65.04**	**65.41**	**63.03**

Furthermore, we observe some interesting results regarding **chunk positional embedding**. Basically, its effectiveness depends on different datasets. We perform an ablation study on chunk positional embedding, which is provided in the Appendix A.2.

7 Conclusion

We propose LaFiCMIL, a large file classification approach based on correlated multiple instance learning. Our method treats large document chunks as c-MIL instances, enabling feature extraction for classification from correlated chunks without substantial information loss. Experimental results demonstrate that our approach significantly outperforms the state-of-the-art baselines across multiple benchmark datasets in terms of both efficiency and accuracy. Our work provides a new perspective for addressing the large document classification problem.

8 Limitations

While LaFiCMIL has demonstrated remarkable results in three different types of long document classification tasks (i.e., binary, multi-class, and multi-label classification), its applicability to other tasks involving lengthy sequences remains to be explored. Methodological enhancements may be

needed to broaden its capabilities, and comprehensive experimentation is essential for validation. Additionally, while our method has proven effective with BERT family models as feature extractors, its efficacy with larger-scale models, particularly Large Language Models (LLMs), in processing extremely long input sequences merits further exploration. However, these aspects fall beyond the scope of this current study. We intend to investigate these areas in our future research endeavors.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Christopher TH Baker. 1977. *The numerical treatment of integral equations*. Oxford University Press.

David Bamman and Noah Smith. 2013. New alignment methods for discriminative book summarization. *arXiv preprint arXiv:1305.1319*.

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

Aydar Bulatov, Yuri Kuratov, and Mikhail S Burtsev. 2023. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on eu legislation. *arXiv preprint arXiv:1906.02192*.

Nhan Cach Dang, María N Moreno-García, and Fernando De la Prieta. 2020. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

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*.

Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Cogltx: Applying bert to long texts. In *NeurIPS*.

Ji Feng and Zhi-Hua Zhou. 2017. Deep miml network. In *AAAI*.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. Codebert: A pre-trained model for programming and natural languages. In *Findings of EMNLP*.

Hazim Hanif and Sergio Maffeis. 2022. Vulberta: Simplified source code pre-training for vulnerability detection. *arXiv preprint arXiv:2205.12424*.

R Hebbar, P Papadopoulos, R Reyes, A F Danvers, A J Polsinelli, S Moseley, D Sbarra, M R Mehl, and S Narayanan. 2021. Deep multiple instance learning for foreground speech localization in ambient audio from wearable devices. *Audio, Speech, and Music Processing*.

Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *ICML*.

Yunjie Ji, Hao Liu, Bolei He, Xinyan Xiao, Hua Wu, and Yanhua Yu. 2020. Diversified multiple instance learning for document-level multi-aspect sentiment classification. In *EMNLP*.

Fahdi Kanavati, Gouji Toyokawa, Seiya Momosaki, Michael Rambeau, Yuka Kozuma, Fumihiro Shoji, Koji Yamazaki, Sadanori Takeo, Osamu Iizuka, and Masayuki Tsuneki. 2020. Weakly-supervised learning for lung carcinoma classification using deep learning. *Scientific reports*.

J Kiesel, M Mestre, R Shukla, E Vincent, P Adineh, D Corney, B Stein, and M Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *13th International Workshop on Semantic Evaluation*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Sachin Kumar, Rohan Asthana, Shashwat Upadhyay, Nidhi Upreti, and Mohammad Akbar. 2020. Fake news detection using deep learning models: A novel approach. *Transactions on Emerging Telecommunications Technologies*, 31(2):e3767.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339.

M Lerousseau, M Vakalopoulou, M Classe, J Adam, E Battistella, A Carré, T Estienne, T Henry, E Deutsch, and N Paragios. 2020. Weakly supervised multiple instance learning histopathological tumor segmentation. In *MICCAI*.

Bin Li, Yin Li, and Kevin W Eliceiri. 2021. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *CVPR*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. 2021a. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, et al. 2021b. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *IEEE ASRU*.

Hyunji Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient classification of long documents using transformers. In *ACL*.

Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2023. Can model fusing help transformers in long document classification? an empirical study. *arXiv preprint arXiv:2307.09532*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. *arXiv preprint arXiv:2010.05324*.

M Kafaei Razavi, Asghar Kerayechian, Mortaza Gachpazan, and Stanford Shateyi. 2014. A new iterative method for finding approximate inverses of complex matrices. In *Abstract and Applied Analysis*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, et al. 2021. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. In *NeurIPS*.

Yash Sharma, Aman Shrivastava, Lubaina Ehsan, Christopher A Moskaluk, Sana Syed, and Donald Brown. 2021. Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification. In *Medical Imaging with Deep Learning*.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.

K Song, L Bing, W Gao, J Lin, L Zhao, J Wang, C Sun, X Liu, and Q Zhang. 2019. Using customer service dialogues for satisfaction analysis with context-assisted multiple instance learning. In *EMNLP*.

Tiezhu Sun, Kevin Allix, Kisub Kim, Xin Zhou, Dongsun Kim, David Lo, Tegawendé F Bissyandé, and Jacques Klein. 2023. Dexbert: Effective, task-agnostic and fine-grained representation learning of android bytecode. *IEEE Transactions on Software Engineering*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, L Jones, A N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

X Wang, Y Yan, P Tang, X Bai, and W Liu. 2018. Revisiting multiple instance neural networks. *Pattern Recognition*.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*.

Gang Xu, Zhigang Song, Zhuo Sun, Calvin Ku, Zhe Yang, Cancheng Liu, Shuhao Wang, Jianpeng Ma, and Wei Xu. 2019. Camel: A weakly supervised learning framework for histopathology image segmentation. In *ICCV*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, et al. 2020. Big bird: Transformers for longer sequences. *NeurIPS*.

H Zhang, Y Meng, Y Zhao, Y Qiao, X Yang, S Coupland, and Y Zheng. 2022. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *CVPR*.

Weijia Zhang. 2021. Non-iid multi-instance learning for predicting instance and bag labels using variational auto-encoder. *arXiv preprint arXiv:2105.01276*.

Yaqin Zhou, Shangqing Liu, Jingkai Siow, Xiaoning Du, and Yang Liu. 2019. Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. In *NeurIPS*.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *ICML*.

A Appendix

A.1 Details of Benchmark Datasets

We provide the detailed descriptions of seven benchmark datasets below.

Following the latest state-of-the-art (Park et al., 2022), we evaluate LaFiCMIL on six benchmark datasets. Hyperpartisan (Kiesel et al., 2019) is a small dataset (containing only 645 documents) designed for **binary classification** task where each document is labeled as `hyperpartisan` or `not hyperpartisan`. There are 53.49% of long documents in Hyperpartisan, i.e., exceeding 512 tokens. The 20NewsGroups (Lang, 1995) contains 20 well-balanced categories and 11 846 documents, which have been widely used over the past 20 years for **multi-class classification** task. Only less than 15% of the documents exceed 512 tokens.

The other 4 datasets are created for the most difficult **multi-label classification** task. The multi-label classification task aims at predicting multiple labels for a given document, which is different from multi-class classification that selects only one label from multiple possible categories. CMU Book Summary (Bamman and Smith, 2013) contains book summaries extracted from Wikipedia with corresponding meta-data from Freebase such as the book genre. After preprocessing, there are 12 788 documents and 227 genre labels such as "Fiction" and "Children's literature". The proportion of long documents that exceed 512 tokens is about 39%. Coming from EU legal documents, EURLEX-57K (Chalkidis et al., 2019) is a quite large dataset that contains 57 000 documents, with more than 51% of them exceed 512 tokens. In total, there are 4271 labels available, some of which do not appear in the training set often or at all, making it a very challenging dataset.

The main purpose of long document classification is to explore more useful information beyond the first 512 tokens. Therefore, in the latest state-of-the-art (Park et al., 2022), the CMU Book Summary and EURLEX-57K are modified to obtain two additional datasets to further evaluate the ability of these models to not fully rely on information from the first 512 tokens. Paired Book Summary is created by combining pairs of documents from CMU Book Summary to obtain a new dataset containing longer documents. With this setup, over 75% of documents in the Paired Book Summary dataset have more than 512 tokens. Regarding the EURLEX-57K, documents inside are usually legal texts with several sections, and the first two sections (i.e., header, recitals) normally carry the most relevant information for classification (Chalkidis et al., 2019). The order of the sections are inverted to ensure that the core information appears at the end of the document in (Park et al., 2022). The inverted EURLEX-57K has the same proportion of the long documents as the original EURLEX-57K dataset.

We evaluate the code defect detection task using the Devign dataset (Zhou et al., 2019) that includes 27 318 manually-labeled functions collected for C programming language.The dataset was created by collecting security-related commits and extracting vulnerable and non-vulnerable functions from the labeled commits. Since this dataset did not have an official dataset split, the code understanding benchmark CodeXCLUE (Lu et al., 2021b) randomly shuffles the dataset and splits it into 80% for training, 10% for validation, and 10% for test, which is adopted by latest state-of-the-arts (Hanif and Maffeis, 2022). The task is formulated as a binary classification to predict whether a function is defective/vulnerable.

Table 7: Ablation study on long document classification to investigate the effectiveness of positional embedding. "PE" indicates Positional Embedding.

Dataset	Only Long Docs		Full Docs	
	With PE	Without PE	With PE	Without PE
Hyperpartisan	**95.00**	95.00	**96.92**	96.92
20News Groups	**87.49**	87.22	84.81	**85.07**
EURLEX	67.14	**67.28**	73.43	**73.72**
Inverted EURLEX	64.52	**65.04**	71.81	**72.03**
Book Summary	**65.41**	64.14	**61.34**	60.44
Paired Summary	61.04	**63.03**	60.66	**62.17**

A.2 Additional Ablation Study

Given an intriguing phenomenon we observed related to **chunk positional embedding**, we conducted an additional ablation study to investigate its effectiveness. Basically, its effectiveness depends on different datasets.

In our implementation, we adopt learnable linear positional embedding in all experiments. We present the results of experiments `with/without` **chunk positional embedding** on long document classification task in Table 7. First and foremost, it is worth emphasizing that LaFiCMIL can achieve state-of-the-art results with or without chunk positional embedding on most datasets. In this section, we discuss under what cir-

cumstances the chunk positional embedding can be beneficial for prediction.

As shown in Table 7, when considering **only long** documents in the test set, we find that positional embedding has no effect on Hyperpartisan, while it yields gains for 20NewsGroups and Book Summary. Regarding the very small dataset Hyperpartisan, with only 65 samples in the test set, it is difficult to cover enough variety of cases for evaluation. Therefore, it is not surprising that the chunk positional embedding does not lead to further improvement on this small test set. In the 20NewsGroups dataset, which is collected from news articles, paragraphs exhibit semantic ordering relationships. Similarly, the contents of documents in the Book Summary dataset also have ordering dependencies. As a result, in the case of these two datasets, chunk positional embedding can effectively exploit the ordering information, leading to an improvement in classification performance.

However, the Paired Summary dataset is created by hard combining two documents selected from the Book Summary, which weakens the effectiveness of positional embedding since there is no sequential relationship between chunks from two different books. The samples in EURLEX-57K and Inverted EURLEX-57K datasets are legal documents, usually consisting of several sections. These sections have no strict sequential relationship between them since generally jumping to read different sections does not affect the understanding of the legal provisions. Therefore, chunk positional embedding fails to bring performance gains on both of these two datasets.

We can also observe a similar pattern when evaluating on the **full** test set, except for the 20NewsGroups dataset. This is due to the fact that more than 85% of the samples in this dataset are short documents containing only a single chunk, which makes the chunk positional embedding provide noise rather than relevant information, which may mislead the model during fine-tuning.

A.3 Code Defect Detection

In this section, we present the empirical study and a detailed analysis for the code defect detection task.

Empirical Setup

We adopt two BERT-like models pre-trained on programming languages as our baselines, i.e., CodeBERT (Feng et al., 2020) and VulBERTa (Hanif and Maffeis, 2022). Following the implementation of CodeBERT in the CodeXGLUE benchmark (Lu et al., 2021b), the length of each chunk is set to 400 tokens. LaFiCMIL can process up to 48 chunks (i.e., 19.2K tokens in total) on a single GPU at a time. Consequently, more than 99.92% (compared to 60.26% in CodeBERT) of code files in the dataset can be adequately processed without truncation to provide comprehensive information for accurate predictions. The latest state-of-the-art VulBERTa proposed a custom tokenizer for the C language, which is pre-trained with an input length of 512 tokens and then fine-tuned on the code defect detection task with a length of 1024 tokens. In LaFiCMIL, we set the chunk length to 512 tokens and employ the same tokenizer. For both CodeBERT and VulBERTa, we fine-tune them on the code defect detection task for 10 epochs using the AdamW optimizer (Loshchilov and Hutter, 2017) and a learning rate of `5e-6`.

Detailed Experimental Analysis

From Table 4, we find that code defect detection is a challenging task on which most existing state-of-the-art models struggle to achieve even a single percentage point improvement over previous models. Nonetheless, our LaFiCMIL helps CodeBERT gain 1.35 percentage points, representing a significant improvement. The gain can be attributed to the fact that LaFiCMIL can extract information from the 40% **large** code files (i.e., exceeding 400 tokens), which is partially missing in CodeBERT. Although the latest state-of-the-art VulBERTa extends the input limit to 1024 tokens, covering 90% of full code files in the dataset, our LaFiCMIL still brings a gain of 0.25 percentage points to it, thanks to the information extracted from the other 10% of **large** files.