

LH-Mix: Local Hierarchy Correlation Guided Mixup over Hierarchical Prompt Tuning

Fanshuang Kong
kongfs@buaa.edu.cn
CCSE, Beihang University
Beijing, China

Richong Zhang*
zhangrc@act.buaa.edu.cn
CCSE, Beihang University
Beijing, China

Ziqiao Wang
zqiaowang@tongji.edu.cn
Tongji University
Shanghai, China

ABSTRACT

Hierarchical text classification (HTC) aims to assign one or more labels in the hierarchy for each text. Many methods represent this structure as a global hierarchy, leading to redundant graph structures. To address this, incorporating a text-specific local hierarchy is essential. However, existing approaches often model this local hierarchy as a sequence, focusing on explicit parent-child relationships while ignoring implicit correlations among sibling/peer relationships. In this paper, we first integrate local hierarchies into a manual depth-level prompt to capture parent-child relationships. We then apply Mixup to this hierarchical prompt tuning scheme to improve the latent correlation within sibling/peer relationships. Notably, we propose a novel Mixup ratio guided by local hierarchy correlation to effectively capture intrinsic correlations. This Local Hierarchy Mixup (LH-Mix) model demonstrates remarkable performance across three widely-used datasets.

CCS CONCEPTS

- Computing methodologies → Supervised learning by classification; Natural language processing.

KEYWORDS

Mixup, Hierarchical Text Classification, Prompt Tuning

ACM Reference Format:

Fanshuang Kong, Richong Zhang, and Ziqiao Wang. 2018. LH-Mix: Local Hierarchy Correlation Guided Mixup over Hierarchical Prompt Tuning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Hierarchical Text Classification (HTC) is a variant of multi-label classification task characterized by labels organized in a predefined hierarchical structure [28]. This hierarchical structure captures relationships and dependencies among labels, with higher-depth

*Corresponding author: zhangrc@act.buaa.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, August 03–07, 2025, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-XXXX-X/18/06
<https://doi.org/XXXXXXX.XXXXXXX>

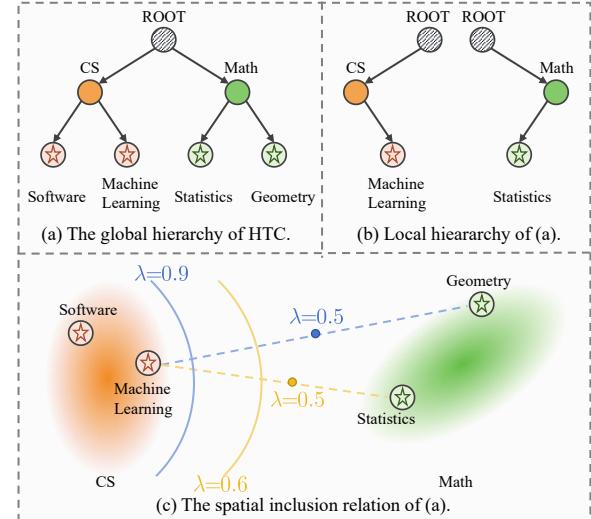


Figure 1: (a) A toy example of global hierarchy in HTC. (b) The local hierarchy of “CS/Machine Learning” and “Math/Statistics”, which are extracted from (a). (c) Transformation from explicit parent-child relations (a) to spatial inclusion relations in latent space. Mixup enables the capture of varying degrees of implicit sibling/peer label correlation through different Mixup ratios λ .

labels containing lower-depth ones. Each text is then assigned to one or more labels within this hierarchy.

The key challenge of HTC lies in effectively modeling the large-scale, imbalanced, and structured label hierarchy [21]. Some existing works [9, 32, 42] consider the hierarchy as a directed acyclic graph, utilizing structure encoders to obtain label representations that incorporate hierarchical information. Nevertheless, this strategy allows each text to share the entire static global hierarchy, introducing redundancy in the graph [16]. This redundancy becomes particularly noticeable as the hierarchy size increases, with numerous labels becoming irrelevant to a specific given target text [15]. In contrast, alternative approaches proposed in [15, 16] invoke a local hierarchy, defined as a text-relevant sub-hierarchy extracted from the global hierarchy. Figure 1 (a-b) illustrate examples of these two types of hierarchies. This local hierarchy can be represented as an inclusion relationship in the latent space, as depicted in Figure 1 (c). For example, “Software” and “Machine Learning” should occupy the same subspace within “CS”, while “Geometry” and “Statistics” should exist in a subspace within the category of “Math”. On top

of that, by treating a local hierarchy as a sequence, language models are able to capture parent-child relationships inherent in the hierarchy. Furthermore, motivated by the effectiveness of prompts in leveraging sequential information, in this work, we formulate the local hierarchy as a sequence using a hierarchical template to encode and align its structure.

It is worth mentioning that labels may hold inherent relevance that extends beyond the constraints imposed by parent-child relationships in the hierarchy. Figure 1 (c) highlights a correlation between different local hierarchies. Specifically, while “CS/Machine Learning” and “Math/Statistics” occupy separate subspaces in the hierarchy, they may exhibit close proximity in the latent space, indicating a nuanced relationship that goes beyond the immediate hierarchical constraints. In fact, due to the widespread correlation among labels in HTC, including both sibling relationships and peer relationships [29], capturing the correlation between local hierarchies becomes crucial. Existing methods, however, lack explicit mechanisms to tackle this specific challenge. An avenue of promise lies in Mixup [39], a technique utilized for augmenting latent correlations between input pairs through the generation of intermediate samples [39]. Therefore, alongside hierarchical prompt tuning, we incorporate Mixup to reveal and leverage the implicit correlation between local hierarchies.

More specifically, we first apply prompt tuning to HTC, where the local hierarchy is treated as a sequence. Virtual tokens are introduced to represent each depth in the hierarchy, thereby disassembling the parent-child relationship and aligning the hierarchy by depth. Within this hierarchical prompt tuning framework, we incorporate Mixup to capture implicit correlations among sibling and peer relationships. Notably, unlike Vanilla Mixup, which interpolates between the input and corresponding target using the same Mixup ratio sampled from a Beta Distribution, recent studies [26, 40] adopt Mixup ratios based on instance correlations. This approach adjusts the mixture degree based on instance similarity, assigning higher mixture ratios to more similar instances to generate more informative samples. Conversely, less similar examples invoke a lower degree of mixture to prevent the generation of out-of-distribution data [1]. As shown in Figure 1 (c), “CS/Machine Learning” should have a higher degree of mixing (e.g., with a Mixup ratio 0.6) with “Math/Statistic”, and a lower degree of mixing (e.g., with a Mixup ratio 0.9) to “Math/Geometry”. Motivated by this observation, we propose a novel approach for Mixup ratio control in our framework, dubbed local hierarchy Mixup (LH-Mix). In particular, LH-Mix assesses the correlation of a local hierarchy pair using the representation derived from the hierarchical prompt. By incorporating a heuristic function to determine the correlation and Mixup ratio, LH-Mix is able to learn hierarchical label correlations more effectively than vanilla Mixup.

Below we summarize the main contributions of our work:

- We introduce LH-Mix, which integrates Mixup based on a depth-level hierarchical prompt, effectively modeling the local hierarchy within sequences. To the best of our knowledge, LH-Mix is the first application of Mixup in HTC.
- To capture the inherent correlations among local hierarchies, we propose a strategy for adjusting the Mixup ratio. This

adaptive approach, guided by the local hierarchy correlation, encourages the generation of enriched in-between samples.

- We evaluate our LH-Mix on three standard HTC datasets, demonstrating its effectiveness through the empirical performance.

2 RELATED WORK

2.1 Hierarchical Text Classification (HTC)

HTC refers to a specific type of multi-label text classification problem, where an instance lies in one or more paths from a taxonomic hierarchy in a top-down manner. The hierarchical structure is typically represented as a tree or a directed acyclic graph, where the root represents the highest-level label, and the leaf nodes correspond to the most specific labels. The key challenge in HTC lies in how to effectively utilize the large-scale, imbalanced, and structured label hierarchy. To perform HTC, various techniques can be employed. Previous works can be categorized into two types [9, 42]: global approach and local approach. The global approach tackles the HTC problem as a flat multi-label classification problem, employing global hierarchy as input and building a single classifier for all labels. There are various strategies that incorporate the global hierarchy, such as capsule network [23], reinforcement learning [21], meta-learning [36], contrastive learning [32], hyperbolic representation [5], or structure encoder [9, 42]. On the contrary, the local approach typically involves constructing individual classifiers for each label [2], each parent node [10], or each level of the label hierarchy [18, 27, 34].

The current SOTA HTC model, HBGL [16], proposes hierarchy-guided BERT with both global and local hierarchies to utilize the prior knowledge of the pre-trained language model. The suboptimal model HPT [33] transforms the global prediction to a local one by hierarchical prompt tuning. These successful applications of local hierarchy have served as inspiration for further investigation of local hierarchy in this paper.

2.2 Mixup

Mixup is a data augmentation method proposed by [39], which aims to enhance the generalization capabilities of neural networks by generating in-between samples through linearly interpolating pairs of input text and their corresponding labels. Vanilla Mixup interpolates input pairs by a mixing ratio sampling from a basic Beta Distribution. However, the mixing ratio obtained through a blind sampling process may not be optimal. To enhance performance, several studies have attempted to control the mixing ratio for pairs. AdaMixUp [13] incorporates the automatic mixing policies from the data through the utilization of an additional network and an objective function to prevent “manifold intrusion”. CAMixup [35] adjusts the mixing ratio through the relation between the predict confidence and accuracy. Remix [8] assigns labels in a manner that favors the minority class by providing disproportionately higher weights to the minority class. Nonlinear Mixup [11] incorporates a nonlinear interpolation policy for both the input and label pairs, wherein the mixing policy for the labels is adaptively learned, leveraging the information from the mixed input. Basically, Mixup could enhance model performance by an adaptive mixing ratio that captures deeper relationships within the latent space between samples.

Due to the strong scalability and effectiveness, Mixup and its variants have shown promising improvements in various tasks, such as classification [12, 38], unsupervised domain adaptation [20, 37], and semi-supervised learning [7]. However, in the context of HTC, there has been limited exploration of applying Mixup to augment the hierarchical label representation.

3 PRELIMINARIES

3.1 HTC Setting

Given a training dataset $\{(X_i, Y_i)\}_{i=1}^N$, where X_i denotes the input text, Y_i is the corresponding label of X_i , and N denotes the size of training dataset. Let \mathcal{V} be the label set. In the HTC setting, \mathcal{V} is further divided into D different subsets, denoted as $\mathcal{V} = \{\mathcal{V}^1, \dots, \mathcal{V}^d, \dots, \mathcal{V}^D\}$, where D is the total depth of the hierarchy and \mathcal{V}^d is the label set of the d -th depth. Importantly, \mathcal{V} can be organized into a tree structure, representing the global hierarchy of the entire dataset. In addition, each input X_i contains multiple labels from \mathcal{V} , and these labels can form a separate subtree, known as the local hierarchy. The goal of HTC is to assign labels from the local hierarchy in \mathcal{V} to each text in the test dataset.

3.2 Hierarchical Prompt for HTC

Typically, a prompt contains a task-specific template with the text serving as input, exploiting the knowledge ingrained within the pre-trained language model [3]. A vanilla prompt framework in HTC, as outlined in [33], involves the creation of D soft tokens within the template, each specifically associated with a depth in the hierarchy. For instance, the input can be formulated as:

$$[\text{CLS}] [\text{Dth}^1] [\text{MASK}] \dots [\text{Dth}^d] [\text{MASK}] \dots [\text{Dth}^D] [\text{MASK}] [\text{SEP}] \\ X [\text{SEP}]$$

In particular, each token $[\text{Dth}^d]$ serves to prompt the prediction of the associated label \mathcal{V}^d within its d -th layer hierarchy. This utilizes the hidden output of the $[\text{MASK}]$ token immediately following it. More precisely, let $\mathbf{h}_{[\text{MASK}]}^d$ be the hidden output of $[\text{MASK}]$ in the d -th layer, the classification procedure is then defined as:

$$\mathbf{p}^d = f_{\mathcal{V}^d}^d(\mathbf{h}_{[\text{MASK}]}^d), \quad (1)$$

where $\mathbf{p}^d \in \mathbb{R}^{|\mathcal{V}^d|}$ is the prediction score vector, and $f_{\mathcal{V}^d}^d$ is a mapping from hidden output to the prediction scores in the d -th layer. Note that $f_{\mathcal{V}^d}^d$ contains a classifier trained through the Masked Language Model task, along with a label words verbalizer that adheres to the label words of \mathcal{V}^d . Notably, by using this hierarchical prompt for HTC, the classification process is now partitioned across each level of the hierarchy, rather than predicting all labels in a single classifier. This approach encodes the local hierarchy as a sequence, allowing for alignment of the hierarchy at each depth level for every input. Consequently, this hierarchical prompt design greatly facilitates subsequent Mixup procedures. We will elaborate on this later.

In the HTC task, many existing models treat the task as a multiple-label classification and invoke the conventional Binary Cross Entropy (BCE) loss. Nevertheless, it has been pointed out that BCE loss ignores the correlation between labels. A more effective alternative, as suggested in [33], is the Zero-bounded Multi-label Cross Entropy (ZMLCE). Particularly, the ZMLCE loss considers that all scores for

positive labels are greater than 0, while those for negative labels are less than 0. The ZMLCE loss is formally defined as follows:

$$\mathcal{L}^d(\mathbf{p}^d, \mathcal{V}_{\text{pos}}^d, \mathcal{V}_{\text{neg}}^d) = \log(1 + \sum_{v \in \mathcal{V}_{\text{neg}}^d} e^{\mathbf{p}_v^d}) + \log(1 + \sum_{v \in \mathcal{V}_{\text{pos}}^d} e^{-\mathbf{p}_v^d}), \quad (2)$$

where $\mathcal{V}_{\text{pos}}^d$ and $\mathcal{V}_{\text{neg}}^d$ are the positive label set and negative label set in the d -th layer, respectively, and v denotes the v -th element of a vector (e.g., \mathbf{p}_v^d refers to the v -th element of the prediction score vector). In the inference phase, a prediction score greater than 0 is considered as positive; otherwise, it is considered as negative.

4 LH-MIX FRAMEWORK

Mixup, as introduced in [39], uses linear interpolation between inputs and their respective labels to generate in-between samples, thereby enriching the inherent structures within the latent space [37]. In this work, we integrate Mixup into the hierarchical prompt framework to better capture the correlations within the hierarchically structured labels of HTC. In particular, we propose a novel adaptive Mixup ratio strategy, guided by the local hierarchy correlation, to further refine the vanilla Mixup scheme. We term this enhanced approach LH-Mix. The framework of LH-Mix is shown in Figure 2.

4.1 Local Hierarchy Correlation

The design of this hierarchical prompt scheme is motivated by the fact that the local hierarchy can be effectively represented as a sequence. Specifically, an important characteristic of hierarchy is its strict hierarchical relationship, where the prediction of lower-level labels relies on the prediction of higher-level labels. To achieve this characteristic, we introduce a depth token $[\text{Dth}^d]$ to represent the depth level of each label. The local hierarchy can be reformulated to a sequence of tokens and this will be further translated to a soft token as hierarchical prompts via a pre-trained language model. We believe this soft prompt integrates both the level and the label information, allowing the representation of hierarchies as sequences. More specially, for a specific input X , its local hierarchy can similarly be expressed as such a sequence by substituting the $[\text{MASK}]$ token with the corresponding gold label. For example, the two local hierarchies “CS/Machine Learning” and “Math/Statistics” in Figure 1 can be respectively represented as:

$[\text{CLS}][\text{Dth}^1]\text{CS}[\text{Dth}^2]\text{Machine Learning}[\text{SEP}]$

$[\text{CLS}][\text{Dth}^1]\text{Math}[\text{Dth}^2]\text{Statistics}[\text{SEP}]$

Consequently, we transform the local hierarchy into a “sentence”. In addition, when a local hierarchy contains multiple paths, we concatenate the labels at the same level together and place them after the corresponding $[\text{Dth}^d]$ token.

Furthermore, notice that by feeding a sentence into a pre-trained language model and extracting the hidden output of $[\text{CLS}]$ token, one can obtain a sentence embedding that can be used to calculate sentence similarity [24]. Similarly, in our context, the $[\text{CLS}]$ output corresponding to the “local hierarchy sequence” serves as the representation of the local hierarchy. This representation can be effectively utilized to compute the local hierarchy correlation. Formally, consider two inputs X_i and X_j , let their local hierarchy representations be denoted by $\mathbf{h}_{i[\text{CLS}]}$ and $\mathbf{h}_{j[\text{CLS}]}$, respectively.

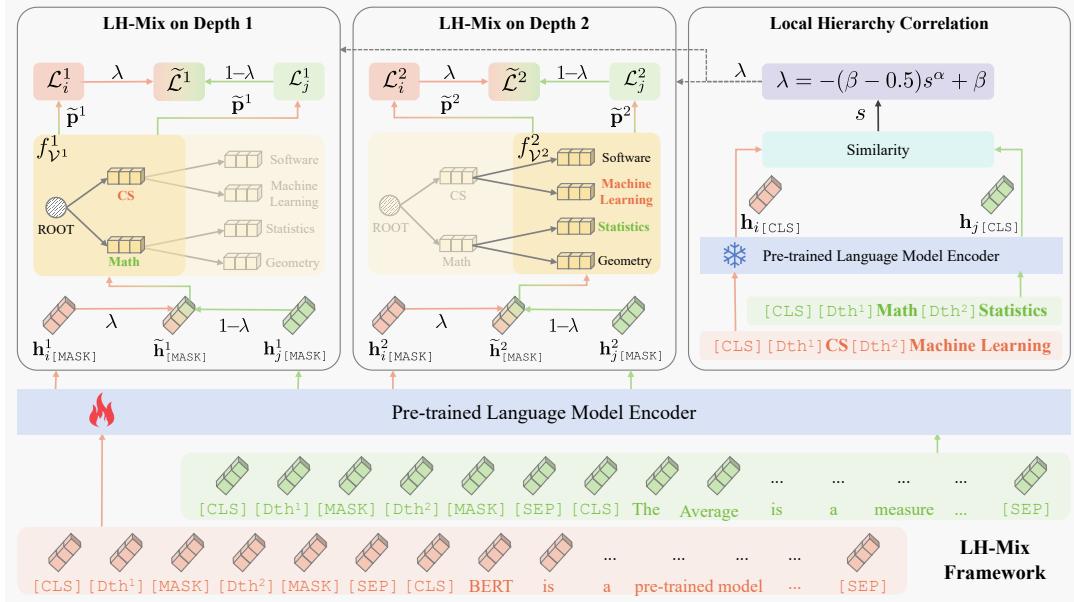


Figure 2: Illustration of LH-Mix. The light orange color scheme represents elements of X_i , and the light green represents X_j . The mixture of orange and green represents the elements related to the Mixup operation.

The distance between these two hierarchies is measured by the normalized cosine similarity, namely,

$$s = 0.5 \left(\frac{\mathbf{h}_i[\text{CLS}] \cdot \mathbf{h}_j[\text{CLS}]}{\|\mathbf{h}_i[\text{CLS}]\| \|\mathbf{h}_j[\text{CLS}]\|} + 1 \right), \quad (3)$$

where \cdot denotes the vector dot product and $\|\cdot\|$ is the L^2 norm. Note that due to the shared hierarchical encoding format, local hierarchy representations use the same pre-trained model encoder. Nonetheless, it is important to highlight that the encoder utilized in similarity calculation, distinct from the one used for classification, remains unaffected by gradient updates.

4.2 Local Hierarchy Correlation Guided Mixup Ratio

Mixup captures sample correlation by generating in-between samples via a Mixup ratio. The magnitude of the Mixup ratio can be controlled to generate varying degrees of “hard examples”, thereby determining the extent to which Mixup affects the data [8, 41]. [1, 26, 40] demonstrate that examples with different levels of similarity should be mixed with varying intensities. Therefore, given the varying correlation among different local hierarchy pairs, it is more appropriate in this context to apply different Mixup ratios to each local hierarchy pair, rather than drawing the Mixup ratio from a fixed distribution (e.g., Beta distribution). While numerous studies have explored the relationship between correlation and the Mixup ratio, there is no well-established theoretical framework that precisely characterizes the numerical relationship between them. In fact, rigorously justifying the effectiveness of Mixup still remains an open problem. Hence, to reflect the relationship between the Mixup ratio and similarity, we design a heuristic function based on intuition. The underlying intuition here is that as the similarity

between two local hierarchies increases, we expect Mixup to better capture the latent correlation between them. Specifically, we expect Mixup to have more impact (i.e., with a Mixup ratio approaching 0.5) on highly correlated local hierarchies, whereas its impact is diminished (i.e., with a Mixup ratio approaching 1) on less correlated local hierarchies. Bear this in mind, to formulate the relationship between local hierarchy similarity s and Mixup ratio λ , we have heuristically designed the following function:

$$\lambda = -(\beta - 0.5)s^\alpha + \beta, \quad (4)$$

where $\alpha > 0$ controls the rate of change of λ with respect to s , and $\beta \in (0.5, 1]$ controls the upper bound of λ . This function has the advantage of covering various common linear or nonlinear relationships between s and λ by simply adjusting the values of α and β . To gain better understanding about Eq. 4, we visualize the effects of β and α in Figure 3. In particular, when $\alpha = 1$, there is a linear relationship between s and λ . When $\alpha < 1$, λ decreases at a slower rate as s increases. Conversely, when $\alpha > 1$, λ decreases at a faster rate as s increases. Moreover, β determines the maximum value of λ , signifying the minimum impact of Mixup.

4.3 Local Hierarchy Mixup

We now apply Mixup with the local hierarchy correlation guided ratio to both input and output simultaneously at each depth of the hierarchy.

Regarding input Mixup, following the approach of previous Mixup variants in text classification [12], we interpolate the hidden output corresponding to the [MASK] token for an input pair at each depth of the hierarchy as follows:

$$\tilde{\mathbf{h}}_{[\text{MASK}]}^d = \lambda \mathbf{h}_{i[\text{MASK}]}^d + (1 - \lambda) \mathbf{h}_{j[\text{MASK}]}^d. \quad (5)$$

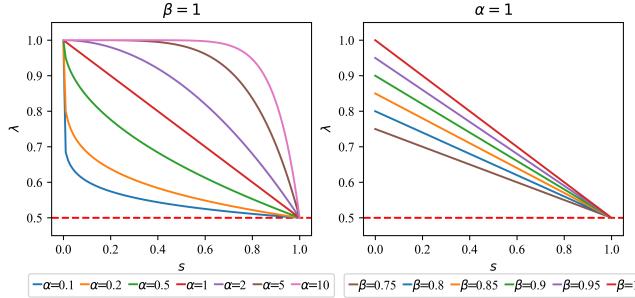


Figure 3: Curves corresponding to Eq. 4. We separately plot the effects of different β and α on the function when $\alpha = 1$ and $\beta = 1$.

Subsequently, we obtain the prediction score $\tilde{\mathbf{p}}^d$ for the mixed input $\tilde{\mathbf{h}}_{[\text{MASK}]}^d$ using $f_{\mathcal{V}^d}^d$, namely,

$$\tilde{\mathbf{p}}^d = f_{\mathcal{V}^d}^d(\tilde{\mathbf{h}}_{[\text{MASK}]}^d). \quad (6)$$

For output Mixup, the straightforward approach involves mixing the labels, as formulated in the vanilla Mixup technique [39]. Alternatively, several studies have demonstrated that mixing the losses is also a valid method. In particular, it has been established that gradients for label mixing and loss mixing are equivalent in the context of cross-entropy loss [4]. In the scenario of the ZMLCE loss, characterized by the division of positive and negative texts with 0 as an anchor, all positive labels and negative labels are treated as distinct combinations. While this design enables ZMLCE to focus on correlations between labels, it neglects the consideration of relative magnitudes among positive labels or negative labels. This oversight can obscure the interpretation of label mixing, making its meaning less explicit. Therefore, we opt to mix the loss terms in our LH-Mix, defined as:

$$\begin{aligned} & \tilde{\mathcal{L}}^d(\lambda, \tilde{\mathbf{p}}^d, \mathcal{V}_{i,\text{pos}}^d, \mathcal{V}_{i,\text{neg}}^d, \mathcal{V}_{j,\text{pos}}^d, \mathcal{V}_{j,\text{neg}}^d) \\ &= \lambda \mathcal{L}_i^d(\tilde{\mathbf{p}}^d, \mathcal{V}_{i,\text{pos}}^d, \mathcal{V}_{i,\text{neg}}^d) + (1 - \lambda) \mathcal{L}_j^d(\tilde{\mathbf{p}}^d, \mathcal{V}_{j,\text{pos}}^d, \mathcal{V}_{j,\text{neg}}^d). \end{aligned} \quad (7)$$

Further, the gradient of models with parameters θ is given by:

$$\begin{aligned} & \nabla_\theta \left(\lambda \left(\log \left(1 + \sum_{v \in \mathcal{V}_{i,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d} \right) + \log \left(1 + \sum_{v \in \mathcal{V}_{i,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d} \right) \right) \right. \\ &+ \left. (1 - \lambda) \left(\log \left(1 + \sum_{v \in \mathcal{V}_{j,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d} \right) + \log \left(1 + \sum_{v \in \mathcal{V}_{j,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d} \right) \right) \right) \\ &= \lambda \left(\frac{\sum_{v \in \mathcal{V}_{i,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d}}{1 + \sum_{v \in \mathcal{V}_{i,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d}} - \frac{\sum_{v \in \mathcal{V}_{i,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d}}{1 + \sum_{v \in \mathcal{V}_{i,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d}} \right) \\ &+ (1 - \lambda) \left(\frac{\sum_{v \in \mathcal{V}_{j,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d}}{1 + \sum_{v \in \mathcal{V}_{j,\text{neg}}^d} e^{\tilde{\mathbf{p}}_v^d}} - \frac{\sum_{v \in \mathcal{V}_{j,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d}}{1 + \sum_{v \in \mathcal{V}_{j,\text{pos}}^d} e^{-\tilde{\mathbf{p}}_v^d}} \right) \end{aligned} \quad (8)$$

Clearly, Eq. 8 delivers a linear combination for the gradients of the mixed prediction scores corresponding to X_i and X_j . In addition, the vanilla Mixup, specifically in the case of cross-entropy, also generates a linear combination of the gradients from the mixed

prediction score. In this sense, the loss mixing for ZMLCE aligns more closely with the vanilla Mixup. This also validates our decision to adopt Eq. 7 for optimization, favoring loss mixing over label mixing.

5 EXPERIMENT

5.1 Datasets and Evaluation Metrics

We evaluate our model on three widely used datasets: WebOfScience (WOS) [18], NYTimes (NYT) [25] and RCV1-V2 [19]. The statistic information is shown in Table 1. For evaluation metrics, we adopt Macro-F1 and Micro-F1 to measure the results following the previous work [14, 16, 32, 33]. Micro-F1 considers the overall precision and recall of all instances, while Macro-F1 represents the average F1-score across labels.

Dataset	D	$ \mathcal{V} $	Avg($ \mathcal{V}_i $)	#Train	#Dev	#Test
WOS	2	141	2.0	30,070	7,518	9,397
NYT	8	166	7.6	23,345	5,834	7,292
RCV1-V2	4	103	3.24	20,833	2,316	781,265

Table 1: Data statistics. D is the maximum depth of the hierarchy. $|\mathcal{V}|$ is the number of labels. Avg($|\mathcal{V}_i|$) is the average number of labels per instance. # represents the number of instance in the datasets.

5.2 Implement Details

Following the previous works, we exploit the pre-trained model bert-base-uncased from Hugging Face Transformers¹ to evaluate our model. For hierarchical prompting, the newly added D tokens [$D\mathbf{th}^d$] are randomly initialized, and the label words verbalizer embeddings are initialized by the average of the label name representation. All parameters are fine-tuned by an Adam [17] optimizer with the learning rate as 3e-5. For LH-Mix related parameters, as Figure 3 shows, we choose α from [0.1, 0.3, 0.6, 1, 2, 5, 10] and β from [0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1]. This configuration is able to encompass the majority of common relationships between s and λ . To accelerate the convergence of the model, we adopt a two-step training strategy. Initially, we train the model for 5 epochs without utilizing Mixup, and then for the remaining epochs, we incorporate Mixup during the training process. We employ the early stopping strategy if there is no improvement of Macro-F1 after 5 epochs. All experiments are conducted on a Tesla V100-SXM2-32GB GPU, with each epoch taking approximately 1800s for training on medium size dataset NYT. For more details about training, we will release code our the Github².

5.3 Comparable Models

As the practice in previous works, we compare LH-Mix with three groups of models: hierarchy-aware model, large language model, and pre-trained language model. In the hierarchy-aware model,

¹<https://github.com/huggingface/transformers>

²<https://github.com/fskong/LH-Mix>

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Hierarchy-aware Models						
TextRNN [42]	83.55	76.99	70.83	56.18	81.57	59.25
HiAGM [42]	85.82	80.28	74.97	60.83	83.96	63.35
HTCInfoMax [9]	85.58	80.05	-	-	83.51	62.71
HiMatch [6]	86.20	80.53	-	-	84.73	64.11
Large Language Model						
ChatGPT [14]	-	-	-	-	51.35	32.20
LLaMA	83.36	73.34	76.72	58.01	87.49	62.12
Pre-trained Language Model						
BERT [32]	85.63	79.07	78.24	65.62	85.65	67.02
BERT+HiAGM [32]	86.04	80.19	78.64	66.76	85.58	67.93
BERT+HTCInfoMax [32]	86.30	79.97	78.75	67.31	85.53	67.09
BERT+HiMatch [6]	86.70	81.06	-	-	86.33	68.66
HGCLR [32]	87.11	81.20	78.86	67.96	86.49	68.31
HPT [33]	87.16	81.93	80.42	<u>70.42</u>	<u>87.26</u>	69.53
HBGL [16]	<u>87.36</u>	<u>82.00</u>	80.47	70.19	87.23	<u>71.07</u>
HJCL [14]	-	-	<u>80.52</u>	70.02	87.04	70.49
HiTIN [43]	87.19	81.57	79.65	69.31	86.71	69.95
LH-Mix	87.27	82.07	80.57	71.26	87.47	71.93

Table 2: Main results of LH-Mix and other baseline models. The underlined results are the current state-of-the-art (SOTA). The bolded results are the best performance including LH-Mix.

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Bert	85.63	79.07	78.24	65.62	85.65	67.02
+Mixup	86.58	81.05	78.89	67.85	86.36	67.33
+LH-Mix	86.86	81.31	79.17	68.46	86.52	70.04
Prompt	86.94	81.37	80.23	70.13	87.11	69.82
+ Mixup	87.13	81.82	80.35	70.63	87.45	71.39
+ LH-Mix	87.27	82.07	80.57	71.26	87.47	71.93

Table 3: Ablation study of different variants.

we compare our model with 4 strong baselines: TextRNN [42], HiAGM [42], HTCInfoMax [9], and HiMatch [6]. In the large language model, we generally report instruction-tuned results by ChatGPT [22] and supervised fine-tuned results of LLaMA-2-7B [30]. In the pre-trained language model, except for substituting the encoder as BERT of the 4 mentioned hierarchy-aware baselines, we also consider 5 newly proposed models including HGCLR [32], HPT [33], HBGL [16], HJCL [14], and HiTIN [43]. Among these baseline models, HPT and HBGL overall achieve state-of-the-art performance. For more details on these comparable models, please refer to Appendix A.1.

5.4 Main Results

The main results of Micro-F1 and Macro-F1 of three datasets are shown in Table 2. As the Table shows, LH-Mix achieves the best performance in five out of the total six metrics, indicating the

effectiveness of LH-Mix. To further evaluate the reproducibility and significance of LH-Mix, we conduct experiments on statistical performance in Appendix A.3.

With a detailed investigation of these results, we first observe that WOS shows a smaller improvement compared to NYT and RCV1-V2. We believe this is due to the smaller number of the depth of WOS, which results in easier classification. We then observe that LH-Mix shows a larger improvement in Macro-F1. We believe this is because Macro-F1 measures label-level F1, while Micro-F1 measures more strict instance-level F1. LH-Mix primarily improves the hierarchical label correlation, thus leading to a greater improvement for Macro-F1.

Furthermore, comparing the HTC results of the instruction-tuned large language model reported in [14], which is based on ChatGPT gpt-turbo-3.5, the results indicate that large models still face significant challenges in encoding complex hierarchical structures. Comparing the results with the supervised fine-tuned

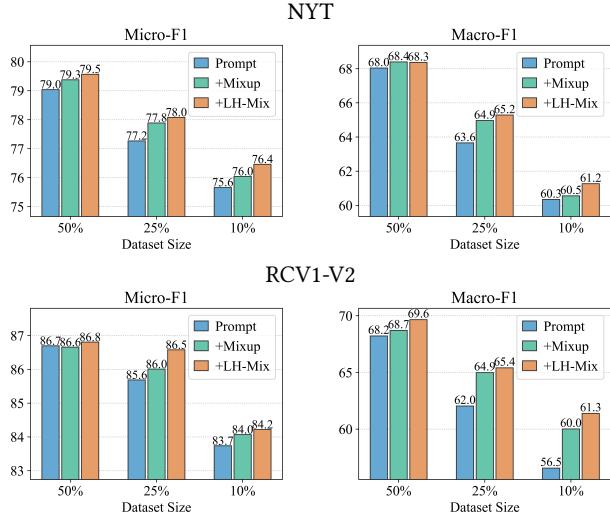


Figure 4: Performance on different downsample ratios to sparse data.

large language models, like LLaMA, we found that fully fine-tuning BERT-based models achieve better performance, indicating the encoder architecture is still effective in natural language understanding. Also, considering the training time for LLaMA is significantly longer than BERT, we believe that LH-Mix currently has distinctive advantages.

5.5 Ablation Study

To evaluate the importance of each variant in LH-Mix, we conduct a series of ablation experiments, including the following variations: BERT, which serves as the baseline model without any additional variant; Prompt, which introduces hierarchical templates for prompts tuning; +Mixup, which apply the vanilla Mixup with Mixup ratio from conventional Beta Distribution based on BERT or Prompt; +LH-Mix, which further utilize hierarchy correlation guided Mixup ratio for +Mixup. The results are shown in Table 3.

The results that Prompt is superior to the basic BERT model, demonstrating the effectiveness of hierarchical templates. Additionally, the performance of the +Mixup is better than its corresponding basic BERT or Prompt model, indicating the utility of using Mixup for encoding hierarchical label correlation. Furthermore, +LH-Mix outperforms +Mixup, providing evidence for the capability of incorporating hierarchical label correlation in Mixup ratio controlling.

Additionally, we notice that numerous models treat the hierarchy as a graph and employ GNN-based models to generate structural label embedding for classification. Thus, we compare Prompt with a series of GNN-based graph models to evaluate the effect of the hierarchical prompt scheme in Appendix A.4.

5.6 Performance on Sparse Datasets

Since LH-Mix is essentially a data augmentation method, it is necessary to evaluate LH-Mix in the low resource setting. Therefore, we downsample 50%, 25%, 10% of the training data for analysis. The results of Prompt, vanilla Mixup (+Mixup) and LH-Mix (+LH-Mix)

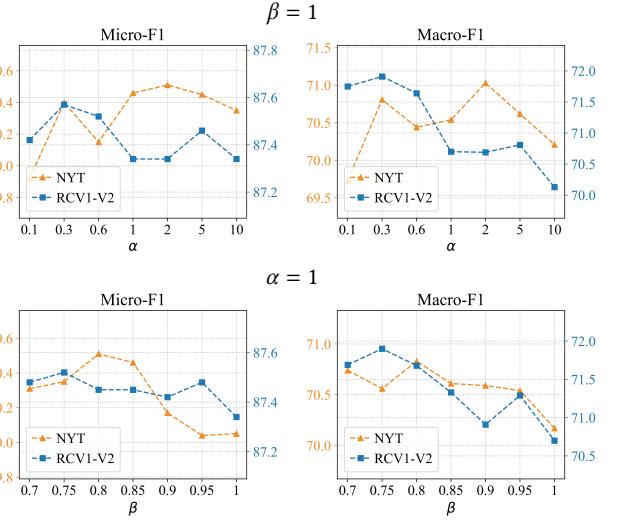


Figure 5: Performance on different β and α , when fixing $\alpha = 1$ and $\beta = 1$ respectively.

are shown in Figure 4. To save space, the results of WOS are shown in Appendix A.2. Results in the Figure indicate that +Mixup and +LH-Mix continue to perform better than Prompt with the decreasing of training samples. This also proves the effectiveness of Mixup methods in hierarchy label correlation capturing on even sparser datasets. Additionally, +LH-Mix is better than +Mixup generally, and as the dataset decreases extremely to 10%, the performance gap between +LH-Mix and +Mixup is enlarged. This observation demonstrates the efficiency of the adaptive Mixup ratio guided by local hierarchy correlation.

5.7 Parameter Analysis

In LH-Mix, the parameters α and β are crucial for controlling the relationship between hierarchy correlation and Mixup ratio. To investigate the impact of α and β , we fix $\alpha = 1$ and $\beta = 1$, and analyze the results as the other parameter varied. The results are shown in Figure 5. Due to the relatively small changes of Micro-F1 (as explained in Section 5.4), the trends may not accurately reflect the final results. Therefore, we temporarily focus on the Macro-F1 for consideration.

5.7.1 Effect of α . When $\beta = 1$ is fixed, for NYT, the Macro-F1 initially increases and then decreases with increasing α . For RCV1-V2, the Macro-F1 gradually decreases. We speculate that this is likely due to differences in the statistical characteristics of the datasets. This demonstrates that the relationship between s and λ does indeed impact the effectiveness of LH-Mix.

5.7.2 Effect of β . Reviewing Figure 3, as β increases, there is a higher likelihood for λ to be larger, indicating that LH-Mix is more likely to have a minor effect. From Figure 5, when $\alpha = 1$ is fixed, the results for both NYT and RCV1-V2 gradually decrease with increasing β . This indicates that a higher degree of LH-Mix leads to better performance, confirming the effectiveness of the LH-Mix.

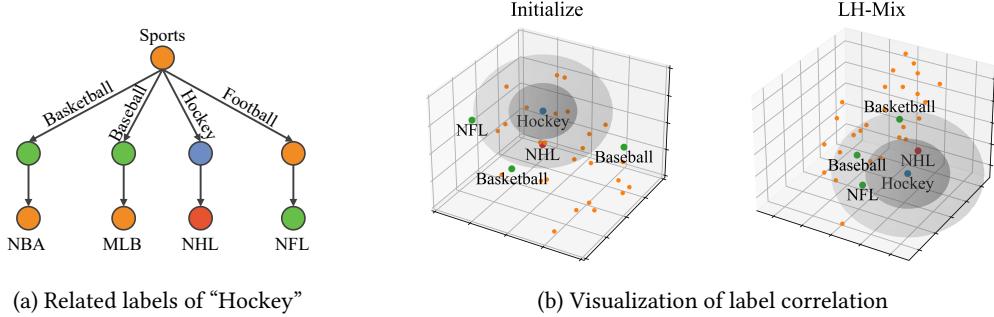


Figure 6: (a) Example of hierarchically related labels on “Hockey”. (b) The visualization of label correlations related to “Hockey” (the blue dot) before and after training by LH-Mix. We select the most similar 30 labels to “Hockey” for display and mark them with orange scatters. Two shades of gray spheres construct the top 3 and 15 similar label spaces, respectively. Red dots indicate labels that are not initially ranked within the top 3 but are included after training, while green dots represent labels that are not initially ranked within the top 15 but are included after training.

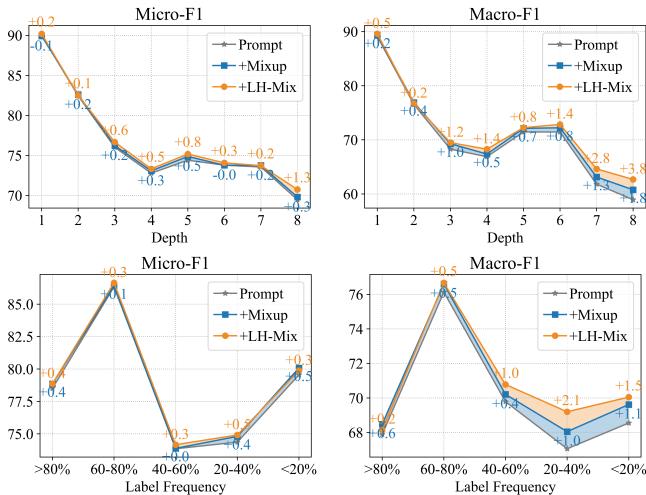


Figure 7: Performance of comparable models on different depths of the hierarchy and different frequencies of labels. The texts in the Figure indicate the superior performance of +Mixup and +LH-Mix compared to Prompt.

5.8 Variation of Label Correlation by LH-Mix

To further analyze whether LH-Mix learns more accurate hierarchical label correlation, we conduct a case study on label similarity. Specifically, we select the label “Hockey” as the target and compute the top 30 similar labels before and after learning with LH-Mix. The visualization of label correlation by t-SNE [31] is shown in Figure 6, marked by Initialize and LH-Mix.

From the Figure, we first observe that the most related label “NHL” has a higher similarity rank after learning (from top 15 to top 3). Additionally, the rank of labels “Baseball”, “Basketball”, and “NFL”, which are also closely related to “NHL”, have improved (from top 30 to top 15). Noted, for conciseness and clarity, we only emphasize the labels with significant differences before and after training. However, for labels with minor variations, such as “NBA”,

“MLB”, and “Football”, we do not provide any special markings. This does not imply that these labels are irrelevant; rather, it is because these labels have consistently been learned effectively. These observations indicate better performance in hierarchical label correlation after learning with LH-Mix.

5.9 Improvements of LH-Mix on Hierarchy

To evaluate the performance of LH-Mix in different hierarchical structures, we analyze the performance of the Prompt, vanilla Mixup (+Mixup), and LH-Mix (+LH-Mix) models on the NYT dataset, which contains the most complex hierarchical structure. Specifically, the complexity of the hierarchical structure is reflected in two aspects: the depth of the hierarchy and the distribution of label frequency. It is generally believed that the deeper the label hierarchy and the lower the label frequency, the more challenging it is to learn label representations, resulting in lower accuracy. The comparison of these models is shown in Figure 7.

From the Figure, we find that as the label hierarchy deepened and the label frequency decreased, the improvement of +Mixup and +LH-Mix compared to the baseline Prompt is larger and larger. This demonstrates the effectiveness of Mixup-related methods in enhancing the learning performance of labels in complex structures. Furthermore, we observe that the improvement achieved by +LH-Mix is superior to that of +Mixup, indicating that our proposed LH-Mix method can further capture the hierarchical label correlation.

6 CONCLUSION

HTC is an important scenario within multi-label text classification and has numerous applications. In this paper, we first propose a hierarchical template to model and align the local hierarchy in the prompt tuning framework. By employing this hierarchical prompt tuning, we formulate the parent-child relationships explicitly and effectively. Based on this, we employ Mixup to capture the implicit sibling/peer relationships under the latent label space. Especially we induce a local hierarchy correlation guided Mixup strategy to regulate the Mixup ratio for improved hierarchical label correlations, named LH-Mix. Extensive experiments on three widely-used HTC datasets confirm the effectiveness of our model.

ACKNOWLEDGMENTS

This work was supported by the National Science and Technology Major Project under Grant 2022ZD0120203, in part by the National Natural Science Foundation of China (No. U23B2056), in part by the Fundamental Research Funds for the Central Universities, and in part by the State Key Laboratory of Complex & Critical Software Environment.

REFERENCES

- [1] Raphael Baena, Lucas Drumetz, and Vincent Gripon. 2022. Preventing manifold intrusion with locality: Local mixup. *arXiv preprint arXiv:2201.04368* (2022).
- [2] Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsoultsiuklis. 2019. Hierarchical Transfer Learning for Multi-label Text Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/p19-1633>
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [4] Oscar Chang, Dung N Tran, and Kazuhito Koishiwa. 2021. Single-Channel Speech Enhancement Using Learnable Loss Mixup.. In *Interspeech*. 2696–2700.
- [5] Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI conference on artificial intelligence*. 7496–7503.
- [6] Haibin Chen, Qianli Ma, Zenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware Label Semantics Matching Network for Hierarchical Text Classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. 4370–4379.
- [7] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2147–2157. <https://doi.org/10.18653/v1/2020.acl-main.194>
- [8] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. 2020. Remix: rebalanced mixup. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI* 16. Springer, 95–110.
- [9] Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip Yu. 2021. HTCInfoMax: A Global Model for Hierarchical Text Classification via Information Maximization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3259–3265.
- [10] Susan Dumais and Hao Chen. 2000. Hierarchical classification of Web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. <https://doi.org/10.1145/345508.345593>
- [11] Hongyu Guo. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4044–4051.
- [12] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *CoRR* abs/1905.08941 (2019). [arXiv:1905.08941](https://arxiv.org/abs/1905.08941) <http://arxiv.org/abs/1905.08941>
- [13] Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3714–3722.
- [14] Jie He, Víctor Gutiérrez-Basulto, Jeff Z Pan, et al. 2023. Instances and Labels: Hierarchy-aware Joint Supervised Contrastive Learning for Hierarchical Multi-Label Text Classification. *arXiv preprint arXiv:2310.05128* (2023).
- [15] SangHun Im, GiBaeg Kim, Heung-Seon Oh, Seongung Jo, and Dong Hwan Kim. 2023. Hierarchical text classification as sub-hierarchy sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 12933–12941.
- [16] Ting Jiang, Deqing Wang, Leilei Sun, Zhongzhi Chen, Fuzhen Zhuang, and Qinghang Yang. 2022. Exploiting Global and Local Hierarchies for Hierarchical Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 4030–4039.
- [17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [18] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. Hdtex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 364–371.
- [19] DavidD. Lewis, Yiming Yang, TonyG. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research, Journal of Machine Learning Research* (Dec 2004).
- [20] Xudong Mao, Yun Ma, Zhenguo Yang, Yangbin Chen, and Qing Li. 2019. Virtual Mixup Training for Unsupervised Domain Adaptation. *arXiv: Computer Vision and Pattern Recognition, arXiv: Computer Vision and Pattern Recognition* (May 2019).
- [21] Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical Text Classification with Reinforced Label Assignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. <http://dx.doi.org/10.18653/v1/d19-1042>
- [22] OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt>.
- [23] Hao Peng, Jianxin Li, Qiran Gong, Senzhang Wang, Lifang He, Bo Li, Lihong Wang, and PhilipS. Yu. 2019. Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification. *IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering* (Jun 2019).
- [24] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- [25] Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium* (Oct 2008).
- [26] Ramit Sawhney, Megh Thakkar, Shrey Pandit, Ritesh Soun, Di Jin, Diyi Yang, and Lucie Flek. 2022. DMIX: Adaptive Distance-aware Interpolative Mixup. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 606–612.
- [27] Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. HFT-CNN: Learning Hierarchical Category Structure for Multi-label Short Text Categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. <https://doi.org/10.18653/v1/d18-1093>
- [28] Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery* 22 (2011), 31–72.
- [29] Junru Song, Feifei Wang, and Yang Yang. 2023. Peer-Label Assisted Hierarchical Text Classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3747–3758.
- [30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [31] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [32] Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7109–7119.
- [33] Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhipang Sui, and Houfeng Wang. 2022. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. *arXiv preprint arXiv:2204.13413* (2022).
- [34] Jonas Wehrmann, RodrigoC. Barros, and Ricardo Cerri. 2018. Hierarchical Multi-Label Classification Networks. *International Conference on Machine Learning, International Conference on Machine Learning* (Jul 2018).
- [35] Yeming Wen, Ghassen Jerfel, Rafael Müller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. 2020. Combining ensembles and data augmentation can harm your calibration. *arXiv preprint arXiv:2010.09875* (2020).
- [36] Jiawei Wu, Wenhan Xiong, and WilliamYang Wang. 2019. Learning to Learn and Predict: A Meta-Learning Approach for Multi-Label Classification. *Cornell University - arXiv, Cornell University - arXiv* (Sep 2019).
- [37] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. 2020. Dual mixup regularized learning for adversarial domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part XXIX 16*. 540–555.
- [38] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *International Conference on Computer Vision (ICCV)*.
- [39] Hongyi Zhang, Moustapha Cisse, YannN. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond Empirical Risk Minimization. *arXiv:Learning* (Oct 2017).
- [40] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. 2022. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2461–2470.
- [41] Yanzhai Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11730–11738.
- [42] Ji Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-Aware Global Model for Hierarchical Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1106–1117.
- [43] He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. 2023. HiTIN: Hierarchy-aware Tree Isomorphism Network for Hierarchical Text Classification. *arXiv preprint arXiv:2305.15182* (2023).

A APPENDIX

A.1 Details of Comparable Models

TextRNN utilizes TextRNN to encode the input and treat HTC as a global text classification problem.

HiAGM [42] utilizes a directed graph hierarchy and hierarchy-aware structure encoders to model label dependencies. It proposes an end-to-end hierarchy-aware global model with two variants: HiAGM-LA, which learns hierarchy-aware label embeddings and performs inductive fusion of label-aware text features, and HiAGM-TP, which directly feeds text features into hierarchy encoders.

HTCInfoMax [9] proposes an information maximization approach with two modules: text-label mutual information maximization and label prior matching. The first module captures interactions between text and labels to filter irrelevant information, while the second enhances the structure encoder’s ability to represent all labels, tackling label imbalance in HTC.

HiMatch [42] introduces a hierarchy-aware label semantics matching network to formulate the text-label semantics relationship. It projects text and label semantics into a joint embedding space and utilizes a joint embedding loss and a matching learning loss to model the matching relationship.

HGCLR [32] introduces hierarchy-guided contrastive learning, embedding the label hierarchy into the text encoder by constructing positive samples based on the hierarchy for hierarchy-aware text representation.

HPT [33] introduces hierarchy-aware prompt tuning for HTC using multi-label MLM. It employs dynamic virtual templates and soft prompts with label words to integrate label hierarchy knowledge, alongside a zero-bounded multi-label cross-entropy loss to align HTC and MLM objectives.

HBGL [16] introduces Hierarchy-guided BERT with global and local hierarchies, which leverages large-scale parameters and prior language knowledge to model hierarchies. It directly models semantic and hierarchical information with BERT, avoiding the intentional fusion of separate modules.

HJCL [14] introduces hierarchy-aware joint supervised contrastive learning, which combines supervised contrastive learning with HTC. It utilizes instance-wise and label-wise contrastive learning techniques and carefully constructs batches to achieve the contrastive learning objective.

HiTIN [43] proposes hierarchy-aware tree isomorphism network to enhance text representations using only label hierarchy’s syntactic information. It converts the label hierarchy into a coding tree guided by structural entropy and incorporates hierarchy-aware information into text representations through a structure encoder.

ChatGPT represents utilizing instruction-tuned language models, such as ChatGPT. Results are directly reported from [14]. They manually design templates and use gpt-turbo-3.5 for predictions. Noted, [14] does not report the results for WOS and NYT, which is probably because the flattened hierarchical labels in the template occupy most of the tokens. Texts and labels in WOS and NYT are relatively long, making them hard to implement.

LLaMA represents utilizing fine-tuned language models, such as LLaMA. We conduct classification training using the hidden layer output of the last token of the LLaMA-2-7B. Additionally, we use LoRA to efficiently fine-tune the LLM.

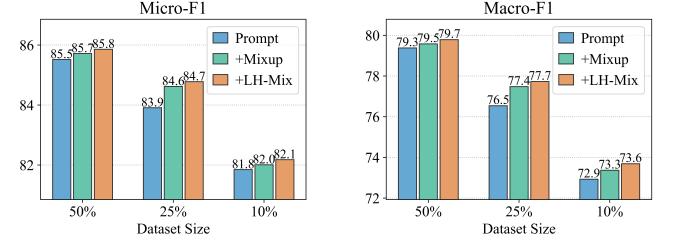


Figure 8: Results on different downsample ratios to WOS.

A.2 Performance on Sparse WOS

The performance of comparable models on sparse WOS is shown in Figure 8. We observe that LH-Mix still outperforms baseline models in different ratios.

A.3 Statistical Performance of LH-Mix

A.3.1 Performance on Reproducibility. To ensure reproducibility and robustness, we repeat the experiments five times using different random seeds for LH-Mix and comparable models. For reasons of performance and code accessibility, we specially select the two SOTA models HPT and HBGL for comparison. The average and standard deviation results are reported in Table 4. From the Table, we find that all models exhibit relatively stable performance. Moreover, LH-Mix consistently achieves optimal performance than other models overall.

A.3.2 T-test with SOTA Models. We conduct T-tests on LH-Mix compared to the two SOTA models, HPT and HBGL. The resulting P-values are presented in Table 5. As per the conventional practice, a P-value less than 0.05 indicates statistical significance. From our analysis, we observe that LH-Mix exhibits a statistically significant improvement over HPT in 5 out of the 6 metrics, and over HBGL in 4 out of 6 metrics. Combining these findings with the results discussed in Section 5.4 of our paper, we conclude that the improvement achieved by LH-Mix is statistically significant.

A.4 Comparison with GNN-based Models

We notice that numerous studies have suggested that graph encoders can obtain better hierarchical label embeddings. However, graph encoders primarily focus on modeling the global hierarchy. As outlined in our introduction, we believe that capturing the local hierarchy can also provide sufficient information. We also empirically evaluate some representative graph encoders within our framework. The results are shown in the Table 6. We find that the gain from the graph encoder is limited. Considering the extra time cost, we believe the hierarchical prompt is effective in capturing hierarchical information.

A.5 Comparison with Data Augmentation Methods

As LH-Mix is also a data augmentation method, we have now evaluated two popular data augmentation techniques for text data: Easy Data Augmentation (EDA) and Back Translation (BT). The results, which are inferior to LH-Mix, are presented in Table 7.

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
HPT	86.84 \pm 0.12	81.29 \pm 0.10	80.28 \pm 0.16	70.02 \pm 0.25	87.19 \pm 0.05	69.50 \pm 0.12
HBGL	87.18 \pm 0.14	81.69 \pm 0.26	80.28 \pm 0.14	69.78 \pm 0.14	86.97 \pm 0.14	70.54 \pm 0.42
LH-Mix	87.15 \pm 0.14	81.82 \pm 0.18	80.51 \pm 0.15	71.03 \pm 0.24	87.39 \pm 0.12	71.49 \pm 0.50

Table 4: Statistical performance on SOTA models and LH-Mix.

P-value	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
LH-Mix vs. HPT	0.0008	0.00001	0.063	0.0004	0.0032	0.00003
LH-Mix vs. HBGL	0.4902	0.2733	0.011	0.000005	0.0060	0.0041

Table 5: T-test of LH-Mix compared to HPT and HBGL.

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Prompt	86.94	81.37	80.23	70.13	87.11	69.82
+GCN	86.81	81.31	80.28	70.45	87.22	68.63
+GAT	86.91	81.20	80.30	70.65	87.16	69.95
+Graphormer	86.86	81.03	80.21	69.79	87.17	70.02

Table 6: Comparison with GNN-based Models.

Model	WOS		NYT		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Prompt	86.94	81.37	80.23	70.13	87.11	69.82
+EDA	86.64	81.24	79.73	69.34	86.98	69.22
+BT	86.63	80.98	79.77	69.13	87.00	70.75
+LH-Mix	87.27	82.07	80.57	71.26	87.47	71.93

Table 7: Comparison with GNN-based Models.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009