Hierarchical Text Classification Optimization via Structural Entropy and Singular Smoothing

Qitong Liu, Hao Peng, Xiang Huang, Zhifeng Hao, Qingyun Sun, Zhengtao Yu, and Philip S. Yu, *Fellow, IEEE*

***Abstract*—With long-tailed data and complex label hierarchy, hierarchical text classification (HTC) is a challenging multi-label text classification task. Applying prompts to pre-trained language models (PLMs) has recently become a mainstream approach in HTC. However, existing prompt-based models experience a significant drop in classification performance on tail labels. Due to the imbalanced data, HTC models still face two challenges. First, text embeddings, learned for classification, often lack distinctiveness for tail categories. Second, label embeddings suffer from significant degeneration, especially for tail labels. To address these issues, in this paper, we propose a novel Hierarchical Text Classification Optimization method via Structural Entropy and SIngular Spectrum Smoothing, namely SIHTC. SIHTC contains two parts: text embedding optimization and label embedding optimization. First, based on the structural information theory, we design a tree aggregation network and construct encoding trees to minimize the structural entropy of texts under the hierarchical labels. In this manner, SIHTC injects label structural information into text embeddings, hierarchically optimizing the embedding space by enclosing the text embeddings within related ground truth labels while separating them from unrelated ground truth labels. Second, we propose a global and local singular spectrum smoothing regularization method to maximize the area under the singular value curve. In this way, SIHTC decreases representation degeneration and learns label embeddings with improved label generalization capability. Extensive experiments are conducted on three popular HTC datasets. The results show that SIHTC outperforms all baseline methods, especially with an advantage in handling tail labels, indicating the effectiveness of the above two optimizations.**

***Index Terms*—Hierarchical Text Classification, Long-tailed Data, Structural Entropy, Singular Spectrum Smoothing.**

I. Introduction

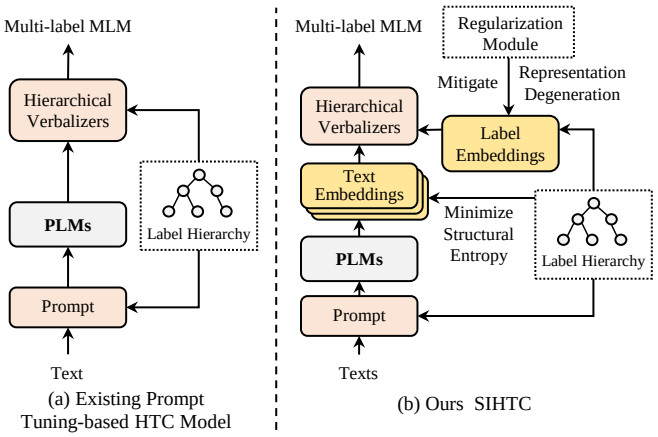


Figure 1: Illustration comparison of existing prompt tuning-based HTC model and our proposed SIHTC.

HIERARCHICAL text classification (HTC) is a specialized form of multi-label text classification where the labels have a hierarchical structure organized as a tree or a directed acyclic graph [1] [2]. The labels of a text correspond to one or more consistent, non-mandatory paths within this hierarchy [3]. Multi-label text classification data typically exhibit a long-tailed distribution, where head labels generally cover the majority of texts, while tail labels cover fewer texts [4] [5]. Apart from HTC, long-tail data is prevalent in many fields, such as relation classification [6], recommender system [7], social event detection [8] and link prediction [9], garnering increasing attention and research.

Inspired by the "in-context learning" capabilities of GPT-3 [10], an increasing number of researchers have adopted prompt tuning place of traditional fine-tuning, thereby narrowing the gap between pre-training strategies of pre-trained language models (PLMs) and downstream tasks [11] [12] [13] [14]. Consequently, research in HTC has shifted toward designing hierarchy-aware prompt tuning-based models, as illustrated in Figure 1(a). In general, existing prompt-tuning-based models [15] [16] [17] construct hierarchical templates using either soft or hard prompts to wrap the input text. This input is then fed into PLMs for the Masked Language Modeling (MLM) task. The resulting outputs, treated as text embeddings, are subsequently mapped to the final classification using hierarchical verbalizers. For example, Wang *et al.* [15] constructs learnable soft templates using a graph encoder. Building on this, Cai *et al.* [17] introduce contrastive learning with momentum update and dynamic queue methods to obtain

Qitong Liu and Xiang Huang are with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China. E-mail: {liuqt, huangxiang}@buaa.edu.cn;

Hao Peng is with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China, with the Hangzhou Innovation Institute of Beihang University, Hangzhou 310053, China, and with the School of Mathematics and Computer Science, Shantou University, Shantou 515063, China. E-mail: penghao@buaa.edu.cn;

Zhifeng Hao is with the Department of Mathematics, College of Science, Shantou University, Shantou 515063, China. E-mail: haozhifeng@stu.edu.cn;

Qingyun Sun is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: sunqy@buaa.edu.cn;

Zhengtao Yu is with the Faculty of Information Engineering and Automation, and Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China. Email: yuzt@kust.edu.cn;

Philip S. Yu is with the Department of Computer Science, University of Illinois Chicago, Chicago, IL 60607, USA. E-mail: psyu@uic.edu.

Manuscript received 12 November 2024; revised 20 May 2025; accepted 12 June 2025. (Corresponding author: Hao Peng.)

positive text samples. Xiong *et al.* [18] apply a dual prompt tuning method to capture interactions between peer labels. Ji *et al.* [16] manually design fixed hard templates and further incorporate hierarchical label knowledge into the verbalizer through probabilistic propagation.

Prompt tuning-based methods further exploit the potential of PLMs, achieving excellent performance in HTC tasks. However, existing methods still struggle with HTC's long-tailed data. They all experience a significant drop in classification performance on tail labels because, when trained on long-tailed distribution data, models tend to learn features associated with prevalent head labels while overlooking features in most tail labels with fewer samples [19]. The challenges posed by long-tailed data can be categorized into two main aspects: **1) Text Embeddings for tail labels lack distinctiveness.** Due to the scarcity of training samples, text embeddings often lack clarity and separability for tail labels, leading to confusion during classification. Most existing HTC models [20] [21] focus solely on the semantic information of the text when learning text embeddings, overlooking the guiding role of the label structure. Although Ji *et al.* [16] attempt to regulate distance between text embeddings using four superficial label relationships, this approach is too coarse-grained for complex label structures in HTC. **2) Representations of long-tail labels suffer from degeneration.** As noted in [22] [23] [24] [25], long-tail labels cause significant degradation in representation quality, often evidenced by the rapid decay of the singular spectrum of the embedding matrix. These degraded embeddings lack rich features and generalization capacity, ultimately resulting in poor classification performance for tail labels. While many current methods use techniques such as average embeddings of tokens [15], Graphormer [26], and GCN [20] to generate label embeddings, their embedding matrices exhibit rapid spectral decay——indicating severe representation degeneration. Consequently, there is an urgent need for research to effectively mitigate this degeneration problem while maintaining embedding fidelity.

To tackle the two challenges of prompt tuning-based methods, we propose a novel Hierarchical Text Classification Optimization method via Structural Information and Singular Smoothing, namely SIHTC. The framework consists of two parts: text embedding optimization using structural information theory and label embedding optimization using singular spectrum smoothing regularization, as shown in Figure 1(b). **First**, we use the label hierarchy to guide the learning of text embeddings, and we are the first to leverage structural information theory to model inter-text relationships, thereby injecting label structural information into text embeddings. Structural information theory [27] implies that minimizing the structural entropy can decode the essential information embedded in the graph. Specifically, we combine the text embeddings with their corresponding labels to form a label-text tree at each level. We then design a tree aggregation network to propagate label structure information from the bottom up across different levels, forming encoding trees. Finally, we introduce a structural entropy loss function to minimize the structural entropy of each encoding tree. Our method clusters texts around their relevant ground-truth labels and separates them from unrelated labels, hierarchically injecting label structural information into the text embeddings. In this way, we optimize the text embedding space, enhancing the distinguishability of text embeddings during classification. Previous studies have applied structural entropy to HTC. [28] minimizes the structural entropy of the label hierarchy to enable classification without relying on prior statistics or label semantics. [29] introduces a contrastive learning module that generates positive samples based on structural information theory. However, both [28] and [29] consider only the structural entropy of the label graph. In contrast, our approach minimizes the structural entropy of a joint graph constructed from both texts and labels, offering significant theoretical and practical advantages. **Second**, inspired by [24], we design a singular spectrum smoothing regularization module to alleviate representation degeneration in HTC's long-tail labels. Specifically, the regularization consists of two components: global singular value smoothing regularization and local singular value smoothing regularization, both implemented through corresponding loss functions. The global loss provides a foundational constraint for each label, while the local loss hierarchically adjusts the constraint intensity based on the label's position within the hierarchy. Building upon the nuclear norm and Frobenius norm [30] [31] [32], the regularization suppresses the largest singular value while amplifying the smaller ones. In this way, we flatten the rapidly decaying singular spectrum, alleviate representation degradation, and optimize the label embedding space.

Extensive experiments are conducted on three large-scale HTC datasets for academic paper classification [33] and news document classification [34] [35]. The results demonstrate that SIHTC outperforms current baseline methods, especially in handling tail labels. All codes of SIHTC are publicly available on GitHub[1].

In summary, the contributions of this paper are as follows:

- We propose a novel optimization method for prompt tuning-based HTC models, named SIHTC. SIHTC effectively addresses the challenges of large-scale, imbalanced data of HTC.
- Based on structural information theory, we carefully inject label structural information into text embeddings. We design a tree aggregation network and a structural entropy loss function to minimize the structural information of texts within the label hierarchy, thereby optimizing the text embedding space.
- We are the first to alleviate label representation degradation in HTC models. We propose a new global and local singular spectrum smoothing regularization method to constrain label representation and optimize the label embedding space.
- Extensive experiments are conducted on three large-scale popular HTC datasets to demonstrate our model's advantages and effectiveness.

We organize this paper as follows: Section II summarizes the relevant definitions and notation used in this paper; Section III describes the framework of the proposed SIHTC; Section IV describes the training details of SIHTC, as well

[1] https://github.com/SELGroup/SIHTC

as the datasets, baselines, and evaluation metrics employed; Section V presents comprehensive experimental results, providing an in-depth analysis of the advantages of SIHTC; Section VI reviews a series of representative works on HTC and structural information theory; and Section VII concludes the paper, discussing potential future research directions.

II. Problem Formulation and Notations

In this section, we formalize related definitions of HTC in Section II-A and the structural entropy-related definitions in Section II-B. Additionally, we summarize the notation used in this paper in Table I.

Table I: Glossary of Notations.

Symbol	Definition
$\mathcal{H}$; $\mathcal{Y}$; E	A label hierarchy; the node and edge set of $\mathcal{H}$
y; l_y; L	A label in $\mathcal{Y}$; the depth of y; the depth of $\mathcal{H}$
t; Y; Y_i	A text; t's label set; t's label set of i-th depth
G; $\mathcal{T}$	A text graph; the encoding tree of G
$\mathcal{V}$; A	The data points set and adjacency matrix of G
α; λ; γ	Node, root node, and leaf node in $\mathcal{T}$
T_α	A partition of data points corresponds to node α
$h(\alpha)$; k	The height of α; the dimension of $\mathcal{T}$
α^-	The parent node of α
g_α; $vol(\alpha)$	The cut of α; The volume of α
$H^{\mathcal{T}}(G;\alpha)$; $H^{\mathcal{T}}(G)$	The structural entropy of α and G within $\mathcal{T}$
h_i; H_T; H_i	A i-th depth text embedding; The whole and i-th depth text embedding matrix
m; M; M_i	A label embedding; the whole and i-th depth label embedding matrix
$\mathcal{T}_*^{(i)}$; $\mathcal{T}^{(i)}$	the text-label tree and encoding tree of i-th depth
$\mathcal{V}_*$; $\mathcal{X}$; $\mathcal{E}$	The nodes set; the nodes' embeddings set; the edge set of $\mathcal{T}_*$
α_t; α_y; α_{t_y}	A text node; a label node; and an aggregated node
$\mathcal{L}_{SE}$; γ_1	The structural entropy loss and it weight hyper-parameter
$\mathcal{L}_{SS}$; γ_2	The singular value smoothing regularization loss and it weight hyper-parameter

A. Hierarchical Text Classification

Hierarchical Text Classification (HTC) aims to assign a set of labels to an input text, with these labels structured in a predefined hierarchy. The formal definitions of *Label hierarchy*, *Input and output* are as follows:

Definition 1 (Label hierarchy)**.** *Label hierarchy is defined as a graph $\mathcal{H} = (\mathcal{Y}, E)$, where $\mathcal{Y}$ is the label set (also the node set of $\mathcal{H}$), and E is the hierarchical connections within the labels (also the edge set of $\mathcal{H}$). $\mathcal{H}$ is a tree structure where each node, except for the root node, has one and only one parent node. The depth of node y is denoted by l_y. For the root node, $l_y = 0$ and for the leaf nodes, $l_y = L$, where L is the depth of $\mathcal{H}$.*

Definition 2 (Input and output)**.** *The input is a set of texts split into different batches. In each batch, $B = \{t_1, t_2, \cdots, t_b\}$, where every text is treated as a sequence of tokens $t_i = \{x_1, x_2, \cdots, x_n\}$. The classification output of each text t_i is a set of labels, $Y = \{Y_1, Y_2, \cdots, Y_m\}$, where Y_i contains labels in i depth of $\mathcal{H}$. The labels in Y correspond to one or more paths in the label structure $\mathcal{H}$, from the root node to leaf or non-leaf nodes.*

B. Structural Entropy

Structural entropy is a measure of the uncertainty of a graph structure. It represents the minimum number of bits required to encode a reachable vertex during a single-step random walk on the graph. Structural information theory models input data as a graph and utilizes encoding trees to measure the graph's structure. An encoding tree that minimizes structural entropy represents the essential structure of the graph. We follow the definitions of the Encoding tree and Structural entropy as presented in [27], which are as follows:

Definition 3 (Encoding tree)**.** *Given a graph $G = (\mathcal{V}, A)$, $\mathcal{V}$ is the set of input data points, A is the adjacency matrix, and the elements in A represent the weights of edges. The encoding tree $\mathcal{T}$ for G is a hierarchical partitioning of G that can be described as follows: (1) Each tree node $\alpha \in \mathcal{T}$ corresponds to a partition of data points $T_\alpha \subseteq \mathcal{V}$. Significantly, the root node λ of $\mathcal{T}$ is associated with the entire set of data points, $T_\lambda = \mathcal{V}$. And for any leaf node γ of $\mathcal{T}$, $\mathcal{T}_\gamma$ contains exactly one data points from $\mathcal{V}$. (2) For any non-leaf node α in $\mathcal{T}$, let its children be denoted as $\beta_1, ..., \beta_{N_\alpha}$, where N_α is the number of children of α. Then, $(T_{\beta_1}, ..., T_{\beta_{N_\alpha}})$ form a partition of T_α.*

The encoding tree captures the graph's complexity and connectivity patterns. Each tree node depicts a partition of the data point set $\mathcal{V}$.

Definition 4 (Structural entropy)**.** *The structural entropy is defined under the graph G and the encoding tree $\mathcal{T}$. The structural entropy of each tree node $\alpha \in \mathcal{T}$ is as follows:*

$$H^{\mathcal{T}}(G;\alpha) = -\frac{g_\alpha}{vol(\lambda)} \log_2 \frac{vol(\alpha)}{vol(\alpha^-)}, \tag{1}$$

where the cut g_α is the weight sum of edges with exactly one endpoint in T_α. And the volume $vol(\alpha)$, $vol(\alpha^-)$, and $vol(\lambda)$ denote the degrees sum of data points within T_α, T_α^-, and T_λ, respectively. The structural entropy of $\mathcal{T}$ is equal to the sum of the entropy of all nodes, as follows:

$$H^{\mathcal{T}}(G) = \sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} H^{\mathcal{T}}(G;\alpha). \tag{2}$$

A smaller structural entropy indicates that the uncertainty of the graph G is lower.

III. Methodology

In this section, we systematically describe the framework of our proposed SIHTC, which contains two optimization methods. First, we introduce the details of the prompt tuning-based HTC model in Section III-A. Then, we describe the text embedding optimization method, which is based on structural information theory in Section III-B. Specifically, we describe the tree aggregation network in Section III-B1 and a structural entropy minimization method in Section III-B2. Second, we describe the label embedding optimization method in Section III-C. We specifically introduce the singular spectrum smoothing regularization. Finally, we describe the objective function in Section III-D.

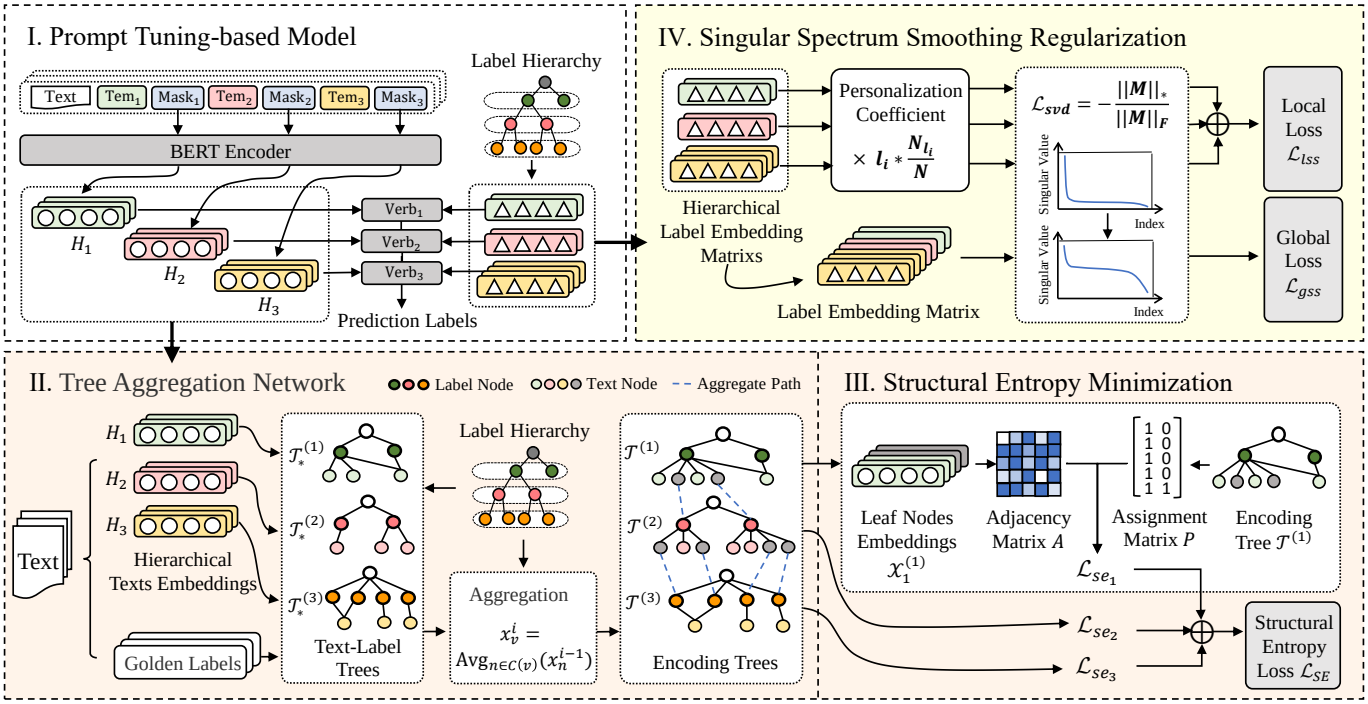


Figure 2: The proposed SIHTC framework. (I) is the framework of prompt tuning-based HTC models. (II) and (III) is the text embedding optimization based on structural information theory. (IV) is the label embedding optimization based on singular spectrum smoothing.

A. Prompt Tuning-based Model

We chose HPT [15] as the foundational model for optimization since it is the most representative prompt tuning-based HTC model. We illustrate HPT's simplified framework in Figure 2 (I). Specifically, with a label hierarchy of depth L, HPT constructs a template in the form of $[\text{Tem}_1][\text{Mask}_1]$ $[\text{Tem}_2]$ $[\text{Mask}_2]$ $\dots$ $[\text{Tem}_L]$ $[\text{Mask}_L]$, where $[\text{Tem}_i]$ encapsulates the information from the i-th depth within the label hierarchy, and $[\text{Mask}_i]$ is used to predict the labels of the text at the i-th depth. Then, HPT packages the text "x" as "$x, template$" and feeds it into BERT for the MLM task. For a batch of texts B, BERT outputs the final hidden states at each mask position, which are the first optimization targets in our proposed SIHTC. We refer to those outputs as text embeddings, the formal definition of which is as follows:

Definition 5 (Text embeddings)**.** *Given a batch of texts B, B's text embedding is denoted as $H_T = \{H_i | i \in [1, L]\}$, where $H_i \in \mathbb{R}^{b \times r}$ is the BERT's output corresponding to $[\text{Mask}_i]$ tokens of all texts in B. For a text t in B, t's text embedding is denoted as $h = \{h_i | i \in [1, L]\}$ which is the set of hidden state vectors corresponding to L $[\text{Mask}]$ tokens of t, where $h_i \in \mathbb{R}^r$ and r is the hidden state dimension of BERT. H_i is composed of all h_i of texts in B.*

Then, HPT utilizes L different hierarchical verbalizers $[\text{Verb}_1][\text{Verb}_2] \dots [\text{Verb}_L]$ to process text embeddings, generating classification results for each text. HPT constructs the verbalizers with label embeddings, which are the second optimization targets in our proposed SIHTC. We formally define label embeddings as follows:

Definition 6 (Label embeddings)**.** *Given a label y, the label embedding of y is the learnable virtual label word $m \in \mathbb{R}^r$ in the HPT's verbalizer. M represents the entire label embedding matrix, and M_i represents the i-th depth label embedding matrix of the label hierarchy.*

B. Text Embedding Optimization

We first design a tree aggregation network to construct hierarchical two-dimensional encoding trees. Following this, we introduce the structural entropy loss function, which optimizes text embeddings by minimizing the structural entropy of the encoding trees. Our method effectively injects label structural information into the text embeddings. Next, we will describe the tree aggregation network and the structural entropy minimization process separately.

1) ***Tree Aggregation Network****:* Previous works ignore the guidance of label structures on text embeddings. The Tree Aggregation Network (TAN) aims to integrate text information and label structure information, as shown in Figure 2(II). During model training, the input to TAN includes the label hierarchy, text embeddings, and ground truth labels. The output of TAN is a set of two-dimensional encoding trees. We present the algorithm for TAN in Algorithm 1. Each encoding tree corresponds to a depth of the label hierarchy. TAN contains two steps: text-label tree construction and inter-level aggregation, where the first step integrates the intra-level information of label structure, and the second step integrates the inter-level information.

Text-label trees construction. In this step, we construct multiple text-label trees, each corresponding to a depth of the label hierarchy. The goal is to partition the text embeddings

according to their ground truth labels. To facilitate the explanation, we first define the text-label tree as follows:

Definition 7 (Text-label tree)**.** *A text-label tree is a height-2 tree that consists of a root node, label nodes, and text nodes, represented as* $\mathcal{T}_* = \{\mathcal{V}_*, \mathcal{X}, \mathcal{E}\}$*. In this structure,* $\mathcal{V}_* = \{\mathcal{V}_0, \mathcal{V}_1, \mathcal{V}_2\}$ *is the level-wise nodes set, where* $\mathcal{V}_0$ *is the root node set,* $\mathcal{V}_1$ *is the label node set, and* $\mathcal{V}_2$ *is the text node set.* $\mathcal{X} = \{\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2\}$ *is the corresponding node embedding set, and* $\mathcal{E}$ *is the edge set between nodes.*

First, we initialize the text-label trees. Given a label hierarchy $\mathcal{H}$ with a depth of L, we construct a text-label tree set $T_* = \{\mathcal{T}_*^{(1)}, \mathcal{T}_*^{(2)}, \ldots, \mathcal{T}_*^{(L)}\}$, where $\mathcal{T}_*^{(i)}$ corresponds to the i-th depth of $\mathcal{H}$. We use $\mathcal{T}_*^{(i)} \in T$ as an example to introduce the construction details of a text encoding tree. $\mathcal{T}_*^{(i)}$ only partitions $H_i \in H_T$, the text embedding matrix corresponding to the i-th mask token. At the beginning, $\mathcal{T}_*^{(i)}$ is initialized to contain only the root node, i.e. $\mathcal{V}_0 = \{\lambda\}$ and the other sets of $\mathcal{T}_*^{(i)}$ are empty sets. Then, we create the label nodes. We add the appropriate label nodes as the children of the root node. Formally,

$$\mathcal{V}_1 = \{\alpha_y | l_y = i\}, \mathcal{E} = \{(\lambda, \alpha_y) | \alpha_y \in \mathcal{V}_1\}, \tag{3}$$

where α_y denotes a label node corresponding to label y which at the i-th depth of $\mathcal{H}$. Finally, we create the text nodes. During model training, a text t has a hierarchical text embedding h_i and a ground truth label set Y_i at the i-th depth of $\mathcal{H}$. We create a new text node α_t for t in $\mathcal{T}_*^{(i)}$ and add it as a child of the ground truth label nodes. Formally,

$$\mathcal{V}_2 = \mathcal{V}_2 \cup \{\alpha_t\}, \mathcal{E} = \mathcal{E} \cup \{(\alpha_y, \alpha_t) | y \in Y_i\}, \mathcal{X}_2 = \mathcal{X}_2 \cup \{h_i\}, \tag{4}$$

where α_y is the label node in $\mathcal{V}_1$ which corresponds to t's ground truth label $y \in Y_i$, and h_i is the text embedding of t in i-th depth, also the the embedding of the text node α_t. Notably, since a text in HTC may belong to multiple paths of $\mathcal{H}$, Y_i may contain various elements, so a text node may have more than one parent label node. We repeat the process described in Equation 4 for all texts in a batch B. Once we complete all texts, we obtain the full text-label tree $\mathcal{T}_*^{(i)}$. Following the above process, we can construct all text encoding trees in T_*, completing the partition of all text embeddings $H_T = \{H_i | i \in [1, L]\}$.

Inter-level aggregation. The text-label trees complete the partition but ignore a parent-child relationship between different depth labels in the label hierarchy. Therefore, we introduce an aggregation operation to bridge the L trees, integrating inter-level label structure information into the text embeddings.

Specifically, We aggregate the L text-label trees from the bottom up based on the label structure. We use the aggregation from $\mathcal{T}_*^{(i)}$ into $\mathcal{T}_*^{(i-1)}$ as an example to introduce the details. Before aggregation, in $\mathcal{T}_*^{(i)}$, only text nodes have embedding, while label node embedding set $\mathcal{X}_1$ is empty till now. Thus, we first aggregate the text nodes into the label nodes for preparation. Formally, for each label node α_y in $\mathcal{V}_1$,

$$\mathcal{X}_1^{(i)} = \mathcal{X}_1^{(i)} \cup \{x_y\}, x_y = \text{average}_{\alpha_t \in C(\alpha_y)}(x_t), \tag{5}$$

Algorithm 1: Proposed TAN.

Input: Label hierarchy $\mathcal{H}$ with depth L, text embedding set $\{h\}$ and ground truth labels set $\{Y\}$ of batch B.
Output: A set of two-dimensional encoding trees: T.

```
/* Text-label trees construction.   */
for i = 1 to L do
    Initialize T_*^(i) with a root node λ;
    Create label nodes of T_*^(i) via Eq. 3;
    for text t in B do
        Create text nodes of T_*^(i) via Eq. 4;
    end
    T_* ← {T_*^(1), T_*^(2), ··· , T_*^(L)};
end
/* Inter-level aggregation.         */
for i = L to 2 do
    for label node α_y in T_*^(i) do
        if α_y has child text nodes then
            T_*^(i).X_1 ← Eq. 5;
        else
            Remove α_y from T_*^(i).V_1;
        end
    end
    /* Aggregation T_*^(i) to T_*^(i-1).   */
    Create new text nodes of T_*^(i-1) via Eq. 7, 9;
    Aggregate embeddings to T_*^(i-1) via Eq. 8;
end
return T = {T^(1), ··· , T^(L)} ← {T_*^(1), ··· , T_*^(L)}
```

where x_y is the embedding of label node α_y, $C(v_y)$ is the children set of α_y, i.e. α_t is a text node which has a edge (α_y, α_t) in $\mathcal{E}^{(i)}$, and x_t is the embedding of α_t. Notably, if a label node has no child text nodes, we remove it from the label nodes set. Formally,

$$\mathcal{V}_1^{(i)} = \mathcal{V}_1^{(i)} \backslash \{\alpha_{y'} | C(\alpha_{y'}) = \emptyset\}, \tag{6}$$

where $\alpha_{y'}$ is a label node with no children nodes. Then, we start the upward aggregation to $\mathcal{T}_*^{(i-1)}$. Specifically, we create text nodes in $\mathcal{T}_*^{(i-1)}$ corresponding to label nodes in $\mathcal{T}_*^{(i)}$. Formally,

$$\mathcal{V}_2^{(i-1)} = \mathcal{V}_2^{(i-1)} \cup \{\alpha_{t_y} | \alpha_y \in \mathcal{V}_1^{(i)}\}, \tag{7}$$

$$\mathcal{X}_2^{(i-1)} = \mathcal{X}_2^{(i-1)} \cup \{x_{t_y} | \alpha_y \in \mathcal{V}_1^{(i)}\}, x_{t_y} = x_y \in \mathcal{X}_1^{(i)}, \tag{8}$$

where α_{t_y} is a new text node in $\mathcal{T}_*^{(i-1)}$ which represents the union of all text nodes belong to α_y in $\mathcal{T}_*^{(i)}$, and x_{t_y} is the embedding of α_{t_y} which is equal to the embedding of α_y in $\mathcal{T}_*^{(i)}$. Then, based on the label hierarchy $\mathcal{H}$, we connect these newly created nodes to their parent nodes. Formally,

$$\mathcal{E}^{(i-1)} = \mathcal{E}^{(i-1)} \cup \{(\alpha_{y'}, \alpha_{t_y}) | \alpha_{y'} \in \mathcal{V}_1^{(i-1)}, (y', y) \in E\}, \tag{9}$$

where E is the edge set of $\mathcal{H}$, y' is the parent label of y in $\mathcal{H}$, and $\alpha_{y'}$ is the label node in $\mathcal{T}_*^{(i-1)}$ corresponds to y'. At

this point, we complete the aggregation from $\mathcal{T}^{(i)}$ to $\mathcal{T}^{(i-1)}$. Refer to this example, repeating Equations 5, 6, 7, 8, 9, we can sequentially complete the aggregation from $\mathcal{T}^{(L)}$ to $\mathcal{T}^{(L-1)}$, $\mathcal{T}^{(L-1)}$ to $\mathcal{T}^{(L-2)}$ and so on, until $\mathcal{T}^{(2)}$ to $\mathcal{T}^{(1)}$.

After the aggregation, we obtain more comprehensive trees. Each tree now accounts not only for the text embeddings of its level but also for the text embeddings of all descendants. We treat the aggregated text-label trees as two-dimensional encoding trees, denoted as $T = \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(L)}\}$. These encoding trees integrate text information and label structure information.

*2) **Structural Entropy Minimization**:* We minimize the structural entropy of the two-dimensional encoding trees obtained in Section III-B1 by designing a structural entropy loss function $\mathcal{L}_{SE}$, as shown in Figure 2(III). We use $\mathcal{T}^{(i)} \in T$ as an example to introduce the design of $\mathcal{L}_{se_i}$. Since structural information theory is defined based on the partition of a graph, we construct a graph that contains all the text nodes of $\mathcal{T}^{(i)}$. Specifically, a graph G_i for $\mathcal{T}^{(i)}$ is construct as follows:

$$A = \sigma(X \times X^T), \tag{10}$$

where $X \in \mathbb{R}^{|\mathcal{V}_2| \times r}$ is an embedding matrix composed of all text nodes embedding in $\mathcal{X}_2$ of $\mathcal{T}^{(i)}$, and σ denotes the sigmoid activation function, which ensures that the elements in the adjacency matrix A are positive. According to the definition of the encoding tree, label nodes of $\mathcal{T}^{(i)}$ serve as partitions of the text nodes. Therefore, we constructed an assignment matrix $P \in \{0,1\}^{|\mathcal{V}_2| \times |\mathcal{V}_1|}$ for G_i, where $|\mathcal{V}_2|$ is the number of text nodes and $|\mathcal{V}_1|$ is the number of label nodes in $\mathcal{T}^{(i)}$. $P_{jk} = 1$ if the j-th text node belongs to the k-th label node in $\mathcal{T}^{(i)}$. During the model training, P remains constant. Then, we propose a loss function $\mathcal{L}_{se_i}$ to minimize the structural entropy of label nodes in $\mathcal{T}^{(i)}$, in other words, to reduce the uncertainty of G_i under fixed partitions. For a two-dimensional encoding tree $\mathcal{T}^{(i)}$ and corresponding G_i, the sum of the structural entropy of the label nodes is defined as follows [27]:

$$H^{\mathcal{T}^{(i)}}(G_i) = \sum_{y=1}^{|\mathcal{V}_1|} -\frac{g_{\alpha_y}}{vol(\lambda)} \log_2 \frac{vol(\alpha_y)}{vol(\lambda)}, \tag{11}$$

where α_y is a label node in $\mathcal{V}_1$, g_{α_y} is the weight sum of cut edges of the partition corresponding to α_y, and $vol(\alpha_y)$ is the degrees sum of text nodes in the partition corresponding to α_y. It is worth noting that, unlike traditional encoding trees in [27], our encoding tree may contain nodes with multiple parent nodes, as shown in Figure 3(A). We consider such nodes to belong to various partitions simultaneously, as illustrated in Figure 3(B). Therefore, the volume and cut of partition α_y in G_i can be calculated as follows:

$$vol(\alpha_y) = (\{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|} AP)_{yy}, \tag{12}$$

$$g_{\alpha_y} = (\{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|} AP)_{yy} - (P^T AP)_{yy}, \tag{13}$$

where A and P are adjacency matrix and assignment matrix of G_i, respectively, and $(\cdot)_{yy}$ indicates the element in the matrix

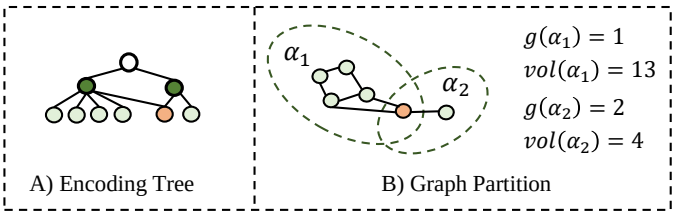


Figure 3: The encoding tree and its corresponding graph partition. The yellow node has more than one parent node, and it belongs to both partition α_1 and partition α_2.

at the y-th row and y-th column. We proposed the structural entropy loss function $\mathcal{L}_{se_i}$ as follows:

$$\begin{aligned}
&\mathcal{L}_{se_i} = H^{\mathcal{T}^{(i)}}(G_i) \\
&= \sum_{y=1}^{|\mathcal{V}_1|} \frac{((P^T - \{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|})AP)_{yy}}{\text{sum}(A)} \log_2 \frac{(\{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|} AP)_{yy}}{\text{sum}(A)} \\
&= \text{trace}(\frac{(P^T - \{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|})AP}{\text{sum}(A)} \odot \log_2 \frac{\{1\}^{|\mathcal{V}_1| \times |\mathcal{V}_2|} AP}{\text{sum}(A)}),
\end{aligned} \tag{14}$$

where trace$(\cdot)$ is an operation that sums up the diagonal elements of the matrix, sum$(\cdot)$ is an operation that sums up all elements in a matrix, and $\odot$ is the Hadamard product. We use matrix operations instead of summation operations, which can significantly improve computational efficiency.

We compute the loss following the Equation 14 for each encoding tree in T and sum them together, resulting in the overall structural entropy loss $\mathcal{L}_{SE}$ as follows:

$$\mathcal{L}_{SE} = \sum_{i=1}^{L} \mathcal{L}_{se_i}, \tag{15}$$

where $\mathcal{L}_{se_i}$ is the structural entropy of $\mathcal{T}^{(i)} \in T$. During the model training process, A and the text embeddings are updated to achieve a smaller structural entropy as the $\mathcal{L}_{SE}$ decreases. In the same time, $\mathcal{L}_{SE}$ optimizes the text embedding space by enclosing the text embeddings on related labels while separating those on unrelated labels.

C. Label Embedding Optimization

The long-tailed distribution of HTC data leads to representation degeneration in label embeddings, which previous works ignore. One manifestation of representation degeneration is the rapid decay of the singular spectrum of the embedding matrix, i.e., the largest singular value is significantly greater than the rest. [24] is the first to propose a singular spectrum smoothing regularization method in sequence recommendation to alleviate the degeneration of the user sequence and item embeddings. Based on [24], we propose a singular spectrum smoothing regularization loss to optimize label embeddings in HTC, as shown in Figure 2(IV). The loss contains two parts: a global part for macro-control and a local part for hierarchical adaptation.

*1) **Global Singular Spectrum Smoothing Regularization Loss**:* In this section, we apply regularization constraints uniformly to all label representations. By suppressing the largest singular value and encouraging the sum of singular values of the embedding matrix, we transform the singular spectrum curve from a rapid decay, indicative of degeneration, to a smoother distribution. Formally,

$$\mathcal{L}_{gss} = -\frac{\|M\|_*}{\|M\|_F}, \tag{16}$$

$$\|M\|_* = \sum_{i=1}^{\min(n,r)} \sigma_i, \|M\|_F = \sqrt{\sum_{i=1}^{\min(n,r)} \sigma_i^2}, \tag{17}$$

where $M \in \mathbb{R}^{n \times r}$ is the label embedding matrix, n is the number of labels in $\mathcal{H}$, and r is the dimension of label embedding. $\|\cdot\|_*$ is the nuclear norm and $\|\cdot\|_F$ is the Frobenius norm of of the matrix, σ is a singular value of M. During training, the increase of the nuclear norm indicates an increase in the sum of singular values, and the decrease of the Frobenius norm implies a reduction in the largest singular value. In summary, the $\mathcal{L}_{gss}$ helps to obtain smoother singular spectrum curves.

*2) **Local Singular Spectrum Smoothing Regularization Loss**:* The global loss alleviates representation degeneration by smoothing the singular spectrum of the embedding matrix formed by all labels. However, in HTC's label hierarchy, the degree of degeneration varies for labels located at different positions. Intuitively, labels at deeper depths suffer from more severe degeneration as they have fewer training samples, making their embeddings harder to distinguish. Therefore, we propose personalized coefficients to differentiate the regularization constraint across different levels of labels. Formally,

$$\mathcal{L}_{lss} = \sum_{i=1}^{L} -\frac{in_i}{n} \frac{\|M_i\|_*}{\|M_i\|_F}, \tag{18}$$

where n_i is the number of labels at the i-th depth, n is the number of total labels, and $M_i \in \mathbb{R}^{n_i \times r}$ is the hierarchical label embedding matrix corresponds to the i-th depth of $\mathcal{H}$. The personalized coefficients achieve two objectives: (1) to apply more substantial regularization constraints to tail labels at deeper levels to suppress degradation and apply relatively weaker constraints to shallow levels labels to retain their rich information, and (2) to normalize the varying number of labels across different levels of the label hierarchy.

Therefore, the singular value smoothing regularization loss is the sum of the above two components. Formally,

$$\mathcal{L}_{SS} = \mathcal{L}_{gss} + \mathcal{L}_{lss}. \tag{19}$$

D. Objective Function

With $\mathcal{L}_{SE}$ minimizes the structural entropy of the text embeddings, $\mathcal{L}_{SS}$ performs singular spectrum smoothing on the label embeddings, the final objective loss function to minimize is defined as follows:

$$\mathcal{L} = \mathcal{L}_{base} + \gamma_1 \mathcal{L}_{SE} + \gamma_2 \mathcal{L}_{SS}, \tag{20}$$

where $\mathcal{L}_{base}$ is the loss function of the based prompt tuning HTC model, γ_1 is the hyper-parameter for controlling the text embeddings optimization weight, and γ_2 is the hyper-parameter for controlling the label embeddings optimization weight.

IV. Experimental Setup

This section provides a comprehensive description of the experimental setup for this study. First, we introduce the datasets of the experiments in Section IV-A. Then, we detail the comparison baselines in Section IV-B. Furthermore, we describe the implementation details of the experiments in Section IV-C, including parameter settings of our proposed SIHTC, implementation for SIHTC, and the software and hardware environment. Lastly, we introduce the evaluation metrics used in the experiments in Section IV-D.

A. Dataset

We conduct experiments on three datasets for hierarchical multi-label text classification to evaluate the performance and effectiveness of our SIHTC. We show the statistics of datasets in Table II. WOS [33] includes abstracts of published papers from the Web of Science. RCV1-v2 [34] is news classification corpora from Reuters, Ltd while NYTimes [35] is news classification corpora from New York Times. Each text in these datasets is annotated with ground truth labels within the label hierarchy. In the WOS dataset, these labels contain only a single path, and in the RCV1-v2 and NYT datasets, they contain multiple paths. We split and preprocess these datasets following [20]. All paths are consistent and non-mandatory.

Table II: Statistics of the three HTC datasets. Depth is the maximum level of label hierarchy. $|\mathcal{Y}|$ is the number of labels. Avg($|Y|$) is the average number of corresponding labels per text.

Dataset	Depth	$\|\mathcal{Y}\|$	Avg($\|Y\|$)	# Train	# Dev	# Test
WOS	2	141	2.0	30070	7518	9397
RCV1-v2	4	103	3.24	20833	2316	781265
NYTimes	8	166	7.6	23345	5834	7292

B. Baselines

To validate the superiority of our proposed SIHTC, we compare it against ten strong baselines belonging to four categories. We introduce these baselines separately as follows:

- **Four hierarchy-aware models.** TextRCNN [36] is a typical text classification model that uses an RNN network structure to extract text features. HiAGM [20] is a hierarchical text classification model with a dual-encoder structure. HTCInfoMax [37] introduces text-label mutual information maximization and label prior matching, applying statistical constraints to label representations. HiMatch [21] is a semantic matching model for hierarchical text classification.
- **Two pre-trained language Models.** Bert [38] is a PLM designed for natural language processing tasks. It combines the Transformer architecture with the MLM objective to generate text embeddings. HGCLR [26] constructs positive

Table III: The experimental results (%) of our proposed method compared to baselines on three datasets. The best results are **bolded**, and the second-best results are underlined. $\Delta_{\text{HPT}}(Abs.\%)$ denotes the improvement of absolute value and percentage achieved by our model compared to the based model HPT.

Model	WOS		RCV1-v2		NYTimes		Average	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Hierarchy-Aware Models								
TextRCNN [20]	83.55	76.99	81.57	59.25	70.83	56.18	78.65	64.14
HiAGM [20]	85.82	80.28	83.96	63.35	74.97	60.83	81.58	68.15
HTCInfoMax [37]	85.58	80.05	83.51	62.71	-	-	-	-
HiMatch [21]	86.20	80.53	84.73	64.11	-	-	-	-
Pretrained Language Models (BERT-based)								
BERT [21]	86.26	80.58	86.26	67.35	-	-	-	-
BERT [26]	85.63	79.07	85.65	67.02	78.24	65.62	83.17	70.57
BERT+HiAGM [26]	86.04	80.19	85.58	67.93	78.64	66.76	83.42	71.67
BERT+HTCInfoMax [26]	86.30	79.97	85.53	67.09	78.75	67.31	83.53	71.46
BERT+HiMatch [26]	86.70	81.06	86.33	68.66	-	-	-	-
HGCLR [26]	87.11	81.20	86.49	68.31	78.86	67.96	84.15	72.49
Structural Entropy-based Models								
TextRCNN+HiTIN [28]	86.66	81.11	84.81	64.37	75.13	61.09	82.20	68.86
BERT+HiTIN [28]	87.19	81.57	86.71	69.95	79.65	69.31	84.52	73.61
HILL [29]	87.28	81.77	87.31	70.12	80.47	69.96	85.02	73.95
Pretrained Language Models with Prompt Tuning								
HPT [15]	87.16	81.93	87.26	69.53	80.42	70.42	84.95	73.96
DPT [18]	87.25	81.51	87.76	**70.78**	80.56	70.28	85.19	74.19
SIHTC (Ours)	**87.65±0.14**	**82.02±0.20**	**87.83±0.09**	70.36±0.38	**80.84±0.07**	**70.87±0.13**	**85.44**	**74.42**
$\Delta_{\text{HPT}}(Abs.\%)$	↑ .49(.56%)	↑ .09(.11%)	↑ .57(.65%)	↑ .83(1.19%)	↑ .42(.52%)	↑ .45(.64%)	↑ .49(.58%)	↑ .46(.62%)

samples for texts, implementing a hierarchical-guided contrastive learning model based on Bert.

- **Two structural entropy-based Models.** HiTIN [28] is a memory-efficient model that does not rely on prior statistics or label semantics. HILL [29] is an information lossless contrastive learning model for HTC. Both baselines are based on structural information theory, but consider only the structural entropy of the label graph. In contrast, our approach accounts for the structural entropy of both texts and labels, offering a significant advantage.
- **Two prompt tuning-based pre-trained language Models.** HPT [15] is a typical model for HTC based on prompt tuning, which constructs learnable soft prompt templates using label information. DPT [18] is a dual prompt tuning model that distinguishes features among peer labels by performing contrastive learning at each hierarchical level.

C. Implementation Details

We select the prompt-tuning-based HPT model [15] as the base model for optimization, integrating our proposed SIHTC method. We keep the relevant parameters of the HPT as originally configured. We set the weight hyperparameters $[\gamma_1, \gamma_2]$ of $\mathcal{L}_{se}$ and $\mathcal{L}_{ss}$ in Equation 20 to $[5e^{-2}, 5e^{-2}]$ for WOS, and $[1e^{-1}, 5e^{-3}]$ for RCV1-v2 and $[1e^{-2}, 5e^{-3}]$ for NYTimes dataset. We set the training epoch number to 30 and uniformly set the batch size to 32. We take the main results as the average of six random experiments. SIHTC is implemented in a software environment of Python 3.9 and PyTorch 1.13 and executed on a hardware environment of NVIDIA A800-SXM4-80GB.

D. Evaluation Metrics

We select two widely used evaluation metrics of HTC: Macro-F1 and Micro-F1 [39]. Macro-F1 computes the unweighted average of the F1-score for each class so that it treats each class equally, regardless of the sample size for each class, making it well-suited for evaluating imbalanced classification problems. Micro-F1 aggregates true positives, false positives, and false negatives across all classes before calculating the F1 score so that it gives equal weight to all instances in the averaging process, making it more sensitive to the performance of larger classes. Due to the large-scale, imbalanced data of HTC, we focused more on the performance of the Macro-F1 in our experiments, as it prefers accuracy for tail classes.

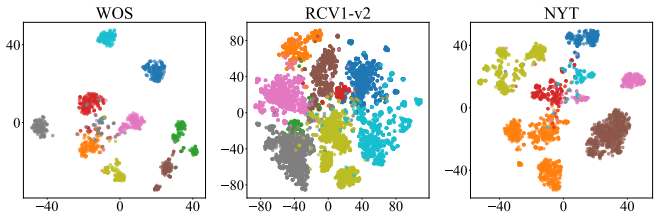
V. Experimental Results

We conduct comparative studies and effectiveness studies and detail the main results in Section V-A. We conduct ablation studies for the components of SIHTC in Section V-B. To explore SIHTC's capability in handling long-tail labels, we conduct long-tail hierarchy studies in Section V-C. Furthermore, we conduct hyperparameter studies and case studies in Section V-D and Section V-E. Finally, we supplement the computational cost in Section V-F.

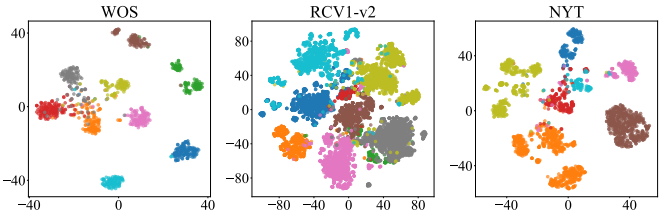
A. Main Result

1) Classification Performance of SIHTC: We provide the experimental results for the effectiveness in Table III. SIHTC outperforms the base model HPT across all three datasets and, except for Macro-F1 on the RCV1-v2 dataset, SIHTC surpasses all advanced baselines. On the WOS dataset, compared to HPT, our proposed SIHTC improves 0.49% and 0.09%

absolute Micro-F1 and Macro-F1, respectively. Additionally, it outperforms the most advanced baseline by 0.40% in Micro-F1. On the RCV1-v2 dataset with a label depth of 4, SIHTC significantly outperforms HPT, with improvements of 0.57% in Micro-F1 and 0.83% in Macro-F1. However, compared to the most advanced baseline, there is a slight decline in Macro-F1. On the NYT dataset, compared to HPT, SIHTC improves by 0.42% in Micro-F1 and 0.45% in Macro-F1. Furthermore, SIHTC achieves an additional 0.28% improvement in Micro-F1 compared to the most advanced baseline. Overall, across the three datasets, SIHTC achieves an average improvement of 0.49% in Micro-F1 and 0.46% in Macro-F1 compared to HPT and outperforms all baselines. Without altering the base model's network architecture or introducing additional parameters, SIHTC achieved a surprising performance boost. Moreover, the more significant improvement in Macro-F1 indicates that SIHTC is better for handling tail labels in HTC.



(a) Visualization of text embeddings trained from HPT.



(b) Visualization of text embeddings trained from SIHTC.

Figure 4: 2D Visualization of text embeddings. Different colors represent texts belonging to different labels.

2) Effectiveness of Structural Entropy Loss: In the three datasets, we select a series of samples belonging to single-path labels and visualize their text embeddings to intuitively demonstrate the effectiveness of our structural entropy loss $\mathcal{L}_{SE}$ on text embeddings. Figure 4(a) shows the visualization of text embeddings from HPT, while Figure 4(b) from SIHTC. Compared to HPT, the embeddings trained by SIHTC exhibit clearer cluster boundaries and better discriminative ability. Furthermore, SIHTC demonstrates a greater advantage for texts with tail labels. This is attributed to our $\mathcal{L}_{SE}$, which guides the model in learning an encoding tree with a lower structural entropy. Given the fixed partition matrix, i.e., the ground truth label assignments for texts, the only way to reduce the structural entropy of the encoding tree is to cluster embeddings of texts with the same labels while separating those with different labels.

3) Effectiveness of Singular Spectrum Smoothing Regularization Loss: We visualize the label embeddings in the three datasets to intuitively demonstrate the effectiveness of our singular spectrum smoothing regularization loss $\mathcal{L}_{SS}$. First, we compare the singular spectrum curves of the label embedding matrices trained by HPT and SIHTC. We provide the experimental results in Figure 5. The curve corresponding to HPT shows a trend of rapid decay, with the largest singular value significantly exceeding the others, indicating that label embeddings suffer from degeneration. Compared to HPT, the curve corresponding to SIHTC is smoother and has a larger Area Under the Curve (AUC), which is attributed to the effect of the $\mathcal{L}_{SS}$. During training, the Frobenius norm term $\|\cdot\|_F$ in $\mathcal{L}_{SS}$ suppresses the largest singular value, while the nuclear norm term $\|\cdot\|_*$ promotes the sum of singular values. Additionally, we visualize the SVD projection of the label embeddings to further demonstrate the benefits of a smoother singular spectrum. As noted in [22], embeddings with better generalization capability are more diversely distributed around the origin in their SVD projections, whereas degraded embeddings are confined in a narrow cone. Figure 6(a) shows that the SVD projection corresponding to HPT is distributed away from the origin in a clustered pattern. The degeneration of label embeddings is particularly severe in the WOS and RCV1-v2 datasets. In contrast, Figure 6(b) demonstrates that the label embeddings trained by SIHTC, especially for tail labels, effectively avoid degeneration. This improvement is attributed to the $\mathcal{L}_{SS}$, which reshapes the distribution of label embeddings by regularizing their singular value spectrum.

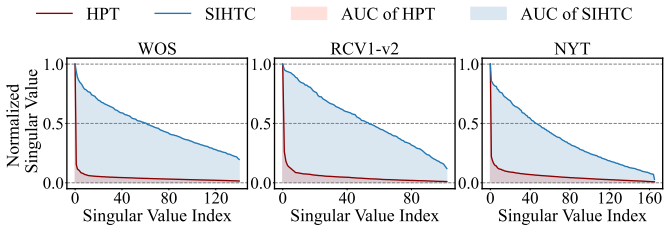
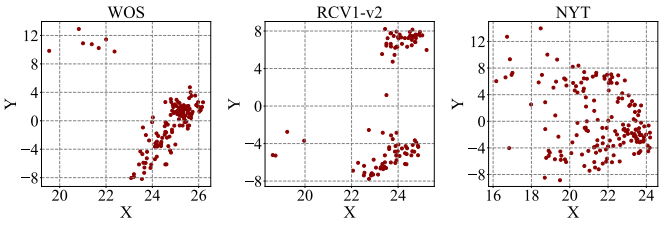
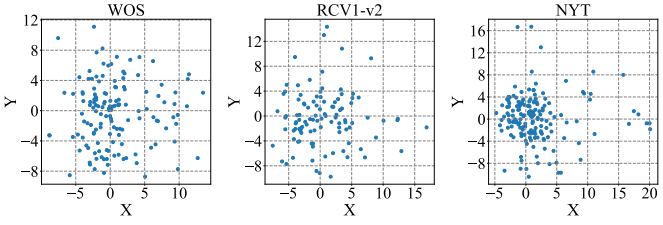


Figure 5: Singular spectrum curve of label embeddings. The shadow represents the area under the curve.



(a) Visualization of label embeddings trained from HPT.



(b) Visualization of label embeddings trained from SIHTC.

Figure 6: 2D Visualization of label embeddings. Red and blue represent the results of HPT and SIHTC, respectively.

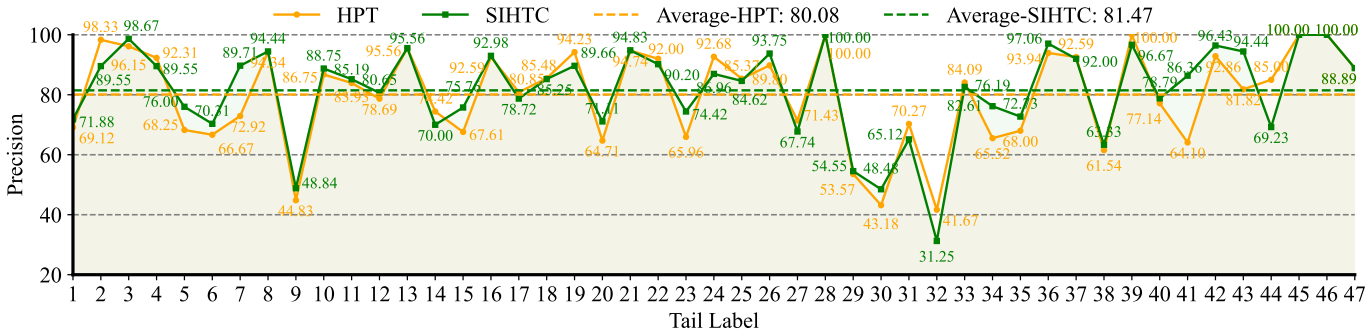


Figure 7: The classification Precision on tail labels of the WOS dataset.

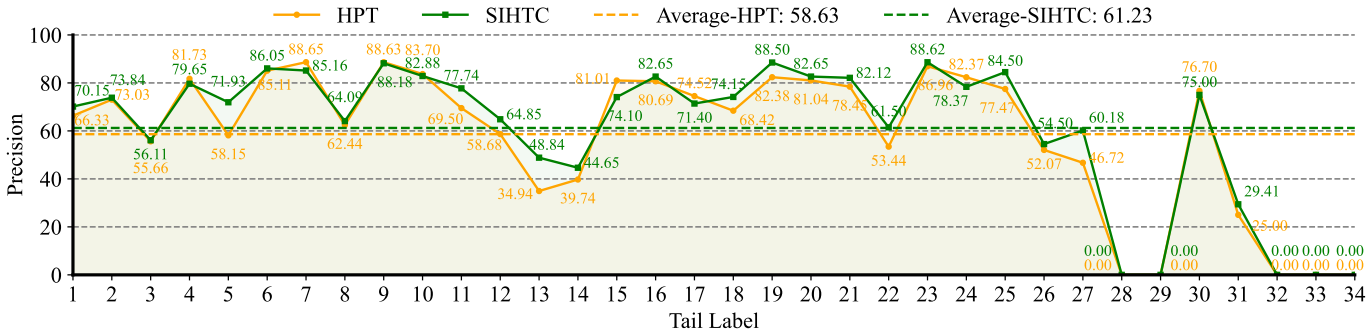


Figure 8: The classification Precision on tail labels of the RCV1-v2 dataset.

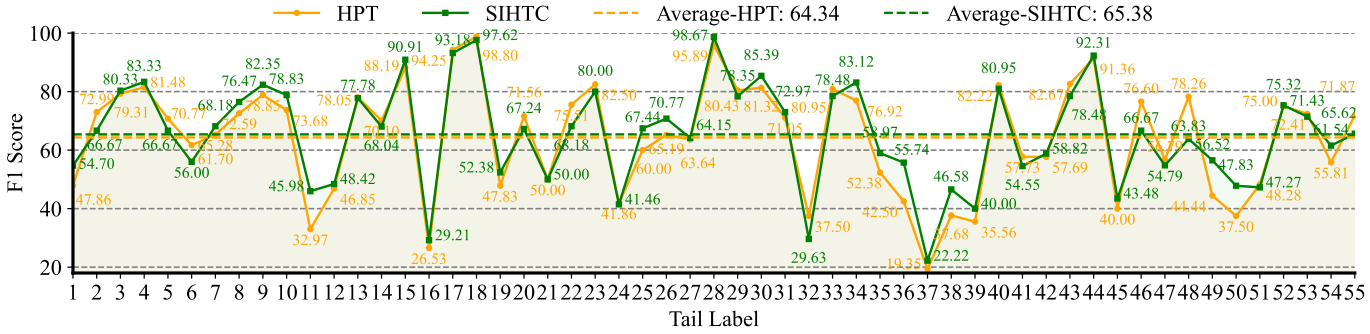


Figure 9: The classification F1 scores on tail labels of the NYT dataset.

Table IV: Ablation Study for three loss functions of SIHTC on Micro-F1. The best results are bolded.

Ablation Models	WOS	RCV1-v2	NYTimes	Average
SIHTC	**87.65**	**87.83**	80.84	**85.44**
r.m. $\mathcal{L}_{SE}$	87.37	87.75	80.75	85.29
r.m. $\mathcal{L}_{gss}$	87.59	87.51	**80.92**	85.34
r.m. $\mathcal{L}_{lss}$	87.53	87.81	80.65	85.33

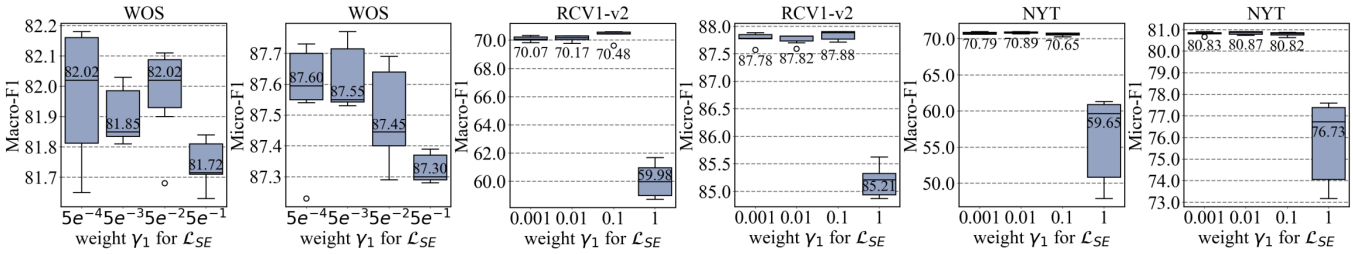
Table V: Ablation Study for three loss functions of SIHTC on Macro-F1. The best results are bolded.

Ablation Models	WOS	RCV1-v2	NYTimes	Average
SIHTC	**82.02**	**70.36**	**70.87**	**74.42**
r.m. $\mathcal{L}_{SE}$	81.86	69.75	70.81	74.14
r.m. $\mathcal{L}_{gss}$	81.75	69.92	70.55	74.07
r.m. $\mathcal{L}_{lss}$	81.61	69.85	70.42	73.96

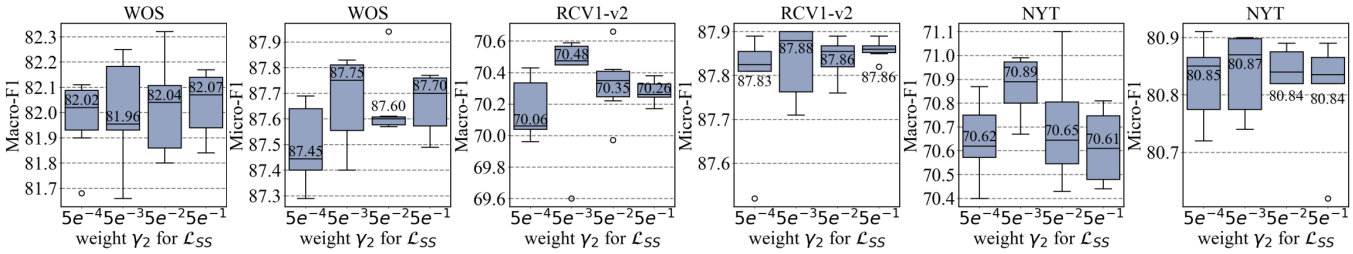
B. Ablation Study

We conduct ablation experiments on the three datasets to further demonstrate the effectiveness of the three loss functions included in SIHTC. The Micro-F1 and Macro-F1 scores of different variants are reported in Table IV and Table V, respectively. Compared to its variants, the complete SIHTC achieves the best results across all three datasets.

First, without **the structural entropy loss** $\mathcal{L}_{SE}$, Micro-F1 scores drop by 0.28%, 0.08%, and 0.09% on the WOS, RCV1-v2, and NYT datasets, respectively, with an average decrease of 0.15%. Macro-F1 scores decrease by 0.16%, 0.61%, and 0.06% on the three datasets, averaging a 0.28% drop. $\mathcal{L}_{SE}$ models inter-text relationships by leveraging label hierarchy information. The performance degradation observed after removing $\mathcal{L}_{SE}$ underscores its effectiveness in capturing hierarchical structural dependencies, which are essential for learning semantically meaningful and discriminative text embeddings. Second, without **the global singular spectrum smoothing regularization loss** $\mathcal{L}_{gss}$, Micro-F1 scores drop by 0.06%, 0.32% on the WOS and RCV1-v2 datasets, respectively. Macro-F1 scores decrease by 0.27%, 0.44%, and 0.32% on the three datasets, respectively, with an average decline of 0.35%. $\mathcal{L}_{gss}$ serves to smooth the singular value spectrum of the label representation matrix. This dual constraint of $\mathcal{L}_{gss}$ suppresses embedding collapse and promotes represen-



(a) The impact of the weight hyperparameter γ_1 for $\mathcal{L}_{SE}$ on the performance of SIHTC.



(b) The impact of the weight hyperparameter γ_2 for $\mathcal{L}_{SS}$ on the performance of SIHTC.

Figure 10: Experimental results of hyperparameter sensitivity.

tational diversity across label embeddings. The performance decline after removing $\mathcal{L}_{gss}$ indicates that, without spectral smoothing, label representations often degenerate–particularly impairing the discrimination of tail labels. Third, without **the local singular spectrum smoothing regularization loss** $\mathcal{L}_{lss}$, Micro-F1 scores drop by 0.12%, 0.02%, and 0.19% on the three datasets, respectively, with an average decrease of 0.11%. Macro-F1 scores drop by 0.41%, 0.51%, and 0.45% on the three datasets, respectively, with an average decrease of 0.46%. $\mathcal{L}_{lss}$ differentiates labels at various depths, providing personalized regularization. The observed decline in classification performance after removing $\mathcal{L}_{lss}$ suggests that this personalized approach effectively enhances the impact of regularization.

Additionally, after removing any component of SIHTC, the drop in Micro-F1 is more significant than in Macro-F1, confirming that our method contributes more to handling tail labels. This discrepancy arises because Micro-F1 is dominated by head labels, making it less sensitive to performance on tail labels. In contrast, Macro-F1 treats all labels equally, providing a clearer reflection of the model's performance on tail labels.

C. Long-Tail Hierarchy Study

We sort all the labels in the training set in descending order based on the number of samples they contain and consider the labels in the bottom third as tail labels. Due to the insufficient training samples, the model's classification performance on tail labels is often significantly lower than on head labels. To further illustrate the advantages of our method, we explore SIHTC's classification performance on tail labels. Figure 7 and Figure 8 illustrate the HPT and SIHTC experimental results of precision on the WOS and RCV1-v2 datasets, respectively. Due to unclear features, the precision of HPT and SIHTC is 0 on some labels. Figure 9 illustrates the experimental result of F1 score on the NYT dataset. Compared with HPT, SIHTC achieves an average precision improvement of 1.39% on the WOS dataset and 2.6% on the RCV1-v2 dataset; meanwhile, it achieves an F-score improvement of 1.04% on the NYTimes dataset. These results demonstrate the superiority of $\mathcal{L}_{SE}$ and $\mathcal{L}_{SS}$ in handling few-shot tail labels. Compared to head labels, tail labels have limited supervision, making the label structure information crucial for generating more discriminative text embeddings. Additionally, the non-degenerated label embeddings learned by $\mathcal{L}_{SS}$ reduce the difficulty of model classification. In conclusion, these two losses promote robust tail label classification from the perspective of text embedding and label embedding, which complement each other.

D. Hyperparameter Study

First, we evaluate the impact of the newly introduced weight hyperparameters γ_1 on the performance of SIHTC. We illustrate the experimental results in Figure 10(a), where we label the median results for each type of experiment. The experiments show that larger values of γ_1 lead to significant declines in F1 scores. γ_1 value of 0.05 for the WOS dataset, 0.1 for the RCV1 dataset, and 0.01 for the NYT dataset result in relatively higher and more stable classification performance. Second, we evaluate the impact of the newly introduced weight hyperparameters γ_2 on the performance of SIHTC and illustrate the experimental results in Figure 10(b). Compared to γ_1, γ_2 is less sensitive. The experiments demonstrate that a smaller weight for γ_2 is sufficient to activate the effect of $\mathcal{L}_{SS}$. A γ_2 value of $5e^{-3}$ for the RCV1 and NYT datasets and $5e^{-2}$ for the WOS dataset achieves relatively higher classification performance.

E. Case Study

We conduct case studies on three datasets to explore the practical optimization effects of SIHTC. [18] categorizes classification errors into three types: missed, excessive, and misjudged. In Table VI, we provide example texts in which

Table VI: Case study on three datasets. We provide each text's index (WOS and RCV1-v2) or name (NYTimes) in the dataset. The incorrect labels predicted by HPT and the corrections made by SIHTC are **bolded**. Notably, the labels predicted by SIHTC are identical to the ground truth labels.

	Type	Text	HPT Predicted Labels	SIHTC Predicted Labels	Ground Truth Labels
WOS	recall missed	(7806) Background: Recurrent respiratory tract infections (RRTIs) have a negative impact on both children 's...	Medical	Medical/**Children's Health**	Medical/Children's Health
	remove excessive	(35573) The immune response is determined by the speed of the T cell reaction to antigens assured by a state...	biochemistry/Immunology **biochemistry/Cell biology**	biochemistry/Immunology	biochemistry/Immunology
	correct misjudged	(21670) Low-cost, high-performance vision sensors in conjunction with aerial sensing platforms are providing...	CS/**Image processing**	CS/**Computer vision**	CS/Computer vision
RCV1-v2	recall missed	(26276) Bulgaria's Finance Ministry said on Tuesday it will meet domestic debt maturity payments due on...	ECAT/E21/E212	ECAT/E21/E212 **MCAT/M13/M131**	ECAT/E21/E212 MCAT/M13/M131
	remove excessive	(23864) Expertise ranging from snack packaging design to nuclear waste disposal and the destruction of chemical...	CCAT/C18/C183 **ECAT GCAT**	CCAT/C18/C183	CCAT/C18/C183
	correct misjudged	(24528) Templeton India Asset Management Company and ITC Threadneedle Asset Management Company...	CCAT/**C15**	CCAT/**C17/C171**	CCAT/C17/C171
NYTimes	recall missed	(0787212) Mr. Dole has not rounded up enough votes to pass his welfare bill. Conservatives like Senator Phil...	Opinion/Opinion/Editorials	Opinion/Opinion/Editorials **Features/Travel/Guides/Destinations/ North America/United States**	Opinion/Opinion/Editorials Features/Travel/Guides/Destinations/ North America/United States
	remove excessive	(0538261) She died after a stroke, said her literary agent, Barbara Kouts.Yoshiko Uchida, a writer of children's...	**Features** News/Obituaries	News/Obituaries	News/Obituaries
	correct misjudged	(1842875) G. THOMAS SIMS and MARK LANDLER-KLAUS KLEINFELD'S FUTUREas the chief...	News/ **New York and Region**	News/**Business**	News/Business

Table VII: Number and proportion of SIHTC correcting three types of classification errors compared to HPT.

Dataset	recall missed	remove excessive	correct misjudged	Total Number
WOS	105 (25.30%)	121 (29.16%)	189 (45.54%)	415
RCV1-v2	14895 (36.29%)	21988 (53.57%)	4163 (10.14%)	41046
NYTimes	108 (25.84%)	274 (65.55%)	36 (8.61%)	418
Average	5036 (36.07%)	7461 (53.45%)	1463 (10.48%)	13960

SIHTC corrects the three types of errors made by HPT. Since SIHTC enhances the representation capability of text embeddings and the generalization capability of label embeddings, it can effectively recall missed labels, remove excessive labels, and correct misjudged labels from HPT's errors. Even when the errors involve an entire label path, SIHTC can fully correct them. Furthermore, we analyze the number and proportion of SIHTC correcting three types of errors, as shown in Table VII. Across the three datasets, the average proportions of recalling missed, removing excessive, and correcting misjudged are 36.07%, 53.45%, and 10.48%, respectively, demonstrating that SIHTC's primary advantage over HPT is its ability to avoid predicting unnecessary labels.

F. Computational Cost of SIHTC

We compare the computational cost of SIHTC with the base model HPT on the three datasets with batch size of 32, as shown in Table VIII. SIHTC does not introduce additional training parameters and maintains a similar evaluation time compared to HPT. Adding two optimization modules for text and label embeddings, the training time per epoch increases by 0.31, 0.31, and 0.36 minutes compared to HPT in the three datasets, while the evaluation time decreases by 0.07, 0, and 0.11 minutes, respectively. Additionally, the two optimizations of SIHTC result in a 514MB, 516MB, and 518MB increase in training memory in the three datasets. Since SIHTC consistently outperforms HPT across all three datasets, the slight increase in training time and memory consumption is an acceptable trade-off.

VI. Related Work

A. Hierarchical Text Classification

The labels in hierarchical text classification possess a hierarchical structure, which can be represented as a tree [40] [41] or a directed acyclic graph (DAG). HTC is widely applied in fields such as research proposal classification [42] and software requirements classification [43]. We categorize HTC research into single encoder, dual encoder, and prompt tuning-based approaches.

Early studies only utilize text encoders, ignoring label semantic and structural information. Various approaches have been explored, including reinforcement learning [3], Graph-CNN [44] [45] and meta-learning [46]. Compared to the single encoder method, the dual encoder method adds a label encoder. Zhou *et al.* [20] propose a dual encoder model structure that extracts text features and label features, using hybrid representations for classification. [37] introduces an information maximization module and a label prior matching module. With the success of the fine-tuning paradigm, many studies employ powerful pre-trained language models (PLMs) as text encoders in HTC. Chen *et al.* [21] pioneer the approach of modeling HTC as a semantic matching problem. Wang *et al.* [26] propose the use of contrastive learning to inject label representations into the text encoder. After sufficient training, this approach yields a hierarchy-aware text representation.

However, due to the significant discrepancy between the objectives of downstream tasks and pre-training tasks, fine-tuning does not fully leverage the potential of PLMs. To address this issue, Wang *et al.* [15] construct a dynamic virtual template and label words, using soft prompts to integrate hierarchical label knowledge. Ji *et al.* [16] use hard prompts to construct templates by level, integrating hierarchical label knowledge into verbalizers and defining a few-shot setting for HTC. Cai *et al.* [17] incorporate contrastive learning into prompt tuning.

Table VIII: Computational cost statistics of HPT and SIHTC on three datasets with batch 32.

Computational Cost	WOS			RCV1-v2			NYTimes			Average		
	HPT	SIHTC	Δ	HPT	SIHTC	Δ	HPT	SIHTC	Δ	HPT	SIHTC	Δ
Params (M)	114.94	114.94	↑ 0	114.92	114.92	↑ 0	114.97	114.97	↑ 0	114.94	114.94	↑ 0
Training Time (min/epoch)	13.56	13.87	↑ 0.31	9.35	9.66	↑ 0.31	10.63	10.99	↑ 0.36	11.18	11.51	↑ 0.33
Evaluation Time (min/epoch)	1.96	1.89	↓ 0.07	0.64	0.64	↓ 0	3.04	2.93	↓ 0.11	1.88	1.82	↓ 0.6
Training Memory (M)	35322	35836	↑ 514	35352	35868	↑ 516	35470	35988	↑ 518	35381	35897	↑ 516

B. Structural Entropy

Structural entropy, first proposed by Li and Pan [27], is a metric to measure the uncertainty of graphs. Structural entropy is a natural extension of Shannon entropy, which extends from unstructured probability distributions to graphs with arbitrary structures. Structural information theory uses an encoding tree to encode graphs. Minimizing structural entropy can reveal the intrinsic structure of a graph. It is initially used to analyze biological information structures [47]. In recent years, structural entropy has been widely applied in various research, including graph structural learning [48], graph contrastive learning [49], reinforcement learning [50], social event detection [51], social bot detection [52], graph classification [53]. [54] proposes a structural entropy-based loss in graph neural network. However, it is limited to the original two-dimensional encoding tree. Our work is the first to extend the structural entropy loss to high-dimensional trees with an aggregation approach.

VII. Conclusion

This paper proposes a novel HTC optimization framework SIHTC based on structural entropy and singular spectrum Smoothing, handling long-tailed data and complex label hierarchies effectively. Firstly, we minimize the structural entropy of texts within the label hierarchy via a tree aggregation network and a structural entropy loss function to inject label structural information into text embeddings, improving their distinctiveness. Secondly, we smooth the singular spectrum of the label embedding matrix to mitigate label representation degeneration and enhance generalization capability, particularly for tail labels. Experiments on three datasets show that SIHTC outperforms all baselines and improves tail label classification performance.

Acknowledgments

This work has been supported by NSFC through grants 62322202, 62441612, 62476163, U21B2027, and U23A2038, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 246Z0102G, the "Pionee" and "Leading Goose" R&D Program of Zhejiang through grant 2025C02044, the Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020, Yunnan Provincial Major Science and Technology Special Plan Projects through grants 202402AG050007 and 202303AP140008, Yunnan Fundamental Research Projects through grant 202301AS070047, Hebei Natural Science Foundation through grant F2024210008, and National Key Laboratory under grant 241-HF-D07-01. The authors would like to thank Dr. Zhiwei Fan for early guidance.

References

[1] A. Sun and E.-P. Lim, "Hierarchical text classification and evaluation," in *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, 2001, pp. 521–528.

[2] A. Sun, E.-P. Lim, W.-K. Ng, and J. Srivastava, "Blocking reduction strategies in hierarchical text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1305–1308, 2004.

[3] Y. Mao, J. Tian, J. Han, and X. Ren, "Hierarchical text classification with reinforced label assignment," in *Proceedings of the EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 2019, pp. 445–455.

[4] X. Zhao, Y. An, N. Xu, and X. Geng, "Variational continuous label distribution learning for multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, pp. 2716–2729, 2023.

[5] D. Zong and S. Sun, "Bgnn-xml: Bilateral graph neural networks for extreme multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6698–6709, 2022.

[6] H. Ren, Y. Cai, R. Y. Lau, H.-f. Leung, and Q. Li, "Granularity-aware area prototypical network with bimargin loss for few shot relation classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4852–4866, 2022.

[7] J. Li, K. Lu, Z. Huang, and H. T. Shen, "On both cold-start and long-tail recommendation with social data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 194–208, 2019.

[8] J. Ren, H. Peng, L. Jiang, Z. Liu, J. Wu, Z. Yu, and P. S. Yu, "Uncertainty-guided boundary learning for imbalanced social event detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 6, pp. 2701–2715, 2024.

[9] Y. Wang, D. Wang, H. Liu, B. Hu, Y. Yan, Q. Zhang, and Z. Zhang, "Optimizing long-tailed link prediction in graph neural networks through structure representation enhancement," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3222–3232.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] J. Li, X. Sun, Y. Li, Z. Li, H. Cheng, and J. X. Yu, "Graph intelligence with large language models and prompt learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6545–6554.

[12] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen, "Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction," in *Proceedings of the ACM Web conference 2022*, 2022, pp. 2778–2788.

[13] W. Zhang, Y. Zhu, M. Chen, Y. Geng, Y. Huang, Y. Xu, W. Song, and H. Chen, "Structure pretraining and prompt tuning for knowledge graph transfer," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2581–2590.

[14] Y. Zhu, Y. Wang, J. Qiang, and X. Wu, "Prompt-learning for short text classification," *IEEE Transactions on Knowledge and Data Engineering*, pp. 5328–5339, 2023.

[15] Z. Wang, P. Wang, T. Liu, B. Lin, Y. Cao, Z. Sui, and H. Wang, "Hpt: Hierarchy-aware prompt tuning for hierarchical text classification," in *Proceedings of the EMNLP*, 2022, pp. 3740–3751.

[16] K. Ji, Y. Lian, J. Gao, and B. Wang, "Hierarchical verbalizer for few-shot hierarchical text classification," in *Proceedings of the ACL*, 2023, pp. 2918–2933.

[17] F. Cai, Z. Zhang, D. Liu, and X. Fang, "Cophtc: Contrastive learning with prompt tuning for hierarchical text classification," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 5400–5404.

[18] S. Xiong, Y. Zhao, J. Zhang, L. Mengxiang, Z. He, X. Li, and S. Song, "Dual prompt tuning based contrastive learning for hierarchical text classification," in *Findings of the ACL*, 2024, pp. 12 146–12 158.

[19] S. Chen, E. Yu, J. Li, and W. Tao, "Delving into the trajectory long-tail distribution for muti-object tracking," in *Proceedings of the IEEE/CVF*, 2024, pp. 19 341–19 351.
[20] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, and G. Liu, "Hierarchy-aware global model for hierarchical text classification," in *Proceedings of the ACL*, 2020, pp. 1106–1117.
[21] H. Chen, Q. Ma, Z. Lin, and J. Yan, "Hierarchy-aware label semantics matching network for hierarchical text classification," in *Proceedings of the ACL/IJCNLP*, 2021, pp. 4370–4379.
[22] J. Gao, D. He, X. Tan, T. Qin, L. Wang, and T. Liu, "Representation degeneration problem in training natural language generation models," in *International Conference on Learning Representations*, pp. 5823–5836.
[23] S. Yu, J. Song, H. Kim, S. Lee, W.-J. Ryu, and S. Yoon, "Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings," in *Proceedings of the ACL*, 2022, pp. 29–45.
[24] Z. Fan, Z. Liu, H. Peng, and P. S. Yu, "Addressing the rank degeneration in sequential recommendation via singular spectrum smoothing," *arXiv preprint arXiv:2306.11986*, 2023.
[25] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 813–823.
[26] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification," in *Proceedings of the ACL*, 2022, pp. 7109–7119.
[27] A. Li and Y. Pan, "Structural information and dynamical complexity of networks," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3290–3339, 2016.
[28] H. Zhu, C. Zhang, J. Huang, J. Wu, and K. Xu, "Hitin: Hierarchy-aware tree isomorphism network for hierarchical text classification," in *Proceedings of the ACL*, 2023, pp. 7809–7821.
[29] H. Zhu, J. Wu, R. Liu, Y. Hou, Z. Yuan, S. Li, Y. Pan, and K. Xu, "Hill: Hierarchy-aware information lossless contrastive learning for hierarchical text classification," in *Proceedings of the NAACL-HLT*, 2024, pp. 4731–4745.
[30] C. Chen, Y. Zhai, Z. Gao, K. Xu, S. Yang, Y. Li, B. Ding, D. Feng, and H. Wang, "Nuclear norm maximization-based curiosity-driven reinforcement learning," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 05, pp. 2410–2421, 2024.
[31] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian, "Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations," in *Proceedings of the IEEE/CVF*, 2020, pp. 3941–3950.
[32] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
[33] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2017, pp. 364–371.
[34] D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research*, vol. 5, no. Apr, pp. 361–397, 2004.
[35] E. Sandhaus, "The new york times annotated corpus," *Linguistic Data Consortium, Philadelphia*, vol. 6, no. 12, p. e26752, 2008.
[36] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the AAAI*, 2015, pp. 2267–2273.
[37] Z. Deng, H. Peng, D. He, J. Li, and P. S. Yu, "Htcinfomax: A global model for hierarchical text classification via information maximization," in *Proceedings of the NAACL-HLT*, 2021, pp. 3259–3265.
[38] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, vol. 1. Minneapolis, Minnesota, 2019, p. 2.
[39] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, no. 1, pp. 69–90, 1999.
[40] S. Dumais and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, pp. 256–263.
[41] K. Wang, S. Zhou, and Y. He, "Hierarchical classification of real life documents," in *Proceedings of the 2001 SIAM International Conference on Data Mining*. SIAM, 2001, pp. 1–16.
[42] M. Xiao, Z. Qiao, Y. Fu, H. Dong, Y. Du, P. Wang, H. Xiong, and Y. Zhou, "Hierarchical interdisciplinary topic detection model for research proposal classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9685–9699, 2023.
[43] M. Binkhonain and L. Zhao, "A machine learning approach for hierarchical classification of software requirements," *Machine Learning with Applications*, vol. 12, p. 100457, 2023.
[44] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1063–1072.
[45] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. S. Yu, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2505–2519, 2019.
[46] J. Wu, W. Xiong, and W. Y. Wang, "Learning to learn and predict: A meta-learning approach for multi-label classification," in *Proceedings of the EMNLP-IJCNLP*, 2019, pp. 4354–4364.
[47] A. Li, X. Yin, B. Xu, D. Wang, J. Han, Y. Wei, Y. Deng, Y. Xiong, and Z. Zhang, "Decoding topologically associating domains with ultra-low resolution hi-c data by graph structural entropy," *Nature communications*, vol. 9, no. 1, p. 3265, 2018.
[48] D. Zou, H. Peng, X. Huang, R. Yang, J. Li, J. Wu, C. Liu, and P. S. Yu, "Se-gsl: A general and effective graph structure learning framework through structural entropy optimization," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 499–510.
[49] J. Wu, X. Chen, B. Shi, S. Li, and K. Xu, "Sega: Structural entropy guided anchor view for graph contrastive learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 37 293–37 312.
[50] X. Zeng, H. Peng, and A. Li, "Adversarial socialbots modeling based on structural information principles," in *Proceedings of the AAAI*, vol. 38, no. 1, 2024, pp. 392–400.
[51] Y. Cao, H. Peng, Z. Yu, and S. Y. Philip, "Hierarchical and incremental structural entropy minimization for unsupervised social event detection," in *Proceedings of the AAAI*, vol. 38, no. 8, 2024, pp. 8255–8264.
[52] Y. Yang, Q. Wu, B. He, H. Peng, R. Yang, Z. Hao, and Y. Liao, "Sebot: Structural entropy guided multi-view contrastive learning for social bot detection," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3841–3852.
[53] J. Wu, S. Li, J. Li, Y. Pan, and K. Xu, "A simple yet effective method for graph classification," in *Proceedings of the IJCAI*, 2022, pp. 3580–3586.
[54] Y. Wang, Y. Wang, Z. Zhang, S. Yang, K. Zhao, and J. Liu, "User: Unsupervised structural entropy-based robust graph neural network," in *Proceedings of the AAAI*, vol. 37, no. 8, 2023, pp. 10 235–10 243.

Qitong Liu is currently a Master's Degree candidate in the School of Cyber Science and Technology at Beihang University. Her research interests include machine learning and deep learning.

Hao Peng is currently a Professor at the School of Cyber Science and Technology at Beihang University. His research interests include representation learning, social network mining, and reinforcement learning. To date, Dr Peng has published over 150 research papers in top-tier journals and conferences, including the IEEE TKDE, TPAMI, TC, ACM TOIS, SIGIR, SIGKDD, ICML, NeurIPS, and Web Conference.

Xiang Huang is currently a Master's Degree candidate in the School of Cyber Science and Technology at Beihang University. Her research interests include machine learning and deep learning.

Zhifeng Hao received his B.S. degree in Mathematics from the Sun Yat-Sen University in 1990, and his Ph.D. degree in Mathematics from Nanjing University in 1995. He is currently a Professor in the Department of Mathematics, College of Science, Shantou University. His research interests involve various aspects of algebra, machine learning, data mining, and evolutionary algorithms.

Qingyun Sun is currently an Assistant Professor at the School of Computer Science and Engineering, and Beijing Advanced Innovation Center for Big Data and Brain Computing at Beihang University. Her research interests include machine learning and graph mining. She has published several papers on IEEE TPAMI, IEEE TKDE, ICML, NeurIPS, Web Conference, AAAI, etc.

Zhengtao Yu received the Ph.D. degree in computer application technology from the Beijing Institute of Technology, Beijing, China, in 2005. He is currently a Professor with the School of Information Engineering and Automation, Kunming University of Science and Technology, China. His current research interests include natural language process, image processing, and machine learning.

Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois at Chicago. Before joining UIC, he was at the IBM Watson Research Center, where he built a world-renowned data mining and database department. He is a Fellow of the ACM and IEEE. Dr. Yu was the Editor-in-Chiefs of ACM Transactions on Knowledge Discovery from Data (2011-2017) and IEEE Transactions on Knowledge and Data Engineering (2001-2004).