ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240673

Hyperbolic Contrastive Learning for Document Representations – A Multi-View Approach with Paragraph-Level Similarities

Jaeeun Nam[a], Ilias Chalkidis[b] and Mina Rezaei[a,c,*]

[a]Department of Statistics, LMU Munich, Munich, Germany
[b]Department of Computer Science, University of Copenhagen, Denmark
[c]Munich Center for Machine Learning, Munich, Germany

Abstract. Self-supervised learning (SSL) has gained prominence due to the increasing availability of unlabeled data and advances in computational efficiency, leading to revolutionized natural language processing with pre-trained language models like BERT and GPT. Representation learning, a core concept in SSL, aims to reduce data dimensionality while preserving meaningful aspects. Conventional SSL methods typically embed data in Euclidean space. However, recent research has revealed that alternative geometries can hold even richer representations, unlocking more meaningful insights from the data. Motivated by this, we propose two novel methods for integrating Hilbert geometry into self-supervised learning for efficient document embedding. First, we present a method directly incorporating Hilbert geometry into the standard Euclidean contrastive learning framework. Additionally, we propose a multi-view hyperbolic contrastive learning framework contrasting both documents and paragraphs. Our findings demonstrate that contrasting only paragraphs, rather than entire documents, can lead to superior efficiency and effectiveness.

1 Introduction

Recently, large language models (LLMs) have emerged as one of the most notable areas in deep learning, gaining widespread recognition as their applications reached the general public [29]. Similarly, AI-generated images and videos [31, 13, 12] have become well-known concepts. The primary drivers behind these advancements include the availability of vast quantities of unlabeled data and significant progress in self-supervised learning (SSL) [38, 39]. This approach exploits the inherent structures within the data, eliminating the need for costly manual annotation, and has demonstrated performance that is comparable to, or even surpasses, that of supervised methods [16, 21, 10, 4].

A key challenge in SSL lies in devising a representation of data that accurately captures its underlying structure and the relationships among data points. Contrastive learning has emerged as a promising strategy for training models to achieve effective image [9, 11] and linguistic representations [16, 10, 21, 33, 36]. This method focuses on learning embeddings that bring similar instances closer while distancing dissimilar ones within a latent space.

Most current approaches rely on *Euclidean space*, leveraging the intuitive notion of distance. However, the *hyperbolic space* has also gained traction for visual [18, 1] and textual representation [14, 42, 26]. This space holds the potential for capturing hierarchical structures more effectively, leading to richer representation. Recent research has demonstrated the success of combining self-supervised contrastive learning with the hyperbolic space for image data [17, 41].

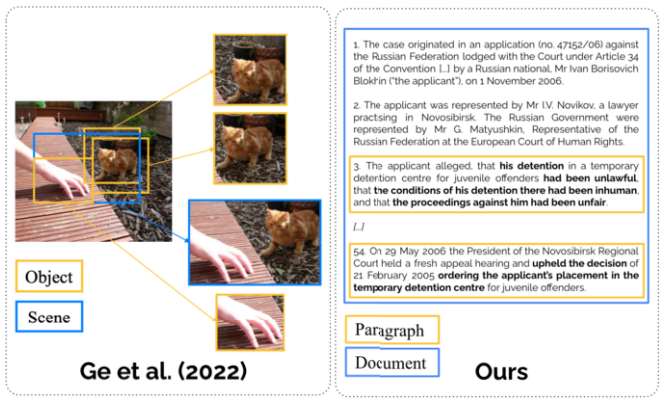

Figure 1: A depiction of the multi-view analogy between visual [17] and text (ours) hierarchies utilized in multi-view contrastive learning for document representation. The scene of an image is interpreted as the document and the objects in the scene as the paragraphs in the document.

Building on the insights from previous studies [32, 35, 17] that highlights the advantages of using hyperbolic space for capturing hierarchical structures in both visual and textual data, Hilbert geometry emerges as a complementary approach, particularly when paired with contrastive learning methods. Much like hyperbolic space, Hilbert geometry excels at modeling complex, hierarchical relationships inherent in the text, facilitating more precise embeddings with reduced dimensional distortion. This characteristic is crucial for text representation, as Hilbert's geometry aligns the geometric relationships within the space more closely with the semantic relationships between text elements—such as paragraphs and sentences. In a contrastive learning framework, this alignment allows for a more effective distinction between similar and dissimilar textual pairs, enhancing the learning process. Consequently, integrating Hilbert geometry not only enriches the text representations but also boosts the efficiency of learning algorithms, resulting in more sophisticated models that better understand and process natural language. This synergy underscores the potential

* Corresponding Author. Email: Mina.Rezaei@stat.uni-muenchen.de.

of expanding beyond traditional Euclidean models to embrace geometries like Hilbert and hyperbolic spaces, which are proving pivotal in advancing the field of text representation learning.

In this paper, we introduce two novel approaches to self-supervised representation learning for efficient text embedding, based on Hilbert geometry – a generalization of the Klein model of hyperbolic space. Our methods, named *One-Branch Geometry Fusion Encoding* and *Two-Branch Geometry Fusion Encoding*, extend the widely used Euclidean contrastive learning framework to incorporate the unique properties of Hilbert space. Additionally, we explore the potential of contrasting paragraphs instead of entire documents for more efficient and effective document representation learning. We evaluate our methods on a number of benchmark NLP tasks and show that they outperform state-of-the-art Euclidean-based methods. Our results suggest that hyperbolic geometry is a promising new approach to self-supervised representation learning for text.

Our research has several implications for the field of NLP. First, it demonstrates the effectiveness of using hyperbolic geometry for self-supervised representation learning of text. Second, it opens up new possibilities for document representation learning by showing that contrasting paragraphs instead of documents can be more efficient and effective. Third, it suggests that hyperbolic geometry may be a useful tool for other NLP tasks, such as machine translation and natural language generation.

2 Background and Related Work

2.1 Hyperbolic Geometry Learning

Hyperbolic space, a type of non-Euclidean geometry, has garnered significant interest due to its properties. It is particularly suited for capturing tree-like hierarchical structures, which has led to its successful application in network science and graph representation. Its effectiveness is also evident in embedding taxonomies [26, 27, 14]. For example, one can consider the hierarchical relationship among the words "animal," "dog," and "pug," or similarly, images representing these categories [18]. In image retrieval, hyperbolic space helps represent hierarchies stemming from whole-to-fragment relationships or, in recognition tasks, from image degradation [20].

Hyperbolic geometry investigates spaces characterized by constant negative curvature. There are five isometric models of hyperbolic spaces identified in the literature: the Lorentz (hyperboloid) model, the Poincaré ball model, the Poincaré half-space model, the Klein model, and the hemisphere model [3, 18, 30, 24]. Among these, the Poincaré ball model is the most frequently utilized [20, 1, 26, 37], primarily because its distance function is differentiable, which facilitates gradient-based optimization techniques [41]. Unlike Euclidean spaces, where lines are straight, lines in hyperbolic space deviate from straightness. Moreover, as one approaches the boundary of hyperbolic space, distances increase significantly, making this geometry particularly effective for representing tree-like structures.

Recent studies have explored the application of contrastive learning within hyperbolic space to enhance image representation [41, 17]. Traditionally, the contrastive learning framework is predicated on Euclidean geometry; however, this framework has been expanded to incorporate learning in hyperbolic space. Notably, [17] diverged from contrasting a single pair of augmented views of an image to contrasting two pairs derived from the same image. For instance, given a *scene* image of a dog playing with a ball, one pair consisted of two cropped views of the *object* region, the dog, contrasted in Euclidean space. The other pair included a *scene* region, showing the dog with the ball, and a contained *object* region, the ball, contrasted in hyperbolic space. This approach leverages the assumed hierarchical relationship between a *scene* and its *objects* to enhance the comprehension of the entire image. Our research extends this exploration by incorporating non-Euclidean geometries into contrastive learning frameworks, specifically examining the implications of Hilbert simplex geometry.

2.2 Hilbert Geometry Learning

Our Hilbert geometry framework is inspired by the foundational research presented in [28], which introduced Hilbert simplex geometry along with its closed-form distance metric. The authors characterize this geometry as a generalization of the Klein model of hyperbolic space and have empirically demonstrated its utility in embedding distance matrices. In this work, we provide a detailed exposition of their mathematical derivations, elucidating how these concepts can be integrated and applied to further our understanding of geometric data representations.

Consider Ω be any open bounded convex set of $\mathbb{R}^d$. The Funk distance $\rho_F^\Omega(p,q)$ between two points $p, q \in \Omega$ is defined by

$$\rho_F^\Omega(p,q) := \begin{cases} log\left(\frac{\|p-\bar{q}\|}{\|q-\bar{q}\|}\right) & , p \neq q \\ 0 & , p = q \end{cases}$$

where $\bar{q}$ denotes the point where an affine ray $R(p,q)$ originating from p and passing through q intersects the boundary $\partial\Omega$, as illustrated in 2. The Hilbert distance $\rho_H^\Omega(p,q)$ is defined as the symmetrization of the Funk distance.

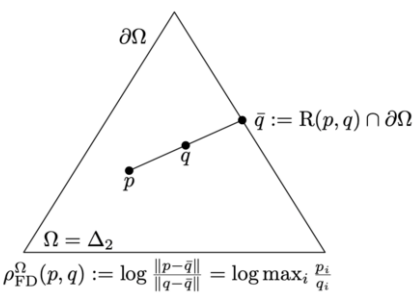

Figure 2: Depiction of the Funk distance defined in the open standard simplex Δ_2 by [28].

$$\rho_H^\Omega(p,q) := \begin{cases} \rho_F^\Omega(p,q) + \rho_F^\Omega(q,p) & p \neq q \\ 0 & p = q \end{cases} \quad (1)$$

Considering the open $(d-1)$-dimensional simplex $\Omega = \Delta_d$, where $\mathbb{R}_{++} := (0, \infty)$.

$$\Delta_d := \left\{(x_1, ..., x_d) \in \mathbb{R}_{++}^d : \sum_{i=1}^{d} x_i = 1\right\} \quad (2)$$

In this case, the Funk distance can be formulated as

$$\rho_F^{\Delta_d}(p,q) = log\, max_{i\in\{1,...,d\}}\}\frac{p_i}{q_i} \quad (3)$$

and because the logarithm function strictly increases, we can further write

$$\rho_F^{\Delta_d}(p,q) = max_{i\in\{1,...,d\}}\}log\frac{p_i}{q_i}. \quad (4)$$

To make the function differentiable, one can approximate the maximum operator by the so-called *log-sum-exp* function LSE. For any $x \in \mathbb{R}^d$ and $\tau > 0$ the approximation formula is denoted by

$$LSE^T := \frac{1}{\tau} log \left(\sum_{i=1}^{d} exp(\tau x_i) \right). \tag{5}$$

Incorporating the approximation we can denote a differentiable pseudo-distance by

$$\tilde{\rho}_H(p,q) = \frac{1}{\tau} log \left(\sum_{i=1}^{d} \left(\frac{p_i}{q_i}\right)^T \right) \left(\sum_{i=1}^{d} \left(\frac{q_i}{p_i}\right)^T \right), \tag{6}$$

which was used in our contrastive learning frameworks to measure the divergence between embeddings. To illustrate the Hilbert simplex geometry and its associated distance function, we visualized balls of constant radius increments centered at a specific point within the simplex. These representations are shown in Figure 3, where each ball depicts the spatial expansion at equidistant steps, providing a clear visualization of the geometry's inherent properties.

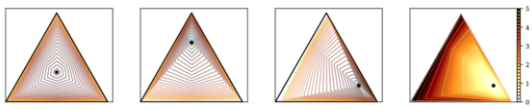

Figure 3: Illustration of balls with constant radius increment step centered at $c \in \Delta_2$ by [28]. The last figure shows the distance color maps with brightness decreasing with increasing distance.

Finally, we converted divergences into similarities by the following strictly monotone function:

$$\psi = \frac{1}{\tilde{\rho}_H(p,q) + 1} \tag{7}$$

2.3 Contrastive Learning

Contrastive learning aims on robust representation learning by comparing pairs of data points. Its primary goal is to align representations of similar ("positive") examples more closely, while distancing those of semantically dissimilar ("negative") examples within a hidden, abstract space known as the latent representation. This technique often utilizes data augmentation to enhance the diversity of input data. As a prominent method in self-supervised learning, it enables models to learn without explicit labels. Figure 4 presents a simplified diagram of a typical contrastive learning model. In this model, a data point—whether an image or text—is subjected to two distinct transformations. These transformed data points are then processed by an encoder and a Multilayer Perceptron (MLP) head. Subsequently, the model evaluates the similarity or difference between these processed representations within a small batch of data points. This evaluative step is integral to the model's training objective, facilitating the learning of meaningful and discriminative features.

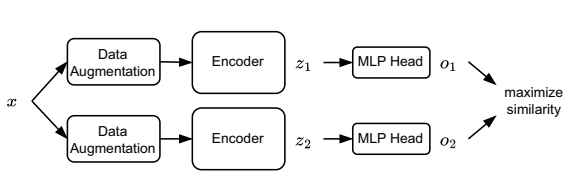

Figure 4: Simplified illustration of a vanilla contrastive learning framework inspired by [8] and [9].

Contrastive learning has become a rising area because of its notable success in Computer Vision. SimCLR [9] is considered one of the pioneers. It is also one of the most simple frameworks. Two views of an image are created by augmentation methods with a random component, such as color distortion and random cropping. They are considered a "positive" example and every other pair in the mini-batch is considered "negative". The cosine similarity of the embeddings, commonly used for visual representations, is measured to compute the temperature-scaled cross-entropy loss.

The success of contrastive learning in visual contexts has prompted its application to textual data as well [16, 40, 22]. For instance, [16] adapted the SimCLR architecture to develop a straightforward method for generating sentence embeddings, demonstrating that complex data augmentation techniques are unnecessary. Instead, they utilized random dropout masks as a minimal augmentation approach, achieving notable performance across various semantic textual similarity tasks. However, their methodology was generally restricted to short textual segments.

Efforts to refine and enhance the basic contrastive learning model have led to various innovations. Different methods of data augmentation, selection of "positive" and "negative" pairs, and adaption of the objective function were proposed [40, 11, 34]. Our work was inspired by the idea of incorporating measures of divergence beyond Euclidean as presented by [35] and [32]. They computed not only the cosine similarity between representations but also added an ensemble of subnetworks on top and considered the functional Bregman divergence.

3 Method

In this study, we enhance self-supervised representation learning by projecting embedding representations into hyperbolic space. Specifically, we propose to investigate two fundamental questions: (i) How can the Hilbert distance be effectively utilized to quantify divergence among embeddings? (ii) How can the Hilbert distance be applied to evaluate similarity among text representations?

These questions serve as the basis for our proposed solutions, which introduce two novel methods for self-supervised representation learning of text based on hyperbolic geometry.

3.1 Self-supervised Representation Network

Consider a randomly sampled mini-batch of sequence data $\boldsymbol{X} = \{\boldsymbol{x}_{s_{(i)}}\}_{i=1}^{N}, \boldsymbol{x_s} \in \mathcal{X} \subseteq \mathbb{R}^p$, the transformation function t derives two augmented views $\boldsymbol{x}_{\boldsymbol{s_{i,1}}} = t(\boldsymbol{x_s}), \boldsymbol{x}_{\boldsymbol{s_{i,2}}} = t'(\boldsymbol{x_s})$ for each sample $\boldsymbol{x}_{s_i} \in \boldsymbol{X}$. The augmented views are obtained by sampling t, t' from a distribution over suitable data augmentations, such as masking parts of sequences [16]. The two augmented views $\boldsymbol{x}_{\boldsymbol{s_{i,1}}}$ and $\boldsymbol{x}_{\boldsymbol{s_{i,2}}}$ are then fed to an encoder network $f_{\boldsymbol{\theta}}$ with trainable parameters $\boldsymbol{\theta} \subseteq \mathbb{R}^d$. The encoder maps distorted samples to a set of corresponding features. We call the output of the encoder the *representation*. The Hilbert network takes the representation vectors $z_{s_{i,1}}$, $z_{s_{i,2}}$, respectively, for $\boldsymbol{x}_{s_{i,1}}$ and $\boldsymbol{x}_{s_{i,2}}$. We define L_{eucl} as:

$$\frac{1}{N} \sum_{i=1}^{N} -log \frac{e^{cos(z_{s_{i,1}}, z_{s_{i,2}})/\tau_e}}{\sum_{k=1}^{2} \sum_{j=1}^{N} e^{cos(z_{s_{i,1}}, z_{s_{j,k}})/\tau_e}} \tag{8}$$

3.2 One-Branch Geometry Fusion Encoding

To enhance the representation, our proposed one-branch algorithm takes the output of two embedding vectors $z_{s_{i,1}}$, $z_{s_{i,2}}$. Each vector

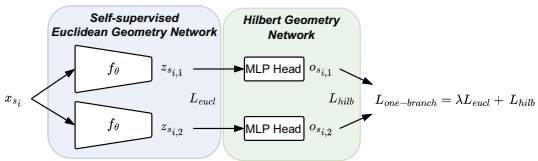

Figure 5: Illustration of our proposed "One-Branch Geometry Fusion Encoding" framework. A sequence from a document is augmented twice and fed to an encoder network to obtain a representation in Euclidean geometry and additionally fed to an MLP Head for one in Hilbert geometry. In each geometry, a contrastive loss is calcluated.

is encoded and then passed through a Hilbert geometry network, denoted as D_{H_ϕ}, which features learnable parameters ϕ. This network is structured with a single Multilayer Perceptron (MLP) head and a softmax layer, producing outputs $o_{s_{i,1}}$ and $o_{s_{i,2}}$ on a simplex. These outputs are then utilized for calculating the loss specific to the respective geometry. Here, we measure the Hilbert distance between positive and negative examples. Additionally, we implement a transformation function ψ that converts distances into similarities, which then formulates the basis for our second objective function, L_{hilb}:

$$\frac{1}{N}\sum_{i=1}^{N} -log\frac{e^{\psi(\tilde{\rho}_H(o_{s_i,1},\, o_{s_i,2}))}}{\sum_{k=1}^{2}\sum_{j=1}^{N} e^{\psi(\tilde{\rho}_H(o_{s_i,1}, o_{s_j,k}))}} \tag{9}$$

in which o_{s_i} represents network outputs, ψ converts divergences into similarity as Eq. (7), and $\tilde{\rho}_H$ is our differentiable Hilbert distance function in Eq. (6). The total loss $L_{one-branch}$ is a linear combination of the Euclidean loss Eq.(8) and the Hilbert loss Eq.(9), weighted by a factor λ, where:

$$\mathcal{L}_{one-branch} = \lambda\, L_{eucl} + L_{hilb} \tag{10}$$

The one-branch geometry fusion coding approach is designed to exploit the structural advantages of both Euclidean and Hilbert spaces in learning document representations. The Euclidean space excels at representing plane geometric structures, while the Hilbert space is more adept at encapsulating hierarchical and intricate geometric configurations. By integrating these spaces, this method seeks to produce a more efficient and expressive representation of documents.

3.3 Two-Branch Geometry Fusion Encoding

The two-branch algorithm takes different granularities of text within the document. As depicted in Fig. 6, a single document x_d is processed alongside portions of its content and paragraphs x_p of text within the document. Our goal was to leverage the inherent suitability of Hilbert geometry for capturing hierarchical relationships more effectively. The similarity in Euclidean geometry is measured between two of the paragraph representations and the similarity in Hilbert geometry is measured between the third paragraph embedding and the document embedding. Our two-branch geometry fusion encoding offers advantages for tasks that require understanding at both the granular level of paragraphs and the holistic level of entire documents. By fusing Euclidean and Hilbert geometries, this algorithm enables a richer and more nuanced understanding of text, effectively capturing both fine-grained details and abstract relationships. The final loss function is computed as:

$$\mathcal{L}_{two-branch} = \lambda' L_{eucl} + L_{hilb} \tag{11}$$

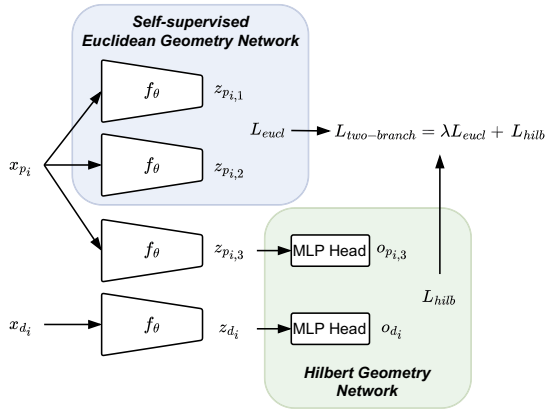

Figure 6: Illustration of our proposed "Two-Branch Geometry Fusion Encoding" framework. A document is fed once and a paragraph from it is fed three times to the encoder. The similarity in Euclidean geometry is measured between two of the paragraph embeddings and the similarity in Hilbert geometry is measured between the third paragraph embedding and the document embedding.

4 Experimental Setup and Results

Datasets We examine our models on two legal datasets as well as a medical text dataset. **ECtHR** [6] is a multi-label classification task, where factual paragraphs from a case description are given and each document is labeled with the article(s) of the European Court of Human Rights (ECHR) believed to have been breached. There are 11k cases and 10 possible labels. **SCOTUS** [7] is a single-label classification task containing 4.7k cases from the Supreme Court of the USA categorized by 14 issue areas. The legal datasets ECtHR-B (ECtHR) and SCOTUS are part of the LexGLUE benchmark by Chalkidis et al. [7]. The medical dataset **MIMIC-III** (MIMIC) [19] is a collection of 50k discharge summaries from US hospitals. It is a multi-label task, where each document is labeled with one or more 1st-level codes from the ICD-9 hierarchy. In total, there are 19 labels.

Division of Documents in Paragraphs For our two-branch algorithm, we needed to divide documents into paragraphs. As ECtHR is a collection of paragraphs pre-defined by the authors, we were able to use them directly. In the case of SCOUTS and MIMIC, we split the documents whenever there were at least two consecutive new lines, potentially preceded by whitespace characters. If the resulting paragraph was shorter than 32 terms, which we regarded as a minimum length of a section with useful content, it was merged with the next unit.

Choice of One Paragraph per Document We had to define pairs of paragraphs and documents. As a document consists of multiple paragraphs, we could either couple every paragraph to the respective document, extract one paragraph for each document, or do something in between. To minimize the computational resources needed, we included every document only once and randomly sampled one paragraph from each document.

To prevent selecting irrelevant paragraphs, like those containing sequences of numbers found in SCOTUS or MIMIC, we assessed the relevance of each paragraph within the documents and excluded those ranking within the lowest 10%. Inspired by the *Inverse Document Frequency (IDF)*, the relevance of a paragraph was determined by the mean *Inverse Paragraph Frequency (IPF)* of all words. Terms common across all paragraphs (e.g. a, the, and) are measured with a

Table 1: Document classification benchmark with micro- and macro-F1 scores and their harmonic averages. SimCSE was implemented following Gao et al. [16]. *DA Longformer* is the domain-adapted Longformer used as a baseline.

	ECtHR		SCOTUS		MIMIC		Average	
Method	mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1
DA Longformer	54.2	35.7	60.9	33.1	66.4	50.5	60.1	38.5
+ SimCSE	59.8	45.4	56.2	29.9	68.0	54.9	61.0	40.7
+ One-branch (ours)	65.0	49.2	65.1	**46.0**	67.1	53.3	65.7	49.3
+ Two-branch (ours)	**65.5**	**53.0**	**66.2**	42.1	**68.9**	**56.4**	**66.9**	**49.7**

Table 2: Comparison of the micro- and macro-F1 scores of SimCSE and One-branch trained either using documents (doc) or paragraphs (par). The harmonic average is reported and training hours are measured by the wall clock time.

	ECtHR		SCOTUS		MIMIC		Average		Training
Method	mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1	**Time (h)**
+ SimCSE (doc)	59.8	45.4	56.2	29.9	68.0	54.9	61.0	40.7	4.6
+ SimCSE (par)	**68.0**	**52.8**	**63.1**	**36.2**	**68.8**	**55.9**	**66.5**	**46.5**	1.2
+ One-branch (doc)	65.0	49.2	65.1	46.0	67.1	53.3	65.7	49.3	4.6
+ One-branch (par)	**68.0**	**51.6**	**64.9**	**44.9**	**68.4**	**55.0**	**67.1**	**50.1**	1.2

lower IPF score than infrequent ones that might be domain-specific. It is calculated by:

$$IPF = \log\left(\frac{\text{number of paragraphs in document}}{\text{number of paragraphs in document containing the term}}\right) \quad (12)$$

4.1 Experimental Setting

Domain-adapted Longformer As our datasets contain long documents with an average number of words of 1613, 1621, and 5853, for ECtHR, MIMIC, and SCOTUS respectively, our model architecture is Longformer [2]. It induces sparsity in the attention mechanism of Transformer using a *sliding window* that focuses on local context. Additionally, we warm-start from models pre-trained on domain-specific corpora. We applied Legal-BERT (small) [5] and BioBERT (medium) [23] for the legal and medical datasets, respectively. The initial positional embeddings were replicated 8 times, resulting in 4096 token positions. Then, most parameters, including word embeddings and Transformer layers, could be transferred to Longformer.

Model Optimization and Hyperparameters Our models were optimized using *AdamW* [25] with 2500 steps per epoch and a weight decay of 0.01. The learning rate was increased from 0 to the maximal learning rate for 10% of the training data before linearly decreasing (*warm-up*). The temperature hyperparameters τ_{eucl} and τ_{hilb} were set to 0.1 and 5, respectively [35, 28]. We performed *Bayesian Optimization and Hyperband* (BOHB) [15] to tune the other hyperparameters (loss weight λ, MLP head embedding size, learning rate, number of epochs, mini-batch size) on the development datasets. The tuning procedure was limited to 50 trials, the search algorithm ran for 20 hours, and the optimization metric was the micro-F1. More details are provided in Section 4.3.

Linear Evaluation We evaluated the models using the standard linear evaluation protocol [9, 11]. A linear classification head on top of the frozen encoder was trained for 20 epochs and a learning rate of $3\text{e}{-5}$. As the datasets are quite imbalanced, we report not only the micro-F1 score but also the macro-F1 score on the test datasets.

4.2 Results and Discussion

The results for document classification are presented in Table 1. The domain-adapted Longformer is compared to SimCSE [16] implemented by ourselves and our algorithms. The results obtained in Table 1 indicate that our proposed method outperforms the baseline and SimCSE-augmented models. The two-branch algorithm shows the best overall performance, suggesting that its strategy for processing and learning from document data is particularly effective for document classification. The mi-F1 and ma-F1 scores are highest with this method, reaching 65.5 and 53.0 in ECHR, 66.2 and 42.1 in SCOTUS, and 68.9 and 56.4 in MIMIC, respectively. According to results obtained in Table 2, paragraph-level training consistently outperforms document-level training across both methods and all datasets. The One-branch algorithm provides the best overall performance, particularly when applied at the paragraph level, suggesting that finer-grained training inputs can lead to more effective document classification. Moreover, paragraph-level methods demonstrate a significant reduction in training time, indicating a more efficient training process without compromising, and even improving, model performance.

4.3 Implementation Details

We will make our code and trained model publicly available to facilitate further research and development in this field. Table 3 outlines the hyperparameter search space used for training in a machine learning context. The hyperparameters include mini-batch size, loss weight λ, MLP head embedding size, learning rate, and the number of epochs. These parameters are crucial for tuning the model to achieve the best performance. The result indicates that the search space for the mini-batch size was constrained by the maximum memory capacity of the GPU, and it was calculated on a log2 scale. This constraint led to differences in the mini-batch sizes available for each algorithm-dataset combination. These hyperparameters are critical to optimize for effective training of the model, with each combination likely to yield different performance results.

Table 4 presents the optimal hyperparameters selected for different document classification methods across three datasets: ECtHR, SCOTUS, and MIMIC. Each row corresponds to a method-dataset combination and lists the best-performing hyperparameters after tuning. Across the datasets, the Two-branch method often has a larger loss weight λ and a lower learning rate compared to the One-branch method, while the mini-batch size is typically smaller. For instance, in the ECtHR dataset, the One-branch method (document level) uses a loss weight of 2, an embedding size of 64, a learning rate of 1e-5, over 15 epochs, with a mini-batch size of 2. In contrast, the Two-branch

Table 3: Search space of the training hyperparameters. The search space of the mini-batch size was limited by the maximal memory capacity of our GPU on log_2-scale resulting in differences between algorithm-dataset combinations.

Search Space of Mini-Batch Size	
One-branch on ECtHR/SCOTUS	$\{2, 4, 8, 16\}$
Two-branch on ECtHR/SCOTUS	$\{2, 4, 8\}$
One-branch on MIMIC	$\{2, 4\}$
Two-branch on MIMIC	$\{2\}$
Search Space of Other Hyperparameters	
loss weight λ	$\{0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$
MLP head embedding size	$\{64, 128\}$
learning rate	$\{1e{-}7, 1e{-}6, 1e{-}5\}$
number of epochs	$\{10, 15, 20\}$

method on the same dataset employs a slightly higher loss weight of 3.5, the same embedding size and learning rate, but fewer epochs (10) and a mini-batch size of 2.

The One-branch method at the paragraph level usually has fewer epochs and smaller mini-batch sizes compared to the document level, suggesting that paragraph-level training may converge faster and require less memory.

Table 4: Tuned hyperparameters (loss weight λ, MLP head embedding size, learning rate, number of epochs, mini-batch size) for every algorithm and dataset.

Dataset	Method	Tuned Values
ECtHR	One-branch (doc)	(2, 64, 1e−5, 15, 2)
	One-branch (par)	(2, 64, 1e−5, 10, 4)
	Two-branch	(3.5, 64, 1e−5, 10, 2)
SCOTUS	One-branch (doc)	(3.5, 128, 1e−5, 20, 2)
	One-branch (par)	(0.5, 64, e−5, 10, 8)
	Two-branch	(3, 128, 1e−6, 15, 4)
MIMIC	One-branch (doc)	(4, 128, 1e−5, 20, 4)
	One-branch (par)	(3, 64, 1e−5, 10, 4)
	Two-branch	(2, 64, 1e−5, 15, 2)

We further assess the generalization capacity of the learned representation on learning a new task. We implemented our two-branch algorithm on the top of DiffCSE and trained on a dataset of sentences from Wikipedia and fine-tuned the pretrained representation for one epoch on seven different semantic textual similarity datasets from the SentEval benchmark suite: MR (movie reviews), CR (product reviews), SUBJ (subjectivity status), MPQA (opinion-polarity), SST-2 (sentiment analysis), TREC (question-type classification), and MRPC (paraphrase detection). Then, we evaluate the test set of each dataset. The results shown in Table 5.

5 Ablation Studies

To examine single components of our two-branch algorithm more closely, we conducted ablation studies. We aimed to better understand the functionality of our method and see how robust it is concerning design choices. Concretely, we investigated multiple possibilities to convert divergences into similarities, worked with paragraphs of varying length, and applied cleansing procedures to the data.

5.1 From Divergence to Similarity

Different from the (cosine) similarity function, a distance function must be converted to represent proximity. Any strictly monotone function could be deployed. [32] explored different possibilities to convert the Bregman divergence to contrast images. Following their work, we tested multiple functions in the Two-branch algorithm using the MIMIC dataset. We only included conversion functions that do not require any hyperparameters, except for the Gaussian kernel for which we could apply the ones tuned by [35]. As expected, the conversion function interacts significantly with the hyperparameters of our algorithm. The results of this interaction are presented in Table 7.

The micro-F1 score does not appear to be affected by the choice of the conversion function as much as the macro-F1 score. While ψ_1 performed the worst, our function ψ_4 performed the best but was comparable to the Gaussian kernel.

5.2 Minimum Length of Paragraphs

We obtained paragraphs from the corpora of documents by splitting them into sections that contain at least 32 terms. To investigate how different values of minimal length affect our methods, we additionally tested our algorithms on paragraphs with at least 64, 128, and 256 terms. In our experiments before, we used the pre-defined paragraphs for ECtHR. Here, we merged paragraphs with the next one, if they were shorter than the minimum length, as done for SCOTUS and MIMIC. This procedure yielded datasets with different sizes shown in Table 8.

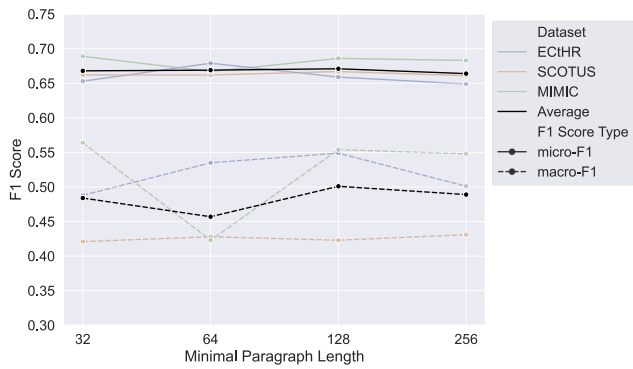

Figure 7: Comparison of our Two-branch algorithm trained using paragraphs of varying minimum lengths. The x-axis is represented on log_2-scale.

Figure 7 depicts the results for Two-branch. For the micro-F1 score, the differences were minimal and a clear tendency was not visible. The macro-F1 scores seemed to be more sensitive. For ECtHR, we observed a better macro-F1 performance with the increasing length of paragraphs up to 128. The score even outperformed Two-branch trained with pre-defined paragraphs by 1.9%. However, on average our Two-branch was not heavily affected by the choice of the minimal paragraph length.

5.3 Data Cleansing

Similar to the reason, why we excluded paragraphs with low relevance in terms of the IPF score in the Two-branch algorithm, cleansing the data might be advantageous. For example, documents in MIMIC often begin with a header that includes dates, gender, etc. Sequences of uppercase letters or newlines are also possible and might affect data quality. To investigate this matter, we tested our models on cleansed data. We removed sequences of uppercase letters, replaced multiple newlines with either a period or space between them by two consecutive newlines, and removed headers from MIMIC.

The results are reported in Table 6. A positive effect of the cleansing procedure was not found. Instead, we observed that it rather tends

Table 5: Comparison on a new baseline (DiffCSE) for the transfer learning task.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
DiffCSE-RoBERTa_base	82.82	88.61	94.32	87.71	88.63	90.40	76.81	87.04
Ours	83.01	88.92	94.37	88.09	88.21	91.02	76.04	87.09

Table 6: Comparison of the micro- and macro-F1 scores of models using data with or without the cleansing procedure.

Method	Cleansing	ECtHR		SCOTUS		MIMIC		Harmonic Mean	
		mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1	mi-F1	ma-F1
Two-branch	with	**66.1**	52.4	66.1	41.6	68.7	55.5	**66.9**	49.0
	without	65.5	**53.0**	**66.2**	**42.1**	**68.9**	**56.4**	**66.9**	**49.7**

Table 7: Comparison of the micro- and macro-F1 scores of our Two-branch algorithm on the MIMIC dataset using different functions suitable for converting divergences into similarities [32], with D being a distance matrix. For the Gaussian kernel ψ_3, we used $\sigma = 2$ as [35] suggested. ψ_4 was used in our models.

Strictly Monotone Functions	mi-F1	ma-F1
$\psi_1 = \sqrt{1-D}$	66.4	50.6
$\psi_2 = 1 - \frac{D}{max(D)}$	67.8	53.3
$\psi_3 = exp(\frac{-D}{2\sigma^2})$	68.7	56.1
$\psi_4 = \frac{1}{D+1}$ (our)	**68.9**	**56.4**

Table 8: Rounded number of paragraphs in the training set after splitting documents into paragraphs with different minimum lengths and using paragraphs pre-defined for ECtHR.

minimum length of paragraphs	ECtHR	SCOTUS	MIMIC
pre-defined paragraphs	213k	-	-
32	159k	214k	472k
64	127k	176k	286k
128	110k	112k	130k
256	105k	31k	54k

to be harmful. The censored and replaced sequences seem not to negatively affect the data quality as we expected.

6 Conclusion

This paper explores the efficacy of non-Euclidean geometries in enriching document representation learning. Our investigation into Hilbert geometry unveils its remarkable potential to enhance the quality of document embeddings. The proposed methods, One-branch and Two-branch, outperform their Euclidean counterparts, demonstrating the effectiveness of Hilbert geometry. Additionally, we uncover that data cleansing is not a critical component of our approach, and that employing paragraphs as training units instead of documents yields enhanced performance and reduced training time.

Ethics Statement

Our proposed model is designed with safeguards to ensure it does not cause harm. All data used for training, testing, and evaluation are sourced from publicly available datasets, adhering to open data principles. Upon acceptance of our work, we will release both the code and the trained model to foster transparency and facilitate further research in the field.

Limitations

In this paper, we concentrate on smaller to medium-sized models, with a parameter count of up to 134 million. This contrasts with recent advancements in LLM, which utilize architectures with billions of parameters. The degree to which the observed enhancements from our network architecture apply to varying model scales or different foundational architectures, such as those used in GPT models, remains uncertain. Additionally, the extent to which our results can be generalized to different application areas, data sets, or influence other NLP tasks like document retrieval and ranking, is yet to be determined.

References

[1] M. G. Atigh, J. Schoep, E. Acar, N. Van Noord, and P. Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4453–4462, 2022.

[2] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[3] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

[4] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[5] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[6] I. Chalkidis, M. Fergadiotis, D. Tsarapatsanis, N. Aletras, I. Androutsopoulos, and P. Malakasiotis. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. *arXiv preprint arXiv:2103.13084*, 2021.

[7] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras. Lexglue: A benchmark dataset for legal language understanding in english. *arXiv preprint arXiv:2110.00976*, 2021.

[8] A. Chaudhary. The illustrated simclr framework, 2020. URL https://amitness.com/2020/03/illustrated-simclr. Accessed: 2023-12-31.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[11] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[12] Craiyon LLC. Craiyon, 2023. URL https://www.craiyon.com/. Accessed: 2023-12-31.

[13] Deepbrain AI. Deepbrain ai studios, 2023. URL https://www.deepbrain.io/aistudios. Accessed: 2023-12-31.

[14] B. Dhingra, C. J. Shallue, M. Norouzi, A. M. Dai, and G. E. Dahl. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*, 2018.

[15] S. Falkner, A. Klein, and F. Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pages 1437–1446. PMLR, 2018.

[16] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[17] S. Ge, S. Mishra, S. Kornblith, C.-L. Li, and D. Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6840–6849, 2023.

[18] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[19] Y. Kementchedjhieva and I. Chalkidis. An exploration of encoder-decoder approaches to multi-label classification for legal and biomedical text. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5828–5843, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.360. URL https://aclanthology.org/2023.findings-acl.360.

[20] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020.

[21] T. Klein and M. Nabi. Scd: Self-contrastive decorrelation for sentence embeddings. *arXiv preprint arXiv:2203.07847*, 2022.

[22] T. Klein and M. Nabi. miCSE: Mutual information contrastive learning for low-shot sentence embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6177, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.339. URL https://aclanthology.org/2023.acl-long.339.

[23] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[24] M. Leimeister and B. J. Wilson. Skip-gram word embeddings in hyperbolic space. *arXiv preprint arXiv:1809.01498*, 2018.

[25] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[26] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

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

[28] F. Nielsen and K. Sun. Non-linear embeddings in hilbert simplex geometry. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 254–266. PMLR, 2023.

[29] OpenAI. Chatgpt [large language model], 2023. URL https://chat.openai.com. Accessed: 2023-12-31.

[30] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

[31] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[32] M. Rezaei, F. Soleymani, B. Bischl, and S. Azizi. Deep bregman divergence for contrastive learning of visual representations. *arXiv preprint arXiv:2109.07455*, 2021.

[33] M. Rezaei, F. Soleymani, B. Bischl, and S. Azizi. Deep bregman divergence for self-supervised representations learning. *Computer Vision and Image Understanding*, 235:103801, 2023.

[34] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[35] D. Saggau, M. Rezaei, B. Bischl, and I. Chalkidis. Efficient document embeddings via self-contrastive bregman divergence learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12181–12190, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.771. URL https://aclanthology.org/2023.findings-acl.771.

[36] D. Saggau, M. Rezaei, B. Bischl, and I. Chalkidis. Efficient document embeddings via self-contrastive bregman divergence learning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12181–12190, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.771. URL https://aclanthology.org/2023.findings-acl.771.

[37] A. Tifrea, G. Bécigneul, and O.-E. Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

[38] A. Vahidi, S. Schosser, L. Wimmer, Y. Li, B. Bischl, E. Hüllermeier, and M. Rezaei. Probabilistic self-supervised representation learning via scoring rules minimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] A. Vahidi, L. Wimmer, H. A. Gündüz, B. Bischl, E. Hüllermeier, and M. Rezaei. Diversified ensemble of independent sub-networks for robust self-supervised representation learning. *ECML*, 2024.

[40] Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, and H. Ma. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020.

[41] Y. Yue, F. Lin, K. D. Yamada, and Z. Zhang. Hyperbolic contrastive learning. *arXiv preprint arXiv:2302.01409*, 2023.

[42] Y. Zhu, D. Zhou, J. Xiao, X. Jiang, X. Chen, and Q. Liu. Hypertext: Endowing fasttext with hyperbolic geometry. *arXiv preprint arXiv:2010.16143*, 2020.