HyILR: Hyperbolic Instance-Specific Local Relationships for Hierarchical Text Classification

Ashish Kumar and **Durga Toshniwal**
Indian Institute of Technology Roorkee, Roorkee, India
{ashish_k,durga.toshniwal}@cs.iitr.ac.in

Abstract

Recent approaches to Hierarchical Text Classification (HTC) rely on capturing the global label hierarchy, which contains static and often redundant relationships. Instead, the hierarchical relationships within the instance-specific set of positive labels are more important, as they focus on the relevant parts of the hierarchy. These localized relationships can be modeled as a semantic alignment between the text and its positive labels within the embedding space. However, without explicitly encoding the global hierarchy, achieving this alignment directly in Euclidean space is challenging, as its flat geometry does not naturally support hierarchical relationships. To address this, we propose Hyperbolic Instance-Specific Local Relationships (HyILR), which models instance-specific relationships using the Lorentz model of hyperbolic space. Text and label features are projected into hyperbolic space, where a contrastive loss aligns text with its labels. This loss is guided by a hierarchy-aware negative sampling strategy, ensuring the selection of structurally and semantically relevant negatives. By leveraging hyperbolic geometry for this alignment, our approach inherently captures hierarchical relationships and eliminates the need for global hierarchy encoding. Experimental results on four benchmark datasets validate the superior performance of HyILR over baseline methods.[1]

1 Introduction

Hierarchical Text Classification (HTC) is a sub-task of multi-label classification where text is assigned to one or more labels, organized hierarchically to reflect relationships among them. HTC is particularly useful in domains where labels are naturally structured, such as news categorization (Sandhaus, 2008), product categorization (Shen et al., 2021), and medical diagnosis (Yan et al., 2023). Despite the advancements of large language models, specialized HTC models remain relevant due to challenges posed by complex hierarchical label structures, inherent label imbalance, and the lack of sufficient annotated datasets.(Torba et al., 2024).

A common approach in dual-encoder-based HTC methods is to model the global label hierarchy to learn label representations (Zhou et al., 2020; Chen et al., 2021; Zhu et al., 2023, 2024). While the global hierarchy provides important structural information, the structure is static across all instances (Wang et al., 2022a), which can introduce redundancy and complexity into the classification framework. In contrast, the hierarchical structure associated with instance-specific positive labels represents dynamic and localized relationships, capturing dependencies between relevant labels. Modeling these local relationships can enable more precise and context-aware classification. Although several recent works (Kumar and Toshniwal, 2024; Wang et al., 2024) incorporate instance-specific hierarchical information, they still rely on encoding the full global hierarchy.

In this paper, we address this limitation by directly modeling instance-specific local relationships as a semantic alignment task, without requiring any global hierarchy encoding. By bringing the text closer to its positive labels in the embedding space, the alignment ensures the capture of these relationships. However, without encoding the global hierarchy, achieving alignment in Euclidean space is challenging because its flat, zero-curvature geometry lacks the capacity for representing hierarchical structures. Hyperbolic space, with its negative curvature, supports exponential growth of distances and volumes, making it well suited to naturally represent such structures. The inherent hierarchical nature of hyperbolic space embeds the labels hierarchically, and semantic alignment in this space ensures the capture of relationships by aligning the

[1] Code is available at:https://github.com/havelhakimi/HyILR

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 872–883
July 28-29, 2025 ©2025 Association for Computational Linguistics

labels according to the instance-specific local hierarchy. We use the Lorentz model for hyperbolic space, as it ensures numerical stability and reduces geometric distortions compared to other hyperbolic models (Nickel and Kiela, 2018; Chen et al., 2022).

We introduce Hyperbolic Instance-Specific Local Relationships (HyILR), a method designed to model instance-specific relationships using the Lorentz model of hyperbolic space. During training, both text and label features are projected into hyperbolic space, where a contrastive loss function aligns the text with its associated positive labels. The loss incorporates a hierarchy-aware negative sampling strategy, that uses structural information from the global hierarchy. For each positive label, the closest negative labels are selected from both its descendants and siblings within the hierarchy, as these represent different aspects of the same category. This ensures the sampled negatives are both structurally and semantically relevant, enabling the contrastive loss to effectively capture instance-specific relationships based on the local hierarchy. Our approach improves the representation of all features. Predictions are then made using the text-label-aware composite features in Euclidean space. The contributions of our work are:

- We propose modeling instance-specific local relationships in hyperbolic space, leveraging its geometric properties to capture hierarchical relationships. Unlike prior dual-encoder HTC methods, our approach does not require explicit encoding of the global label hierarchy, thereby simplifying the overall architecture.
- We introduce HyILR, which models instance-specific local relationships as a semantic alignment task, achieved through contrastive learning with hierarchy-aware negative sampling in the Lorentz model of hyperbolic space. To the best of our knowledge, no existing work in HTC has utilized Lorentzian geometry for this purpose.
- Experimental results across four distinct datasets demonstrate the superiority of HyILR in improving classification performance.

2 Related Work

HTC approaches are divided into local and global methods. Local methods train separate classifiers for different sections of the hierarchy but rely on localized context, often leading to inconsistencies (Kowsari et al., 2017; Wehrmann et al., 2018; Shimura et al., 2018). In contrast, global methods use a single classifier that incorporates the entire label hierarchy, making them more efficient and the focus of recent research. Several methods that constrain the classifier using hierarchical path information, such as reinforcement learning (Mao et al., 2019), meta-learning (Wu et al., 2019), and capsule networks (Aly et al., 2019), have been explored for global HTC. Zhou et al. (2020) proposed a graph encoder to explicitly model the entire label hierarchy and introduced two variants for text and label feature interaction. Building on this, several methods based on dual-encoder frameworks have been proposed. Deng et al. (2021) integrates an information maximization module to link text samples with target labels while reducing the influence of irrelevant labels. Chen et al. (2021) projects text and labels into a shared embedding space, using a semantic matching function to relate text to its corresponding labels. Wang et al. (2022a) employs contrastive learning to embed label information into the text encoder. Wang et al. (2022b) injects hierarchical label knowledge into soft prompts and reformulates HTC as a masked language modeling task. Zhu et al. (2023) builds a coding tree by minimizing structural entropy and uses a lightweight graph encoder for hierarchy-aware feature extraction. Kumar and Toshinwal (2024) introduces a custom multi-label loss to model label correlations in a hierarchy-aware manner. Zhu et al. (2024) introduces an information-lossless framework for generating contrastive samples while preserving semantic and syntactic information from the input. Distinct from dual-encoder approaches, some methods adopt a generative framework (Prajapat and Toshniwal, 2024; Iso et al., 2024), formulating HTC as a label sequence generation task based on level and path dependencies (Huang et al., 2022; Yu et al., 2022).

The application of hyperbolic methods for HTC remains underexplored. Existing approaches (Chen et al., 2020; Chatterjee et al., 2021) that use hyperbolic space rely on the Poincaré ball model for projection, which distorts distances near the boundary and can introduce numerical instabilities (Nickel and Kiela, 2018; Desai et al., 2023). In contrast, our method utilizes the Lorentz model and incorporates dynamic instance-specific label information.

3 Preliminaries

A *Riemannian manifold* (M, g) is a smooth manifold M equipped with a Riemannian metric g, which assigns an inner product g_p to the tangent space T_pM at each point $p \in M$ in a differentiable manner. The tangent space T_pM, consisting of all tangent vectors at p, is a vector space that provides a linear approximation of M near p; the metric g_p equips T_pM with an inner product structure, making it locally resemble a Euclidean space.

Hyperbolic space, a type of Riemannian manifold with constant negative curvature, differs fundamentally from Euclidean space, which has zero curvature. Due to their incompatible curvatures an n-dimensional hyperbolic space cannot be perfectly represented in Euclidean space $\mathbb{R}^n$ without distorting angles, distances, or both (e.g., Poincaré model, Klein model). In our study, we use the Lorentz model, which represents hyperbolic space as a submanifold in $\mathbb{R}^{n+1}$.

3.1 Lorentz Model

We represent the n-dimensional hyperbolic space $\mathcal{H}^n$ using the Lorentz model, which embeds the hyperbolic space as a sub-manifold within the higher-dimensional ambient space $\mathbb{R}^{n+1}$. Geometrically, this corresponds to the upper sheet of a two-sheeted hyperboloid as shown in Figure 1. Formally, any vector $\mathbf{u} \in \mathbb{R}^{n+1}$ has the form $\mathbf{u} = [\mathbf{u}_s, u_t]$, where $\mathbf{u}_s \in \mathbb{R}^n$ represents the *space*-like component, and $u_t \in \mathbb{R}$ is the *time*-like component. This terminology of *space* and *time*-like components originates from special relativity theory, where the hyperboloid's axis of symmetry is associated with the time-like component, while all other axes are referred to as space components (Nickel and Kiela, 2017). The Lorentzian inner product $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ for two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n+1}$ is given as:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}} = \langle \mathbf{u}_s, \mathbf{v}_s \rangle - u_t v_t \tag{1}$$

where $\langle \mathbf{u}_s, \mathbf{v}_s \rangle$ is the standard Euclidean dot product and the Lorentzian norm is given as: $\|\mathbf{u}\|_{\mathcal{L}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{L}}}$.

The Lorentz model $\mathcal{H}^n$, characterized by curvature $-k$ (where $k > 0$), is defined as the set:

$$\mathcal{H}^n = \{\mathbf{u} \in \mathbb{R}^{n+1} : \langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{L}} = -1/k\} \tag{2}$$

where all vectors in $\mathcal{H}^n$ satisfy the constraint :

$$u_t = \sqrt{1/k + \|\mathbf{u}_s\|^2} \tag{3}$$

Geodesics. In the Lorentz model, geodesics are curves formed by the intersection of the hyperboloid with hyperplanes that pass through the origin of the ambient space $\mathbb{R}^{n+1}$. These curves represent the shortest paths between points in hyperbolic space, analogous to straight lines in Euclidean geometry, but they appear as hyperbolas when viewed in the ambient space. The geodesic distance in the Lorentz space is given by:

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{1/k} \cosh^{-1}(-k\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}}) \tag{4}$$

Tangent Space. The tangent space at a point $\mathbf{p} \in \mathcal{H}^n$ is the set of all vectors orthogonal to $\mathbf{p}$ under the Lorentzian inner product:

$$T_{\mathbf{p}}\mathcal{H}^n = \{\mathbf{q} \in \mathbb{R}^{n+1} : \langle \mathbf{p}, \mathbf{q} \rangle_{\mathcal{L}} = 0\} \tag{5}$$

Given a vector $\mathbf{z} \in \mathbb{R}^{n+1}$, it can be projected onto the tangent space $T_{\mathbf{p}}\mathcal{H}^n$ using the projection formula:

$$\mathbf{q} = \text{proj}_{\mathbf{p}}(\mathbf{z}) = \mathbf{z} + k\, \mathbf{p}\, \langle \mathbf{p}, \mathbf{z} \rangle_{\mathcal{L}} \tag{6}$$

Exponential Map. The exponential map projects a vector $\mathbf{q} \in T_{\mathbf{p}}\mathcal{H}^n$ from the tangent space at point $\mathbf{p} \in \mathcal{H}^n$ back onto the hyperboloid $\mathcal{H}^n$:

$$\mathbf{x} = \exp_{\mathbf{p}}(\mathbf{q}) = \cosh(\sqrt{k}\|\mathbf{q}\|_{\mathcal{L}})\mathbf{p} + \frac{\sinh(\sqrt{k}\|\mathbf{q}\|_{\mathcal{L}})}{\sqrt{k}\|\mathbf{q}\|_{\mathcal{L}}}\mathbf{q} \tag{7}$$

In this study, we consider these maps by fixing $\mathbf{p}$ at the origin of the hyperboloid, $\mathbf{O} = [\mathbf{0}, \sqrt{1/k}]$, where all spatial components are zero and the time component is $\sqrt{1/k}$.

4 Methodology

In this section, we explain the components of HyILR, including text-label-aware feature generation, projection into hyperbolic space, and the loss functions used. Figure 1 illustrates the overall architecture of our model.

4.1 Text-Label-Aware Features

We use BERT for encoding the text, as it has been widely used in previous HTC studies (Wang et al., 2022a,b; Zhu et al., 2023, 2024). For an input document D, the encoded text representation is given as: $X = f_{bert}(D)$, where $X \in \mathbb{R}^{s \times h}$, with s representing the token sequence length and h denoting the feature size. To compute text-label-aware features, we apply a label-text attention mechanism using a learnable parameter matrix $W_L \in \mathbb{R}^{h \times c}$, where c is the number of labels:

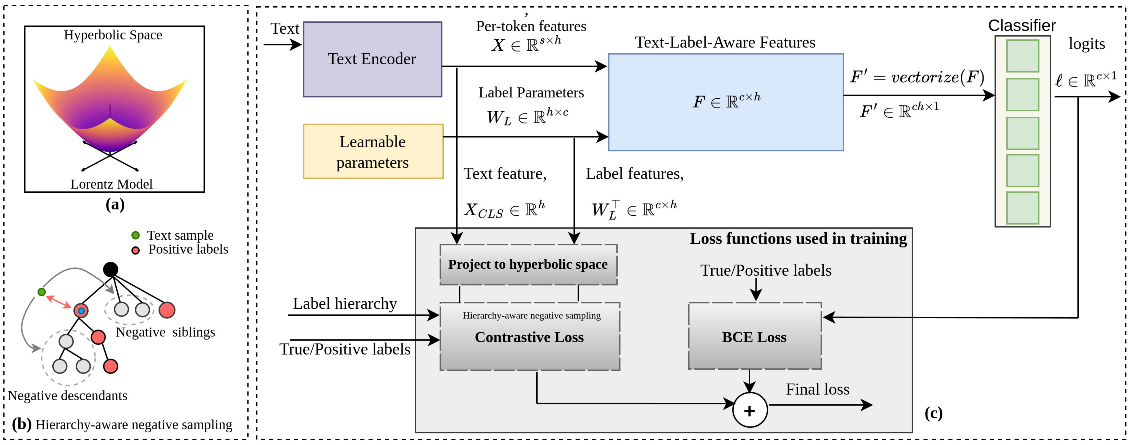

Hyperbolic Space
Lorentz Model
(a)
Text sample
Positive labels
Negative siblings
Negative descendants
(b) Hierarchy-aware negative sampling
Text
Text Encoder
Per-token features
$X \in \mathbb{R}^{s\times h}$
Text-Label-Aware Features
Classifier
logits
Label Parameters
$W_L \in \mathbb{R}^{h\times c}$
$F \in \mathbb{R}^{c\times h}$
$F' = vectorize(F)$
$F' \in \mathbb{R}^{ch\times 1}$
$\ell \in \mathbb{R}^{c\times 1}$
Learnable parameters
Text feature,
$X_{CLS} \in \mathbb{R}^{h}$
Label features,
$W_L^\top \in \mathbb{R}^{c\times h}$
Loss functions used in training
Project to hyperbolic space
True/Positive labels
Label hierarchy
Hierarchy-aware negative sampling
Contrastive Loss
BCE Loss
True/Positive labels
Final loss
+
(c)

Figure 1: (a) Illustration of hyperbolic space $\mathcal{H}^2$ in Euclidean space $\mathbb{R}^3$ (b) For the focused positive label (blue dot), one negative label each is selected from its descendants and siblings based on their distance to the text. This is repeated for all positive labels to form the complete negative label set (c) Architecture of HyILR: The forward pass computes text-label-aware features, which are passed through a classifier to generate predictions. During training, features are projected into hyperbolic space, where contrastive loss captures instance-specific relationships.

$$A = XW_L; \quad F = \mathrm{softmax}(A^\top)X \tag{8}$$

This process helps the model capture the semantic relationships between the text and labels, allowing it to focus on the most relevant tokens for each label. The resulting feature matrix $F \in \mathbb{R}^{c\times h}$ is vectorized to obtain $F' \in \mathbb{R}^{ch\times 1}$ and fed into a classifier. Finally, we obtain the logit vector $\boldsymbol{\ell} \in \mathbb{R}^c$ as:

$$F' = \mathrm{vectorize}(F); \quad \boldsymbol{\ell} = W_c^\top F' + \mathbf{b} \tag{9}$$

where $W_c \in \mathbb{R}^{ch\times c}$ and $\mathbf{b} \in \mathbb{R}^c$ represent the weights and bias of the classifier. The predicted labels are obtained by applying the sigmoid(.) on the logit vector as: $\hat{\boldsymbol{y}} = \mathrm{sigmoid}(\boldsymbol{\ell})$

4.2 Projection onto the Lorentz Hyperboloid

Let $\mathbf{e}_{\mathrm{enc}} \in \mathbb{R}^h$ be the encoded text/label vector. To project it onto the Lorentz hyperboloid $\mathcal{H}^h$ embedded in $\mathbb{R}^{h+1}$, we transform it into $\mathbf{e} = [\mathbf{e}_{\mathrm{s}}, e_{\mathrm{t}}]$, where the space component $\mathbf{e}_{\mathrm{s}} = \mathbf{e}_{\mathrm{enc}}$ and the time-like component $e_{\mathrm{t}} = 0$. Thus, the extended vector $\mathbf{e} \in \mathbb{R}^{h+1}$ is given as $\mathbf{e} = [\mathbf{e}_{\mathrm{enc}}, 0]$. The vector $\mathbf{e}$ is orthogonal to the hyperboloid origin $\mathbf{O} = [\mathbf{0}, \sqrt{1/k}]$ under the Lorentzian inner product, i.e., $\langle \mathbf{e}, \mathbf{O}\rangle_{\mathcal{L}} = 0$, and thus lies in the tangent space at $\mathbf{O}$. Since the time-like component is initially set to zero, the exponential map can be used to parameterize only the *space* component $\mathbf{e}_{\mathrm{s}}$, while the *time*-like component can be recomputed later to satisfy the hyperboloid constraint as given in Eqn 3. Thus, the exponential map can be derived from the generalized formulation in Eqn. 7 as:

$$\exp_{\mathbf{O}}(\mathbf{e}_{\mathbf{s}}) = \cosh(\sqrt{k}\|\mathbf{e}\|_{\mathcal{L}})\mathbf{0} + \frac{\sinh(\sqrt{k}\|\mathbf{e}\|_{\mathcal{L}})}{\sqrt{k}\|\mathbf{e}\|_{\mathcal{L}}}\mathbf{e}_{\mathbf{s}} \tag{10}$$

where the first term is zero. Additionally, the Lorentzian norm $\|\mathbf{e}\|^2_{\mathcal{L}} = \langle \mathbf{e}, \mathbf{e}\rangle_{\mathcal{L}}$ simplifies to the Euclidean norm of the space components, i.e., $\|\mathbf{e}\|^2_{\mathcal{L}} = \langle \mathbf{e}, \mathbf{e}\rangle_{\mathcal{L}} = \langle \mathbf{e}_{\mathrm{s}}, \mathbf{e}_{\mathrm{s}}\rangle - 0 = \|\mathbf{e}_{\mathrm{s}}\|^2$. The final form for exponential map after all substitutions is:

$$\phi(\mathbf{e}_{\mathrm{s}}) = \exp_{\mathbf{O}}(\mathbf{e}_{\mathrm{s}}) = \frac{\sinh(\sqrt{k}\|\mathbf{e}_{\mathrm{s}}\|)}{\sqrt{k}\|\mathbf{e}_{\mathrm{s}}\|}\mathbf{e}_{\mathrm{s}} \tag{11}$$

This approach efficiently embeds Euclidean vectors into hyperbolic space while maintaining the geometric properties of the Lorentz model.

4.3 Loss Functions

4.3.1 Contrastive Loss

We apply contrastive loss in hyperbolic space to align labels based on instance-specific local relationships. To achieve this, we utilize structural information from the global label hierarchy tree H in our negative label selection, ensuring that negative labels are not just arbitrarily close in embedding space but also structurally meaningful. Specifically, we select negative labels from both descendants and siblings of each positive label. Negative descendants, which represent more fine-grained subcategories, prevent the assignment of overly specific labels when the context does not warrant them. Negative siblings, which belong to the same hierarchical level but denote distinct categories, help differentiate between closely related but conceptually distinct labels. The following outlines the overall steps in our contrastive loss formulation.

Exponential Map Transformation. For a batch of m samples, let $T \in \mathbb{R}^{m \times s \times h}$ denote the contextualized token embeddings obtained from the BERT encoder. The embedding of the $[CLS]$ token, $T_{[CLS]} \in \mathbb{R}^{m \times h}$, aggregates the sequence's information and serves as the text feature. Label features are derived from the transpose of learnable parameter matrix as $W_L^\top \in \mathbb{R}^{c \times h}$. The text and label features are then projected into hyperbolic space using the exponential map (Eqn. 11), as:

$$T_{\mathcal{H}} = \phi(\alpha_t T_{[CLS]}); \quad L_{\mathcal{H}} = \phi(\alpha_l W_L^\top) \tag{12}$$

where α_t and α_l are learnable scalars used to scale the text and label features, respectively, ensuring unit norm before projection.

Hierarchy-aware negative sampling. Given a sample i with a positive label set $P(i)$, for each positive label $p \in P(i)$, we select the negative descendant label with the smallest geodesic distance to the text as:

$$N_1 = \{ \underset{j \in Desc(p,H)}{\operatorname{argmin}} \; d(T_{\mathcal{H}_i}, L_{\mathcal{H}_j}) \,|\, p \in P(i) \} \tag{13}$$

where $d(.,.)$ represents the geodesic distance as defined in Eqn. 4, and $T_{\mathcal{H}_i}$ and $L_{\mathcal{H}_j}$ denote the hyperbolic embeddings of the text i and label j, respectively. $Desc(p, H)$ denotes the negative descendant set, which consists of all nodes in the subtree rooted at p within the global hierarchy tree H that are not part of the positive label set. Similarly, we select the negative sibling label with the smallest geodesic distance to the text as:

$$N_2 = \{ \underset{j \in Sib(p,H)}{\operatorname{argmin}} \; d(T_{\mathcal{H}_i}, L_{\mathcal{H}_j}) \,|\, j \notin N_1, p \in P(i) \} \tag{14}$$

where the negative sibling set, denoted as $Sib(p, H)$, consists of all nodes at the same level as p, excluding positive labels. Due to specific hierarchical constraints, a negative label may be selected multiple times—for example, when all but one label at a level are positive, leading all positive labels to choose the same remaining label as their negative sibling. We ensure that only unique negative labels are selected. The overall negative label set for sample i is obtained as: $N(i) = N_1 \cup N_2$. For each positive label, one negative label is selected from each of the sets $Desc(p, H)$ and $Sib(p, H)$, provided they are non-empty; no negative label is chosen when both sets are empty. However, as the contrastive loss utilizes the complete negative set $N(i)$ across all positive labels, the absence of negatives for some labels does not hinder learning.

Loss Formulation. For a sample i, a positive pair $(T_{\mathcal{H}_i}, L_{\mathcal{H}_p})$ consists of its hyperbolic embedding and that of its positive label p. Similarly, a negative pair $(T_{\mathcal{H}_i}, L_{\mathcal{H}_n})$ consists of its hyperbolic embedding and that of a negative label $n \in N(i)$. The contrastive loss is defined as:

$$Loss_{CL} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|P(i)|} \sum_{p \in P(i)} -\log \left(\frac{e^{-d(T_{\mathcal{H}_i}, L_{\mathcal{H}_p})/\tau}}{\sum_{s \in S(i)} e^{-d(T_{\mathcal{H}_i}, L_{\mathcal{H}_s})/\tau}} \right) \tag{15}$$

where $|P(i)|$ denotes the size of $P(i)$, and $S(i) = N(i) \cup P(i)$. τ is the temperature hyperparameter.

4.3.2 Total Loss

The overall loss for HyILR is the sum of Binary Cross Entropy (BCE) and contrastive loss, expressed as: $Loss_{\text{HyILR}} = Loss_{BCE} + \lambda Loss_{CL}$ where $Loss_{BCE}$ is calculated from the logit vector obtained in Eqn 9, and λ controls the weight of the contrastive loss.

5 Experiment

5.1 Experiment Setup

5.1.1 Datasets and Evaluation Metrics

We used four widely recognized benchmark datasets for HTC in our experiments: WOS (Kowsari et al., 2017), RCV1-V2 (Lewis et al., 2004), NYT (Sandhaus, 2008), and BGC [2] (Aly et al., 2019). The statistics for all datasets are presented in Table 1. While each sample in WOS follows a single label path, the other datasets allow for multiple label paths. Similar to previous works (Wang et al., 2022a; Zhu et al., 2023, 2024), we adopt the label taxonomy structure and data preprocessing steps as described in Zhou et al. (2020). For evaluation, we use the Micro-F1 and Macro-F1 scores, consistent with the existing HTC studies (Chen et al., 2021; Wang et al., 2022a; Zhu et al., 2023, 2024).

5.1.2 Implementation Details

We conduct the experiments using an NVIDIA Tesla V100 GPU with 16 GB of memory on a system equipped with an Intel Xeon Gold 6248 processor (40 cores) and 192 GB of RAM. We use the pretrained *bert-base-uncased* [3] as the text en-

[2] https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html

[3] https://huggingface.co/google-bert/bert-base-uncased

Name	Levels	Label Count	Train	Val	Test	Mean-$\|L\|$
WOS	2	141	30070	7518	9397	2.0
RCV1-V2	4	103	20833	2316	781265	3.3
BGC	4	146	58715	14785	18394	3.01
NYT	8	166	23345	5834	7292	7.6

Table 1: Statistical details for the datasets. Levels indicates the number of hierarchy levels, Label count represents the total number of labels, and Mean-$|L|$ denotes the mean number of labels per sample.

coder. Text and label features have dimension h, set to 768. The curvature k is a scalar initialized as 1, and the scalars α_t and α_l are initialized as $1/\sqrt{h}$. We learn all the scalars in the logarithmic space as: $\log(k)$, $\log(\alpha_t)$, and $\log(\alpha_l)$. The weight λ of the contrastive loss is set to 0.3 for WOS, 0.4 for RCV1-V2 and BGC, and 0.6 for NYT, determined via grid search with $\lambda \in \{0.1, 0.2, \ldots, 1.0\}$. τ is fixed at 0.07 for all datasets. During training, the batch size is set to 10, and the Adam optimizer is used with the learning rate fixed at 1e-5. We train the model end-to-end using PyTorch. Training stops if neither Macro-F1 nor the Micro-F1 score improves on the validation set over six consecutive epochs.

5.1.3 Baselines

We compare HyILR against recent dual-encoder HTC methods that model the global label hierarchy. HiAGM (Zhou et al., 2020) constructs a graph encoder to model the global hierarchy and proposes a bi-encoder framework for classification. HTCInfoMax (Deng et al., 2021) introduces an information maximization module between the text and its positive labels to enhance HiAGM. HiMatch (Chen et al., 2021) proposes a semantics matching network by projecting text and labels in a joint embedding space. HGCLR (Wang et al., 2022a) incorporates hierarchical information into the text encoder by performing contrastive learning between the text and positive samples constructed under hierarchy guidance. HPT (Wang et al., 2022b) uses prompt tuning to align the downstream task with the pre-training objective by adding hierarchy-aware soft prompts. HiTIN (Zhu et al., 2023) constructs a coding tree using structural entropy and integrates its hierarchical information into text features with a graph encoder. HILL (Zhu et al., 2024) employs an information lossless strategy, generating positive samples for contrastive learning directly through the graph encoder. In contrast to the encoder-based approaches, Seq2Tree (Yu et al., 2022) and PAAM-HiA-T5 (Huang et al., 2022) are generative models that utilize the T5 (Raffel et al., 2020) architecture. Seq2Tree formulates a constrained decoding strategy with a dynamic vocabulary, while PAAM-HiA-T5 employs path-adaptive attention to capture path dependencies. Apart from these generative models, all other baselines use BERT as the text encoder. We did not compare with the two hyperbolic methods (Chen et al., 2020; Chatterjee et al., 2021) based on the Poincaré ball model due to unclear code details in their repositories but evaluated a variant of our model using the Poincaré ball transformation in the ablation study.

5.2 Main Results

The experimental results are presented in Table 2. The first part of the table compares HyILR with results reported in prior studies. Our method outperforms existing approaches on all datasets except WOS, where methods with a generative framework, PAAM-HiA-T5 and Seq2Tree, performed better, and HyILR achieved the second-best results. HyILR learns instance-specific relationships by aligning text with multiple positive labels. However, in WOS, where each sample has only two positive labels, this limited alignment reduces performance gains compared to other datasets.

For comparison and analysis, we implemented two existing contrastive learning-based approaches, HGCLR and HILL, alongside our model, as shown in the second part of the table. HGCLR constructs contrastive samples with hierarchy guidance but relies on a masking-based approach that may introduce noise, whereas HILL improves upon this by deriving positive samples directly from graph encoder representations, avoiding data augmentation. To evaluate statistical significance, we performed paired t-tests comparing HyILR against each baseline. At a confidence level of 0.05, HyILR demonstrates statistically significant improvements in performance measures. Details of the statistical tests and results are provided in the Appendix A.

Among our implemented models, the second-best results are achieved by HGCLR on WOS and by HILL on the remaining datasets. In terms of Macro-F1 score, HyILR outperforms HGCLR by 0.9% on WOS and surpasses HILL by 2%, 3%, and 1.7% on RCV1-V2, BGC, and NYT, respectively. Similarly, for Micro-F1 score, HyILR improves upon HGCLR by 0.4% on WOS and exceeds HILL by 0.6%, 1.4%, and 1.5% on RCV1-V2, BGC, and NYT, respectively. While HGCLR and HILL rely on modeling the static global hierarchy, HyILR

Model	WoS		RCV1-V2		BGC		NYT	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BERT (Wang et al., 2022a)	85.63	79.07	85.65	67.02	-	-	78.24	66.08
HiAGM (Wang et al., 2022a)	86.04	80.19	85.58	67.93	-	-	78.64	66.76
HTCInfoMax (Wang et al., 2022a)	86.30	79.97	85.53	67.09	-	-	78.75	67.31
HiMatch (Chen et al., 2021)	86.70	81.06	86.33	68.66	78.89	63.19	76.79	63.89
Seq2Tree (Yu et al., 2022)	87.20	**82.50**	86.88	70.01	79.72	63.96	-	-
PAAM-HiA-T5 (Huang et al., 2022)	**90.36**	81.64	87.22	70.02	-	-	77.52	65.97
HGCLR (Wang et al., 2022a)	87.11	81.20	86.49	68.31	-	-	78.86	67.96
HPT (Wang et al., 2022b)	87.16	81.93	87.26	69.53	-	-	80.42	70.42
HiTIN (Zhu et al., 2023)	87.19	81.57	86.71	69.95	-	-	79.65	69.31
HiLL (Zhu et al., 2024)	87.28	81.77	87.31	70.12	-	-	80.47	69.96
HyILR (Ours)	87.48	81.96	**87.41**	**71.20**	**81.52**	**67.85**	**81.26**	**70.71**
Our Implementation								
HGCLR	87.09$_{\pm 0.26}$	81.08$_{\pm 0.28}$	86.27$_{\pm 0.27}$	68.09$_{\pm 0.30}$	79.86$_{\pm 0.31}$	64.10$_{\pm 0.34}$	78.53$_{\pm 0.28}$	67.20$_{\pm 0.35}$
HILL	86.51$_{\pm 0.23}$	80.93$_{\pm 0.30}$	86.76$_{\pm 0.27}$	69.15$_{\pm 0.36}$	80.12$_{\pm 0.30}$	64.82$_{\pm 0.37}$	79.74$_{\pm 0.30}$	69.05$_{\pm 0.35}$
HyILR (Ours)	**87.48**$_{\pm 0.19}$	**81.96**$_{\pm 0.22}$	**87.41**$_{\pm 0.23}$	**71.20**$_{\pm 0.30}$	**81.52**$_{\pm 0.24}$	**67.85**$_{\pm 0.28}$	**81.26**$_{\pm 0.23}$	**70.71**$_{\pm 0.28}$

Table 2: Comparison of results. The original studies of HiAGM and HTCInfoMax do not use a BERT encoder; we compare results from (Wang et al., 2022a), which implements their BERT-based version. The results for HiMatch on BGC and NYT are reported by (Yu et al., 2022) and (Huang et al., 2022), respectively. For our implemented models, we report the average scores over 8 runs with random seeds, in addition to the results from their respective source papers. Second-best results are underlined in both parts of table. ± denotes standard deviation.

focuses on local hierarchical relationships, avoiding the complexity and redundancy associated with encoding the entire hierarchy. Moreover, their contrastive loss formulation relies on batch-based implicit negatives, whereas HyILR uses hierarchy-aware negative sampling for more challenging contrasts.

5.3 Hierarchy-consistent evaluation

We perform a hierarchy-consistent evaluation, where the hierarchical structure of labels is based on the predefined global label hierarchy. In this stricter evaluation, a label is considered correct only if all its ancestor labels are also predicted correctly. Table 3 presents the Hierarchy-consistent Micro-F1 (Hi-MiF1) and Macro-F1 (Hi-MaF1) scores for our implemented models on datasets with deeper hierarchies (RCV1-V2, BGC, and NYT). HyILR demonstrates an increase in Hi-MaF1 by 1.6%, 2.6%, and 1.7% on RCV1-V2, BGC, and NYT, respectively, compared to the second-best score. In contrast to graph encoder-based methods that explicitly encode the global hierarchical structure, HyILR only utilizes hierarchical information during negative sampling to enhance contrastive learning in hyperbolic space. This enables it to implicitly capture instance-specific hierarchical label dependencies, resulting in better hierarchy-consistent predictions.

Model	RCV1-V2		BGC		NYT	
	Hi-MiF1	Hi-MaF1	Hi-MiF1	Hi-MaF1	Hi-MiF1	Hi-MaF1
HGCLR	85.94	67.51	79.43	63.60	78.04	66.27
HILL	86.46	68.54	79.92	63.86	78.64	67.34
HyILR (Ours)	**87.13**	**70.18**	**80.76**	**66.50**	**80.55**	**69.06**

Table 3: Comparison of Hierarchy-consistent scores. The second best results have been underlined

5.4 Ablation Study

We conducted five ablation studies (Table 4). First, we removed the contrastive loss (w/o CL) and trained the model only with BCE loss. The significant drop in performance highlights the importance of contrastive learning in modeling instance-specific relationships. Next, we removed the projection of features into hyperbolic space (Eqn. 12) and applied contrastive loss directly in Euclidean space, using Euclidean distance as the similarity measure (CL-Euclidean (Distance)). However, alignment in Euclidean space is less effective, as its geometry does not naturally capture hierarchical relationships, explaining its underperformance compared to HyILR. A similar performance drop was observed when using cosine similarity in Euclidean space.

We also replaced the Lorentz model with the Poincaré ball model for hyperbolic contrastive learning (CL-Poincaré). While the Poincaré variant outperforms the Euclidean-based variant, it still lags behind HyILR. We further ablated the label-text attention module by replacing it with elementwise multiplication between the text feature of the sample $X_{[CLS]} \in \mathbb{R}^h$ and the label features

Model	WoS		RCV1-V2		BGC		NYT	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
w/o CL	86.10	80.18	85.90	67.33	79.10	63.42	78.70	66.95
CL-Euclidean (Distance)	86.32	80.54	86.23	68.20	79.58	63.84	78.97	68.10
CL-Poincaré	87.03	81.05	86.92	69.74	80.10	66.06	79.95	69.42
w/o Label-text att.	86.55	80.62	86.70	68.82	79.72	64.33	79.20	68.74
w/o HNS CL-Lorentz	86.80	80.73	86.55	68.96	79.90	64.57	79.16	68.95
HyILR (Ours)	**87.48**	**81.96**	**87.41**	**71.20**	**81.52**	**67.85**	**81.26**	**70.71**

Table 4: Ablation study results for HyILR

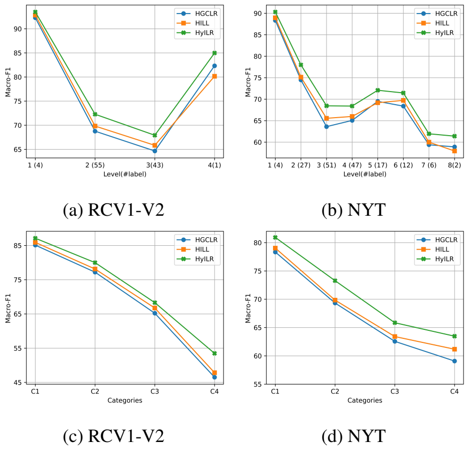
$W_L^\top \in \mathbb{R}^{c \times h}$, yielding $F \in \mathbb{R}^{c \times h}$ (w/o label-text att.). The performance drop highlights the importance of label-text attention, which computes text-label-aware features using weighted attention scores over the token representations. Finally, we validate the effectiveness of our Hierarchy-aware Negative Sampling (HNS) by replacing it with a random negative sampling strategy in the Lorentz model (CL-Lorentz w/o HNS), which results in reduced performance. By focusing on semantically and structurally relevant negative labels, the negative sampling strategy in HyILR enables more effective contrastive learning in hyperbolic space.

We did not ablate the BCE loss, as it optimizes independent label predictions, which is essential in multi-label classification. While the contrastive loss aligns texts with relevant labels, it does not provide supervision for individual label predictions; removing BCE slowed convergence in our experiments due to the absence of this supervision.

5.5 Performance under imbalanced hierarchy

We analyze model performance under hierarchical imbalance, considering two key aspects: (1) the uneven distribution of labels across hierarchy levels and (2) the long-tail effect caused by varying label frequencies. Figure 2 presents the performance on the RCV1-V2 and NYT datasets, which have four and eight hierarchy levels, respectively, with the ratio of samples between the most and least frequent labels exceeding 100 in both. A similar analysis for the WOS and BGC datasets is provided in the Appendix B.

Figure 2 (a-b) illustrates the performance of our implemented models across various hierarchy levels. The mid-levels have a larger number of labels, whereas the deeper levels, which are increasingly fine-grained, contain fewer labels. HyILR shows improvements in performance, especially at mid and deeper levels, where labels become increasingly specific and fine-grained. To analyze the long-tail effect, we sort the labels in descending order by document count and divide them into four equal-sized groups (C1–C4). C1 and C2 represent frequent labels, while C3 and C4 correspond to increasingly sparse labels. Figure 2 (c-d) shows model performance across these categories, with a decline as sparsity increases in categories C3 and C4. However, HyILR consistently outperforms the others, demonstrating its ability to mitigate the long-tail effect. Overall, its instance-specific modeling allows it to focus on each label regardless of granularity or frequency, leading to improved performance across all hierarchy levels and label categories.



(a) RCV1-V2

(b) NYT

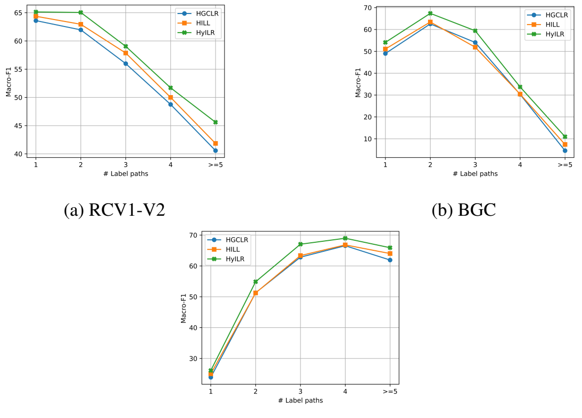
(c) RCV1-V2

(d) NYT

Figure 2: Performance under imbalanced hierarchy : (a-b) Level-wise, (c-d) Label frequency categories

5.6 Model Performance in Relation to Label Path Complexity

In HTC, labels for each sample can belong to one or multiple paths in the label hierarchy, reflecting the multi-label and hierarchical nature of the task. Analyzing model performance across different numbers of label paths provides insights into how well models handle varying levels of label path complexity. Figure 3 illustrates model performance across samples grouped by the number of label paths they belong to, for the RCV1-V2, BGC, and NYT datasets, all of which include multiple label paths. Across all datasets, our proposed model, HyILR, consistently outperforms as label path complexity increases, demonstrating its ability to effectively navigate and classify within complex hierarchical structures.



(a) RCV1-V2

(b) BGC

(c) NYT

Figure 3: Performance comparison across label paths

5.7 Computational Efficiency

We conducted our experiments on an NVIDIA Tesla V100 GPU. The training time for each experiment was approximately 8, 13, 25.5, and 14 hours for the WOS, RCV1-V2, BGC, and NYT datasets, respectively. In Table 5, we compare the computational efficiency of HyILR with two existing baselines on the RCV1-V2 dataset. Although all methods are based on contrastive learning, HyILR demonstrates a lower training computation time and faster inference. Furthermore, the parameter count of HyILR is comparable to that of the existing methods.

Model	#Params	Training time	Inference
	(M)	(min/epoch)	(ms/sample)
HGCLR	119	20.08	10.55
HILL	116	14.33	11.03
HyILR (Ours)	117	10.11	10.29

Table 5: Comparison of parameters and runtime on RCV1-V2 dataset

6 Conclusion

In this paper, we introduced HyILR, a method for modeling instance-specific local relationships in hyperbolic space. By leveraging the Lorentz model, our approach frames the problem as a semantic alignment task in hyperbolic space, aligning text with its positive labels based on their local hierarchical relationships. This alignment is achieved through contrastive loss, which is equipped with a hierarchy-aware negative sampling strategy to incorporate both structural and semantic information while selecting negative labels. Our approach removes the need for global hierarchy encoding, thereby simplifying the classification framework. Comparisons with existing baselines demonstrate that HyILR outperforms state-of-the-art methods and achieves better hierarchical consistency, even without modeling the redundant global structure.

7 Limitations

HyILR is sensitive to the hyperparameter λ, which controls the weight of the contrastive loss, and requires tuning for each dataset. Additionally, HyILR relies on the hierarchy structure to obtain challenging negatives, but in some cases, no negative labels may be available for a given positive label. This can happen, for example, when a leaf label node has no siblings or when a label's only negative sibling has already been selected as a negative descendant for another label. While the model currently utilizes the complete negative set across all positive labels to mitigate this issue, exploring new strategies to obtain negative labels in such cases could further improve contrastive learning.

Acknowledgment

This study was funded by the PMRF (Prime Minister's Research Fellow) program, run by the Ministry of Education, Government of India. We also acknowledge National Supercomputing Mission (NSM) for providing computing resources of "PARAM Ganga" at IIT Roorkee, which is implemented by C-DAC and supported by the Ministry of Electronics and Information Technology (MeitY) and Department of Science and Technology (DST), Government of India.

References

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

Soumya Chatterjee, Ayush Maheshwari, Ganesh Ramakrishnan, and Saketha Nath Jagarlapudi. 2021. Joint learning of hyperbolic label embeddings for hierarchical multi-label classification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main*

Volume, pages 2829–2841, Online. Association for Computational Linguistics.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7496–7503.

Haibin Chen, Qianli Ma, Zhenxi 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 (Volume 1: Long Papers)*, pages 4370–4379, Online. Association for Computational Linguistics.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5672–5686, Dublin, Ireland. Association for Computational Linguistics.

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*, pages 3259–3265, Online. Association for Computational Linguistics.

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. 2023. Hyperbolic image-text representations. In *International Conference on Machine Learning*, pages 7694–7731. PMLR.

Wei Huang, Chen Liu, Bo Xiao, Yihua Zhao, Zhaoming Pan, Zhimin Zhang, Xinyun Yang, and Guiquan Liu. 2022. Exploring label hierarchy in a generative way for hierarchical text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1116–1127, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hayate Iso, Xiaolan Wang, and Yoshi Suhara. 2024. Noisy pairing and partial supervision for stylized opinion summarization. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 13–23, Tokyo, Japan. Association for Computational Linguistics.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.

Ashish Kumar and Durga Toshinwal. 2024. Hlc: hierarchically-aware label correlation for hierarchical text classification. *Applied Intelligence*, 54(2):1602–1618.

Ashish Kumar and Durga Toshniwal. 2024. Modeling text-label alignment for hierarchical text classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 163–179. Springer.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

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)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788. PMLR.

Dharmendra Prajapat and Durga Toshniwal. 2024. Improving multi-domain task-oriented dialogue system with offline reinforcement learning. In *2024 IEEE International Conference on Big Data (BigData)*, pages 2013–2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Evan Sandhaus. 2008. The New York Times Annotated Corpus - Linguistic Data Consortium. *The New York Times*.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249, Online. Association for Computational Linguistics.

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*, pages 811–816, Brussels, Belgium. Association for Computational Linguistics.

Fatos Torba, Christophe Gravier, Charlotte Laclau, Abderrhammen Kammoun, and Julien Subercaze. 2024. A study on hierarchical text classification as a seq2seq task. In *European Conference on Information Retrieval*, pages 287–296. Springer.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022a. 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)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, and Houfeng Wang. 2024. Utilizing local hierarchy with adversarial training for hierarchical text classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17326–17336, Torino, Italia. ELRA and ICCL.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo C. Barros. 2018. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. 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)*, pages 4354–4364, Hong Kong, China. Association for Computational Linguistics.

Jiahuan Yan, Haojun Gao, Zhang Kai, Weize Liu, Danny Chen, Jian Wu, and Jintai Chen. 2023. Text2Tree: Aligning text representation to the label tree hierarchy for imbalanced medical classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7705–7720, Singapore. Association for Computational Linguistics.

Chao Yu, Yi Shen, and Yue Mao. 2022. Constrained sequence-to-tree generation for hierarchical text classification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 1865–1869, New York, NY, USA. Association for Computing Machinery.

Jie 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*, pages 1106–1117, Online. Association for Computational Linguistics.

He Zhu, Junran Wu, Ruomei Liu, Yue Hou, Ze Yuan, Shangzhe Li, Yicheng Pan, and Ke Xu. 2024. HILL: Hierarchy-aware information lossless contrastive learning for hierarchical text classification. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4731–4745, Mexico City, Mexico. Association for Computational Linguistics.

He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. 2023. HiTIN: Hierarchy-aware tree isomorphism network for hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7809–7821, Toronto, Canada. Association for Computational Linguistics.

A Details of statistical test

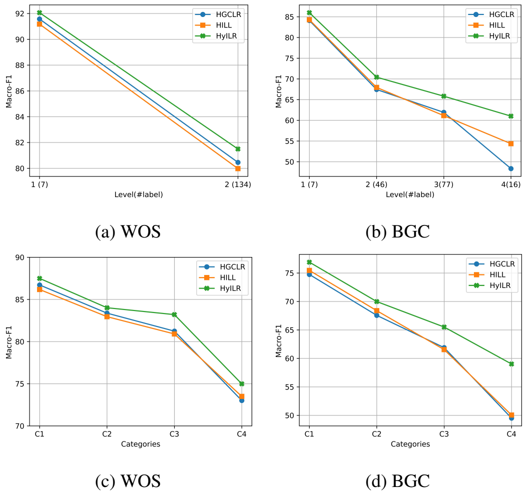
We used Micro-F1 and Macro-F1 scores to evaluate our model's performance. Each experiment was run eight times with random seeds, and the average scores were reported. To determine the statistical significance of the observed improvements, we performed one-sided paired t-tests, comparing our model's performance with that of other implemented models, as shown in Table 6. Except for the Micro-F1 score in the HyILR vs. HGCLR comparison on the WOS dataset, all p-values were below 0.05, confirming the statistical significance of our model's improvements.

B Performance under imbalanced hierarchy for WOS and BGC

We present the results under an imbalanced hierarchy for the WOS and BGC datasets in this section. While WOS has a shallow two-level hierarchy, BGC has a deeper four-level hierarchy. Moreover, both datasets exhibit varying label frequencies, with the ratio of samples between the most and least frequent labels exceeding 1,000. Figure 4 (a-b) illustrates the performance across hierarchy levels, showing a consistent improvement for HyILR at all levels. Similarly, Figure 4 (c-d) presents the results under label frequency categories, where HyILR performs better, particularly for sparse labels in categories C3 and C4.

Dataset	Metrics	Model Pair	p-value (t-test)
WOS	Micro-F1	HyILR vs. HILL	**1.1e-5**
		HyILR vs. HGCLR	**2e-4**
	Macro-F1	HyILR vs. HILL	**2e-4**
		HyILR vs. HGCLR	0.06
RCV1-V2	Micro-F1	HyILR vs. HILL	**5.9e-5**
		HyILR vs. HGCLR	**1.7e-5**
	Macro-F1	HyILR vs. HILL	**2.9e-5**
		HyILR vs. HGCLR	**3.2e-8**
BGC	Micro-F1	HyILR vs. HILL	**1.4e-5**
		HyILR vs. HGCLR	**8.1e-6**
	Macro-F1	HyILR vs. HILL	**9.7e-7**
		HyILR vs. HGCLR	**2.1e-7**
NYT	Micro-F1	HyILR vs. HILL	**4.1e-7**
		HyILR vs. HGCLR	**2.7e-7**
	Macro-F1	HyILR vs. HILL	**2.6e-7**
		HyILR vs. HGCLR	**2.6e-7**

Table 6: One-sided t-test results for model comparisons on different datasets



(a) WOS

(b) BGC

(c) WOS

(d) BGC

Figure 4: Performance under imbalanced hierarchy : (a-b) Level-wise, (c-d) Label frequency categories

C Hyperparameter sensitivity

The performance of our proposed approach is sensitive to the value of λ, which controls the weight of the contrastive loss in the overall loss function of the model. We conducted a grid search on λ values ranging from 0.1 to 1 (in increments of 0.1) to find the optimal value for each dataset. Table 7 shows the results on the validation set for the NYT dataset with different values of λ. Similarly, we obtained the optimal value of λ for the other datasets.

λ	Micro-F1	Macro-F1
0.1	68.94	79.96
0.2	69.23	79.72
0.3	69.33	79.64
0.4	71.40	81.36
0.5	70.16	80.52
0.6	**71.73**	**81.64**
0.7	69.98	79.90
0.8	71.12	80.83
0.9	69.84	80.10
1.0	70.92	80.73

Table 7: Performance of HyILR on the NYT validation set for varying values of λ.