8-2023

Hyperbolic graph topic modeling network with continuously updated topic tree

Ce ZHANG
Singapore Management University, cezhang.2018@phdcs.smu.edu.sg

Rex YING

Hady Wirawan LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Hyperbolic Graph Topic Modeling Network with Continuously Updated Topic Tree

Delvin Ce Zhang
Singapore Management University
Singapore
delvin.ce.zhang@gmail.com

Rex Ying
Yale University
New Haven, Connecticut, USA
rex.ying@yale.edu

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

Connectivity across documents often exhibits a hierarchical network structure. Hyperbolic Graph Neural Networks (HGNNs) have shown promise in preserving network hierarchy. However, they do not model the notion of topics, thus document representations lack *semantic interpretability*. On the other hand, a corpus of documents usually has high variability in degrees of topic specificity. For example, some documents contain general content (e.g., sports), while others focus on specific themes (e.g., basketball and swimming). Topic models indeed model latent topics for semantic interpretability, but most assume a flat topic structure and ignore such *semantic hierarchy*. Given these two challenges, we propose a Hyperbolic Graph Topic Modeling Network to integrate both *network hierarchy* across linked documents and *semantic hierarchy* within texts into a unified HGNN framework. Specifically, we construct a two-layer document graph. Intra- and cross-layer encoding captures network hierarchy. We design a topic tree for text decoding to preserve semantic hierarchy and learn interpretable topics. Supervised and unsupervised experiments verify the effectiveness of our model[1].

CCS CONCEPTS

• **Information systems → Data mining**; **Document topic models**; • **Computing methodologies → Topic modeling**.

KEYWORDS

Hyperbolic Graph Neural Networks; Text Mining; Topic Modeling

ACM Reference Format:
Delvin Ce Zhang, Rex Ying, and Hady W. Lauw. 2023. Hyperbolic Graph Topic Modeling Network with Continuously Updated Topic Tree. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599384

1 INTRODUCTION

Text documents are usually connected in a network structure, e.g., academic papers in a citation network, Web pages in a hyperlink network, etc. Connectivity across documents usually exhibits a hierarchical topological structure, e.g., an academic paper is extended by following research works, which are then further developed by other papers; a breaking news article is traced by following articles reporting subsequent events, etc. Such scenarios contain a central document with hierarchical network links to others (Fig. 1(a)). Hyperbolic Graph Neural Networks (HGNNs) [6, 23] have shown promise in preserving such *network hierarchy* when inferring document embeddings. However, when modeling documents, we usually assume a notion of latent topics [5]. Each document is associated with a topic distribution, and each topic is semantically interpreted by its keywords. Topic model provides *semantic interpretability* [5]. However, existing HGNNs do not assume topic structure, resulting in uninterpretable document embeddings.

A corpus of documents usually has high variability in degrees of topic specificity. Some documents contain general concepts (e.g., survey papers summarize a broad area), while others focus on specific topics (e.g., regular papers deal with specific problems), illustrated by Fig. 1(b). Modeling such *semantic hierarchy* could better preserve latent topics of texts and improve the quality of topic discovery. However, most existing topic models, e.g., LDA [5] and GATON [42], are flat models without hierarchical topics.

Challenges. First, though Hyperbolic Graph Neural Networks can capture network hierarchy, they lack topic modeling component, resulting in uninterpretable document representations. Modeling *semantically interpretable topics* within the corpus allows us to better understand the main themes of documents.

Second, existing topic models are mainly flat models and ignore the *semantic hierarchy* within text documents. Since a corpus may contain documents with different semantic detailedness, modeling semantic hierarchy could reveal insightful topic structure.

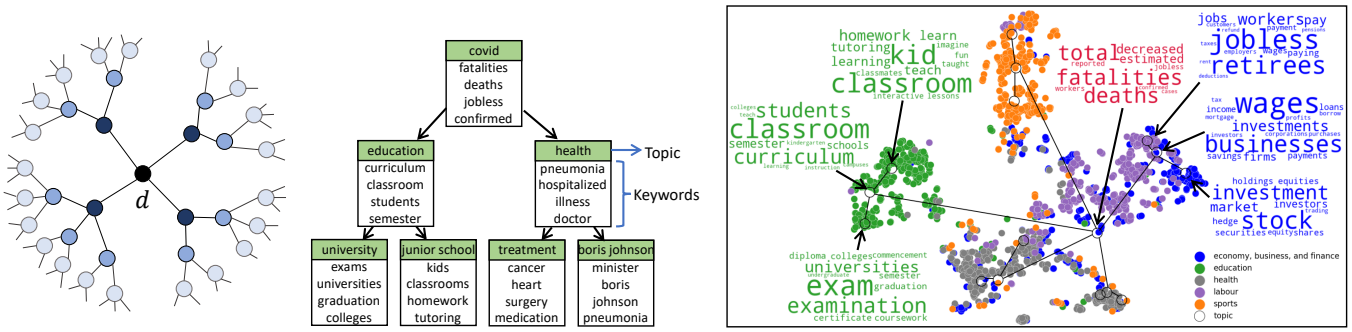
Third, while hierarchical topic models, e.g., nCRP [12], consider semantic hierarchy, they represent each document using only one topic, which is insufficient for documents with a mixture of different topics. They are also not designed for network structure, thus cannot model *network hierarchy* shown by document connectivity.

Approach. Our approach is based on the insight that network and semantic hierarchy can be integrated into a unified framework, i.e., a central document on the network usually describes general semantics, while surrounding documents tend to focus on specific topics. Thus, we design a **H**yperbolic **G**raph **T**opic **M**odeling Network (HGTM) with a continuously updated topic tree to model both network hierarchy across linked documents and semantic hierarchy within text content. *i*) To model network hierarchy, we construct a two-layer document graph, where both intra- and cross-layer hyperbolic topic encoding capture network hierarchy. *ii*) To preserve semantic hierarchy and learn interpretable topics, we design

[1]Code and datasets are available at https://github.com/cezhang01/hgtm.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599384



(a) Network hierarchy (b) Semantic hierarchy of a coronavirus news corpus (c) Visualization of documents' topic proportions with a three-level topic tree

Figure 1: (a) Network hierarchy. (b) Semantic hierarchy. (c) Visualization of documents' topic proportions with a 3-level topic tree. Each topic has keywords for semantic interpretability. Docs close to certain topics denote high probability of semantics.

a novel topic tree for document decoding. The root topic summarizes general concept, while leaf topics become specific. General documents are decoded by root topic, and specific ones sample leaf topics. Moreover, different corpora contain unique hierarchical topic structures. Some corpora present a deep topic tree, while others are relatively flat. To match the semantic hierarchy of different corpora, the topic tree is continuously updated during learning.

Besides, both our topic encoder and tree-structured decoder are in hyperbolic space, not Euclidean. Hyperbolic space has exponentially growing volume and is more suited for hierarchical structure than Euclidean space, whose volume grows only polynomially [6, 23]. Both network and semantic hierarchies show a tree-like structure. The number of nodes on the tree grows exponentially, not polynomially, as the depth increases. Leveraging hyperbolic space could preserve hierarchy and improve document representations. In contrast, previous topic models are designed in Euclidean space.

Fig. 1(a) illustrates network hierarchy. The number of neighbors of the central document grows exponentially. Fig. 1(b) is the semantic hierarchy of a coronavirus news corpus learned by our model. Topics are in a tree structure. Each topic is interpreted by its keywords. Different paths from root to leaves cover different topics (e.g., left paths cover education, right paths cover health). Different levels denotes different topic specificity ("covid→education→university" gradually narrows down the semantics). Fig. 1(c) shows the visualization of documents' topic proportions learned by our model with both network and semantic hierarchies. Documents close to certain topics represent high probability of these topics' semantics, making document representations semantically interpretable. These topics are organized in a three-level tree, indicating semantic hierarchy.

Contributions. *First*, we propose HGTM, which unifies HGNN and topic modeling to jointly model both network hierarchy and semantic hierarchy. To incorporate network hierarchy, we design a two-layer document graph, and simulate intra- and cross-layer topic encoding in hyperbolic space. *Second*, to model semantic hierarchy, we propose a topic tree in hyperbolic space for document decoding. *Third*, to match the unique hierarchical topic structures of different corpora, we design a method to continuously update the topic tree.

2 RELATED WORK

Graph neural networks (GNNs) are mostly in Euclidean space [18, 37]. Recently, hyperbolic space attracts much attention, e.g., HGCN [6] and HGNN [23]. HAT [49] extends GAT in hyperbolic space. LGCN [50] modifies hyperbolic aggregation and transformation. Q-GCN [40] designs a pseudo-Riemannian for hierarchical and spherical graph. κGCN [1] extends to the product of different spaces. They do not have topic modeling nor semantic hierarchy.

Flat topic models are previously graphical models [5]. Recently, neural models are popular [8, 17, 27, 34]. There are topic models with GNN [42, 47, 53]. They are flat models, without topic hierarchy.

Hierarchical topic models extract tree-structured topics. nCRP [4, 12] pioneers this area. Graphical models include [11, 14, 21, 30, 51]. TSNTM [15] and HTV [31] are neural ones. They have semantic hierarchy, but are not designed network hierarchy.

Relational topic models are designed for documents in a network structure, e.g., graphical [7, 19] and neural models [2, 38, 39, 44–46]. These models consider both document content and network connectivity, but no one incorporates network hierarchy.

Text classification models are previously based on CNN [16] and RNN [22]. Recent ones are GNNs [9, 24, 43]. They do not have topic modeling, leading to uninterpretable representations. VGATM [48] integrates topic modeling into GNNs, but still ignores semantic hierarchy and network hierarchy.

3 PROBLEM FORMULATION AND PRELIMINARIES

We formulate the research problem and introduce preliminaries here. Table 1 summarizes main mathematical notations.

3.1 Problem Formulation

We are given a corpus of documents with network structure $C = \{\mathcal{D}, \mathcal{E}\}$. $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^{N}$ is a set of $N = |\mathcal{D}|$ documents. Each document d contains N_d words in the vocabulary $\mathcal{V}$, i.e., $\mathbf{d} = \{w_{d,n}\}_{n=1}^{N_d} \subseteq \mathcal{V}$. $\mathcal{E}$ is a set of network links where $e_{ij} \in \mathcal{E}$ if there is a link between document d_i and d_j. We follow previous works [38, 39, 44] and model an undirected network, i.e., $e_{d_i,d_j} = e_{d_j,d_i}$, though it is straightforward to extend to directed networks. As in [38], if we do not observe network structure $\mathcal{E}$, we instead generate κNN links using documents' content similarity. The neighbors of a document d_i are those directly linked to d_i, denoted as neighbor set $\mathcal{N}(i)$.

Given C as input, we design a topic model that outputs topic proportions $\mathbf{Z}_{\mathcal{D}} = \{\mathbf{z}_d\}_{d\in\mathcal{D}}$ for $N = |\mathcal{D}|$ documents, preserving network topology hierarchy $\mathcal{E}$ and textual semantic hierarchy $\mathcal{D}$.

Table 1: Summary of mathematical notations.

Notation	Description
$\mathcal{C}$	a corpus
$\mathcal{D}$	a set of $N = \|\mathcal{D}\|$ documents
$\mathcal{V}$	vocabulary
$\mathcal{E}$	network links among documents
$\mathcal{N}(i)$	document d_i's neighbor set
$\mathbb{P}^{n,c}$	Poincaré ball space with dimension n and curvature c
$\mathcal{T}_{\mathbf{x}}\mathbb{P}^{n,c}$	tangent (Euclidean) space around hyperbolic vector $x \in \mathbb{P}^{n,c}$
$\exp^c_{\mathbf{x}}(\mathbf{v})$	exponential map, projecting tangent vector $\mathbf{v}$ to Poincaré ball
$\log^c_{\mathbf{x}}(\mathbf{y})$	logarithmic map, projecting hyperbolic vector $\mathbf{y}$ to $\mathbf{x}$'s tangent space
$h_{\mathbb{P}^{n,c}}(\mathbf{x}, \mathbf{y})$	geodesic distance between hyperbolic vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathrm{PT}^c_{\mathbf{x}\to\mathbf{y}}(\mathbf{v})$	parallel transport, transporting $\mathbf{v}$ from $\mathbf{x}$'s tangent space to $\mathbf{y}$'s
H	dimension of topic proportions $\mathbf{z}_d$
$\mathbf{z}_d$	topic proportion of document d
$\boldsymbol{\pi}_d$	path distribution of document d over topic tree
r	a path on topic tree, consisting of topics from root to leaf
$\mathbf{t}_k$	hyperbolic topic embedding of topic k
$\boldsymbol{\delta}_d$	level distribution of document d over topic tree
S	the depth of topic tree
$q(\mathbf{z}_d)$	variational posterior of document d, parameterized by our encoder
$\log p(\cdot\|\cdot)$	log-likelihood term of data generation
$p(\mathbf{z})$	predefined prior distribution
W	dimension of documents' raw input features
$\mathbf{v}_d$	a vector sampled from Euclidean Gaussian
$\boldsymbol{\beta}$	topic-word distribution $\boldsymbol{\beta} \in \mathbb{R}^{\|\mathcal{V}\|\times K}$
M	number of negative samples at Eq. 25

3.2 Preliminaries

Hyperbolic space can better preserve hierarchical structure. We are thus motivated to design the model in hyperbolic space.

Hyperbolic space. *Hyperbolic space* is a non-Euclidean geometry with a constant negative curvature $c < 0$. The curvature c measures how a geometric object deviates from a flat plane. Poincaré ball is one of the representative models of hyperbolic space. In this paper, we use Poincaré ball, though our method is also applicable to other models in hyperbolic space, e.g., hyperboloid model.

Poincaré ball and tangent space. *Poincaré ball* $\mathbb{P}^{n,c}$ is the set of n-dimensional vectors with Euclidean norm smaller than $-\frac{1}{c}$,

$$\mathbb{P}^{n,c} = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{x}, \mathbf{x}\rangle_2 < -\frac{1}{c}\}. \tag{1}$$

$\langle\cdot,\cdot\rangle_2$ is inner product between two vectors. *The tangent space* of Poincaré ball centered at point $\mathbf{x}$ is n-dimensional Euclidean space,

$$\mathcal{T}_{\mathbf{x}}\mathbb{P}^{n,c} = \{\mathbf{v} \in \mathbb{R}^n | \langle \mathbf{v}, \mathbf{x}\rangle_2 = 0\}. \tag{2}$$

Tangent space $\mathcal{T}_x\mathbb{P}^{n,c}$ is a local and first-order approximation of $\mathbb{P}^{n,c}$ around $\mathbf{x}$ and is a Euclidean space. We will use tangent space to perform matrix operations in this paper. For $\mathbf{v}$ and $\mathbf{w}$ in $\mathcal{T}_x\mathbb{P}^{n,c}$, $g^c_{\mathbf{x}}(\mathbf{v}, \mathbf{w}) = \langle \mathbf{v}, \mathbf{w}\rangle_2 \to \mathbb{R}$ is a Riemannian metric tensor [23], and $(\mathbb{P}^{n,c}, g^c_{\mathbf{x}})$ is a Riemannian manifold with negative curvature $c < 0$.

Exponential and logarithmic map. The mapping between Poincaré ball and its tangent space is by exponential and logarithmic map. For each point $\mathbf{x} \in \mathbb{P}^{n,c}$ on Poincaré ball and a tangent vector $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathbb{P}^{n,c}$, the *exponential map* projects $\mathbf{v}$ to Poincaré ball.

$$\exp^c_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{|c|}\frac{\lambda^c_{\mathbf{x}}||\mathbf{v}||_2}{2}\right)\frac{\mathbf{v}}{\sqrt{|c|}||\mathbf{v}||_2}\right) \tag{3}$$

where $\lambda^c_{\mathbf{x}} = \frac{2}{1-c||\mathbf{x}||^2}$, and Möbius addition $\oplus_c$ is [25]

$$x \oplus_c \mathbf{y} = \frac{(1 - 2c\langle \mathbf{x}, \mathbf{y}\rangle_2 - c||\mathbf{y}||_2^2)\mathbf{x} + (1 + c||\mathbf{x}||_2^2)\mathbf{y}}{1 - 2c\langle \mathbf{x}, \mathbf{y}\rangle_2 + c^2||\mathbf{x}||_2^2||\mathbf{y}||_2^2}. \tag{4}$$

Reversely, the *logarithmic map* projects a point $\mathbf{y} \in \mathbb{P}^{n,c}$ $(\mathbf{y} \neq \mathbf{x})$ on Poincaré ball to the tangent space of $\mathbf{x}$ by

$$\log^c_{\mathbf{x}}(\mathbf{y}) = \frac{2}{\sqrt{|c|}\lambda^c_{\mathbf{x}}}\tanh^{-1}\left(\sqrt{|c|}||-\mathbf{x}\oplus_c \mathbf{y}||_2\right)\frac{-\mathbf{x}\oplus_c \mathbf{y}}{||-\mathbf{x}\oplus_c \mathbf{y}||_2}. \tag{5}$$

In this paper, we use $\log^c_{\mathbf{x}}(\cdot)$ with base $\mathbf{x}$ and curvature c to denote hyperbolic logarithmic map, and use $\log(\cdot)$ without base and curvature to denote normal logarithm.

Geodesic distance. *Geodesic distance* in Poincaré ball space is the generalization of the length of a straight line in Euclidean space. Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{P}^{n,c}$, their geodesic distance is

$$h^c_{\mathbb{P}^{n,c}}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{|c|}}\cosh^{-1}\left(1 - \frac{2c||\mathbf{x}-\mathbf{y}||_2^2}{(1 + c||\mathbf{x}||_2^2)(1 + c||\mathbf{y}||_2^2)}\right). \tag{6}$$

In our model, we will use geodesic distance to measure the similarity between two points in Poincaré ball space.

Parallel transport. Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{P}^{n,c}$ $(\mathbf{x} \neq \mathbf{y})$, *parallel transport* can transport a vector on $\mathbf{x}$'s tangent space to $\mathbf{y}$'s.

$$\mathrm{PT}^c_{\mathbf{x}\to\mathbf{y}}(\mathbf{v}) = \frac{\lambda^c_{\mathbf{x}}}{\lambda^c_{\mathbf{y}}}\mathrm{gyr}[\mathbf{y}, -\mathbf{x}]\mathbf{v} \in \mathcal{T}_{\mathbf{y}}\mathbb{P}^{n,c} \tag{7}$$

where $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathbb{P}^{n,c}$, and gyration operator [25] is $\mathrm{gyr}[\mathbf{x}, \mathbf{y}]\mathbf{v} = \ominus_c(\mathbf{x} \oplus_c \mathbf{y}) \oplus_c (\mathbf{x} \oplus_c (\mathbf{y} \oplus_c \mathbf{v}))$. Here $\mathbf{x} \ominus_c \mathbf{y} = \mathbf{x} \oplus_c -\mathbf{y}$.

4 MODEL ARCHITECTURE AND ANALYSIS

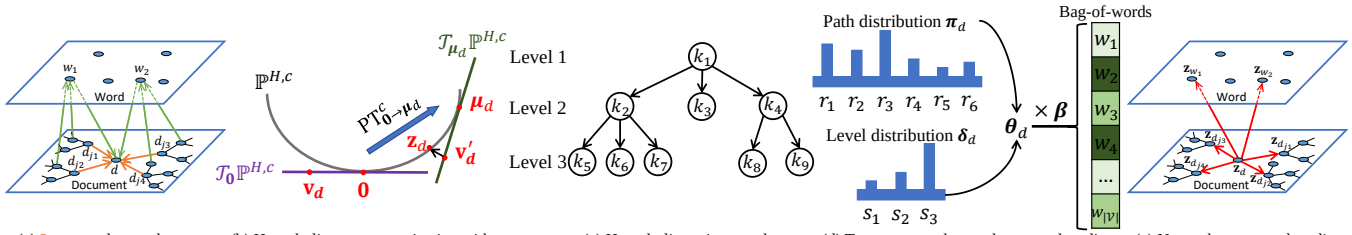
We introduce the details of **H**yperbolic **G**raph **T**opic **M**odeling Network (HGTM) with a continuously updated topic tree.

4.1 Generative Process

As an overview, we describe the generative process of HGTM. Given $\mathcal{C}$, we generate documents $\mathcal{D}$ and network $\mathcal{E}$ with topic tree.

(1) For each word $w \in \mathcal{V}$:
  (a) Draw H-dim hyperbolic topic proportion $\mathbf{z}_w \sim p(\mathbf{z}_w)$.
(2) For each document $d \in \mathcal{D}$:
  (a) Draw H-dim hyperbolic topic proportion $\mathbf{z}_d \sim p(\mathbf{z}_d)$.
  (b) Obtain path distribution over topic tree $\boldsymbol{\pi}_d = f_\pi(\mathbf{z}_d)$.
  (c) Obtain level distribution over topic tree $\boldsymbol{\delta}_d = f_\delta(\mathbf{z}_d)$.
  (d) For each word $w_{d,n}$ in document d where $n = 1, 2, ..., N_d$:
    (i) Draw a path $r_{d,n} \sim \mathrm{Categorical}(\boldsymbol{\pi}_d)$.
    (ii) Draw a level $s_{d,n} \sim \mathrm{Categorical}(\boldsymbol{\delta}_d)$.
    (iii) Draw a word $w_{d,n} \sim \mathrm{Categorical}(\boldsymbol{\beta}_{r_{d,n},s_{d,n}}, \mathbf{z}_d, \mathbf{z}_{w_{d,n}})$.
  (e) If d's label exists, draw its label $y_d \sim p(y_d|\mathbf{z}_d)$.
(3) For each pair of documents d_i and d_j where $d_i, d_j \in \mathcal{D}$:
  (a) Draw a network link $e_{d_i,d_j} \sim p(e_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j})$.

To preserve semantic hierarchy and learn interpretable topics, we design a hyperbolic topic tree and generate document content based on the tree at Step 2. Specifically, different paths represent different topics, while different levels on the same path denote different topic specificity. Root topic summarizes general concept, topics close to leaves focus on specific sub-concepts. This is why each document with its unique content has its own path and level distributions over the tree (Step 2b–2c). For each document, we generate its words by repeatedly sampling a path and a level at Step 2d. Thus general documents tend to be decoded by root topic, and specific ones sample leaf topics. Semantic hierarchy can thus be captured.



(a) Intra- and cross-layer topic encoding (b) Hyperbolic reparameterization with parallel transport and obtain $\mathbf{z}_d \sim \mathcal{WN}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$ (c) Hyperbolic topic tree where each node k is a topic. (d) Tree-structured textual content decoding. $\boldsymbol{\beta}$ is topic-word distribution. (e) Network structure decoding

Figure 2: Model architecture. (a) Given a corpus with text content and network connections, we construct a two-layer graph. We simulate intra- and cross-layer topic propagation to capture network hierarchy. (b) We use hyperbolic reparameterization to sample topic proportions for documents. (c-d) We design a latent topic tree in hyperbolic space for hierarchical text decoding to capture semantic hierarchy. (e) Finally, we use learned topic proportions of documents to reconstruct the network structure.

After drawing a path $r_{d,n}$ and a level $s_{d,n}$ (Step 2(d)i–2(d)ii), we already select one topic. Thus $\boldsymbol{\beta}_{r_{d,n},s_{d,n}}$ is the topic-word distribution of the selected topic, used to semantically interpret the topic.

We aim to maximize the log-likelihood $\mathcal{L}(C)$ of above generative process. Directly maximizing the log-likelihood is intractable, we follow VAE [17] and instead maximize its evidence lower bound.

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{Z}_\mathcal{D}, \mathbf{Z}_\mathcal{V})} \Big(\sum_{d \in \mathcal{D}} \big[\lambda_{\text{text}} \log p(\mathbf{d} | \mathbf{z}_d, \mathbf{Z}_\mathcal{V}) + \lambda_{\text{label}} \log p(\mathbf{y}_d | \mathbf{z}_d) \big] + \sum_{d_i, d_j \in \mathcal{D}} \log p(e_{d_i,d_j} | \mathbf{z}_{d_i}, \mathbf{z}_{d_j}) \Big) - \Big(\text{KL}\big[q(\mathbf{Z}_\mathcal{D}) || p(\mathbf{Z}_\mathcal{D})\big] + \text{KL}\big[q(\mathbf{Z}_\mathcal{V}) || p(\mathbf{Z}_\mathcal{V})\big] \Big) \tag{8}$$

Upper letter $\mathbf{Z}_\mathcal{D} \in \mathbb{P}^{N \times H}$ denotes a collection of hyperbolic topic proportions of all N documents in Poincaré ball. The topic proportion of each document is H-dimensional. $\mathbf{d}$ is the text content of d. λ_{text} and λ_{label} are hyperparameters, controlling the importance of respective term. $\lambda_{\text{label}} = 0$ for unsupervised training if labels are not observed, otherwise $\lambda_{\text{label}} > 0$ for supervised training. $q(\mathbf{Z}_\mathcal{D}, \mathbf{Z}_\mathcal{V})$ is variational posterior with structured mean-field assumption $q(\mathbf{Z}_\mathcal{D}, \mathbf{Z}_\mathcal{V}) = q(\mathbf{Z}_\mathcal{D}) q(\mathbf{Z}_\mathcal{V}) = \prod_{d \in \mathcal{D}} q(\mathbf{z}_d) \prod_{w \in \mathcal{V}} q(\mathbf{z}_w)$.

Variational posteriors $q(\mathbf{z}_d)$ and $q(\mathbf{z}_w)$ are topic encoders, producing H-dimensional hyperbolic topic proportions of documents and words. Log-likelihoods $\log p(\cdot|\cdot)$ are decoders. Specifically, the textual content log-likelihood $\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_\mathcal{V})$ is a hyperbolic tree-structured topic decoder. Below we elaborate the details of hyperbolic graph topic encoder and tree-structured topic decoder. We then introduce how to update the topic tree to match the semantic hierarchy of different corpora. Finally, we design KL divergence and objective function. See Fig. 2 for an overview of model architecture.

4.2 Hyperbolic Graph Topic Encoder

Hyperbolic graph convolutional topic encoders $q(\mathbf{z}_d)$ and $q(\mathbf{z}_w)$ project documents and words to H-dimensional topic proportions in hyperbolic space. Given a corpus C, considering documents and words as vertices, we construct a bipartite graph. Intra-layer links within document layer are document connections. Cross-layer links are word occurrences in the document, shown by Fig. 2(a).

4.2.1 ***Intra-Layer Topic Encoding.*** The orange arrows at Fig. 2(a) show the direction of intra-layer propagation.

Hyperbolic feature initialization. Since documents' raw input features, e.g., Bag-of-Words or the average of pretrained word embeddings, are usually in Euclidean space, we first project them into Poincaré ball in order to do subsequent hyperbolic operations. Let W-dimensional zero vector $\mathbf{0} = [0, 0, ..., 0]^\top \in \mathbb{P}^{W,c}$ denote the origin of Poincaré ball, where W is the dimension of documents' input features. We discover that input features $\mathbf{x}_d^E$ lie in the tangent space of the origin by definition, $\mathbf{x}_d^E \in \mathcal{T}_\mathbf{0}\mathbb{P}^{W,c}$, due to $\langle \mathbf{x}_d^E, \mathbf{0} \rangle_2 = 0$ where $\langle \cdot, \cdot \rangle_2$ is inner product. We use superscript $(^E)$ to denote Euclidean features. We thus consider the origin $\mathbf{0}$ as a reference point and map Euclidean features $\mathbf{x}_d^E$ to Poincaré ball by exponential map.

$$\mathbf{z}_d^{(l=0)} = \exp_\mathbf{0}^c(\mathbf{x}_d^E) \in \mathbb{P}^{W,c}. \tag{9}$$

Here $\mathbf{z}_d^{(l=0)}$ is the hyperbolic input feature in Poincaré ball space with curvature c. We will explain superscript $(^{l=0})$ shortly.

Feature transformation. To learn low-dimensional topic proportions for documents, we have hyperbolic feature transformation,

$$\tilde{\mathbf{z}}_d'^{(l)} = \exp_\mathbf{0}^c(\mathbf{W}^{(l)} \log_\mathbf{0}^c(\mathbf{z}_d^{(l-1)})) \in \mathbb{P}^{n,c}. \tag{10}$$

l is the l-th graph convolutional step. Previous works [13, 37] call it the l-th convolutional layer. But in order to differentiate from our two-layer document graph, we instead name it convolutional step. $\mathbf{z}_d^{(l-1)} \in \mathbb{P}^{n',c}$ is the hyperbolic topic proportion output from previous convolutional step. $\mathbf{z}_d^{(l=0)}$ is the hyperbolic input feature obtained at Eq. 9. $\mathbf{W}^{(l)} \in \mathbb{R}^{n \times n'}$ is learnable parameter in Euclidean space. Eq. 10 first projects hyperbolic topic proportion $\mathbf{z}_d^{(l-1)}$ to tangent space $\mathcal{T}_\mathbf{0}\mathbb{P}^{n',c}$ by logarithmic map. It then performs matrix multiplication in tangent (Euclidean) space, whose result is projected back to hyperbolic space by exponential map. n and n' are the dimensions of the l-th and $(l-1)$-th convolutional step.

To perform bias addition, we adopt parallel transport. We define learnable bias $\mathbf{b}^{(l)} \in \mathbb{R}^n$ as Euclidean parameter, located in the tangent space of the origin, $\mathbf{b}^{(l)} \in \mathcal{T}_\mathbf{0}\mathbb{P}^{n,c}$, again due to $\langle \mathbf{b}^{(l)}, \mathbf{0} \rangle_2 = 0$. We then parallel transport $\mathbf{b}^{(l)}$ to the tangent space of $\tilde{\mathbf{z}}_d'^{(l)}$. Finally, we map it back to hyperbolic space by exponential map.

$$\tilde{\mathbf{z}}_d^{(l)} = \exp_{\tilde{\mathbf{z}}_d'^{(l)}}^c(\text{PT}_{\mathbf{0} \to \tilde{\mathbf{z}}_d'^{(l)}}^c(\mathbf{b}^{(l)})) \in \mathbb{P}^{n,c}. \tag{11}$$

Neighbor aggregation. A document's neighbors share latent semantics to different extent. We here design hyperbolic attention.

$$a_{ij} = \text{softmax}\Big(\sigma\big(\boldsymbol{\beta}^{(l)\top} [\log_\mathbf{0}^c(\tilde{\mathbf{z}}_{d_i}^{(l)}) || \log_\mathbf{0}^c(\tilde{\mathbf{z}}_{d_j}^{(l)})]\big)\Big) \text{ where } d_j \in \mathcal{N}(i). \tag{12}$$

$\sigma(x) = \frac{1}{1+e^{-x}}$ is sigmoid, $[\cdot||\cdot]$ is concatenation, $\boldsymbol{\beta}^{(l)} \in \mathbb{R}^{2n}$ is learnable Euclidean parameter. Hyperbolic points are mapped to tangent space to evaluate attention. We then aggregate d_i's neighbors by

$$\mathbf{z}_{d_i}^{(l)} = f_{\text{act}}^{c,c'}\Big(\exp_{\mathbf{0}}^{c}\Big(\frac{1}{2}\big(\log_{\mathbf{0}}^{c}(\tilde{\mathbf{z}}_{d_i}^{(l)}) + \sum_{d_j \in \mathcal{N}(i)} a_{ij} \log_{\mathbf{0}}^{c}(\tilde{\mathbf{z}}_{d_j}^{(l)})\big)\Big)\Big). \quad (13)$$

Hyperbolic points are mapped to tangent space for aggregation, whose result is mapped back to hyperbolic space. Finally, hyperbolic activation $f_{\text{act}}^{c,c'}(\mathbf{x}) = \exp_{\mathbf{0}}^{c'}(f_{\text{act}}(\log_{\mathbf{0}}^{c}(\mathbf{x})))$ produces the output of the current l-th convolutional step. We set $f_{\text{act}}(x) = x$ for the final step, and $f_{\text{act}}(x) = \tanh(x)$ for previous steps. To summarize,

$$\mathbf{z}_{d_i,\text{intra}}^{(l)} = f_{\text{act}}^{c,c'}\Big(\mathbf{z}_{d_i}^{(l-1)}, \{\mathbf{z}_{d_j}^{(l-1)} | d_j \in \mathcal{N}(i)\}\Big) \text{ where } l = 1, 2, ..., L. \quad (14)$$

L is the max convolutional step. We finally get H-dimensional hyperbolic topic proportion $\mathbf{z}_{d,\text{intra}}^{(L)} \in \mathbb{P}^{H,c}$ from intra-layer encoder.

4.2.2 ***Cross-Layer Topic Encoding.*** We design our topic encoder on top of HGNN [6, 23]. However, HGNN has only intra-layer encoding. Since we model a document graph with both documents and words, we extend HGNN and design cross-layer topic encoding. A document contains many words, and these words in turn appear in more documents. Such graph hierarchy is captured by cross-layer links, shwon by green arrows at Fig. 2(a). The encoding is similar to Eq. 9–13, except that the parameters $\mathbf{W}^{(l)}$ and $\boldsymbol{\beta}^{(l)}$ are replaced with cross-layer ones. Propagating from words to documents, we obtain H-dimensional topic proportions from cross-layer encoder.

$$\mathbf{z}_{d,\text{cross}}^{(l)} = f_{\text{act}}^{c,c'}\Big(\mathbf{z}_{d}^{(l-1)}, \{\mathbf{z}_{w_{d,n}}^{(l-1)} | w_{d,n} \in \mathbf{d}\}\Big) \text{ where } l = 1, 2, ..., L. \quad (15)$$

$\mathbf{d} = \{w_{d,n}\}_{n=1}^{N_d}$ is d's words. Symmetrically, propagating from documents to words, we have H-dimensional topic proportion for words.

$$\mathbf{z}_{w}^{(l)} = f_{\text{act}}^{c,c'}\Big(\mathbf{z}_{w}^{(l-1)}, \{\mathbf{z}_{d}^{(l-1)} | \forall d, w \in \mathbf{d}\}\Big) \text{ where } l = 1, 2, ..., L. \quad (16)$$

Hyperbolic reparameterization. Finally, we unify intra- and cross-layer topic proportions by hyperbolic mean pooling.

$$\boldsymbol{\mu}_d = \exp_{\mathbf{0}}^{c}\Big(\frac{1}{2}\big(\log_{\mathbf{0}}^{c}(\mathbf{z}_{d,\text{intra}}^{(L)}) + \log_{\mathbf{0}}^{c}(\mathbf{z}_{d,\text{cross}}^{(L)})\big)\Big) \in \mathbb{P}^{H,c}. \quad (17)$$

Since we aim to output both mean and covariance from the final convolutional step, we repeat the final L-th step twice with different parameters, and obtain $\boldsymbol{\mu}_d$ and Σ_d for each document d. We use reparameterization to sample topic proportion of document d, i.e., $\mathbf{z}_d \sim q(\mathbf{z}_d) = \mathcal{WN}(\boldsymbol{\mu}_d, \Sigma_d)$. Here $\mathcal{WN}(\cdot,\cdot)$, wrapped Gaussian distribution [33] with hyperbolic mean and covariance, is the hyperbolic version of Euclidean Gaussian. To sample from wrapped Gaussian, we first sample an instance $\mathbf{v}_d$ from Euclidean Gaussian $\mathbf{v}_d \sim \mathcal{N}(\mathbf{0}, \log_{\mathbf{0}}^{c}(\Sigma_d))$ by Eq. 18. $\mathbf{v}_d \in \mathcal{T}_{\mathbf{0}}\mathbb{P}^{H,c}$ by definition. The final topic proportion $\mathbf{z}_d$ is obtained by first transporting $\mathbf{v}_d$ to $\boldsymbol{\mu}_d$'s tangent space, then mapping it to the hyperbolic space (Eq. 19).

$$\mathbf{v}_d = \mathbf{0} + \big(\log_{\mathbf{0}}^{c}(\Sigma_d)\big)^{1/2}\boldsymbol{\epsilon} \in \mathcal{T}_{\mathbf{0}}\mathbb{P}^{H,c} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (18)$$

$$\mathbf{z}_d = \exp_{\boldsymbol{\mu}_d}^{c}(\text{PT}_{\mathbf{0}\to\boldsymbol{\mu}_d}^{c}(\mathbf{v}_d)) \in \mathbb{P}^{H,c}. \quad (19)$$

$\mathbf{z}_d \in \mathbb{P}^{H,c}$ is H-dimensional hyperbolic topic proportion of d. It is the output from our hyperbolic graph topic encoder, i.e., $\mathbf{z}_d \sim q(\mathbf{z}_d)$. We similarly repeat Eq. 18–19 and obtain $\mathbf{z}_w \in \mathbb{P}^{H,c}$ for words. The hyperbolic reparameterization process is shown by Fig. 2(b).

4.3 Probabilistic Decoder

Having demonstrated the design of hyperbolic graph topic encoders $q(\mathbf{z}_d)$ and $q(\mathbf{z}_w)$, we now turn to probabilistic decoders, i.e., log-likelihood terms $\log(\cdot|\cdot)$ at Eq. 8. Specifically, the textual content log-likelihood $\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_{\mathcal{V}})$ is a hyperbolic topic tree decoder: we input topic proportions of document $\mathbf{z}_d$ and one of its words $\mathbf{z}_{w_{d,n}}$ to the decoder, and design a topic tree to decode the word by $\log p(w_{d,n}|\mathbf{z}_d, \mathbf{z}_{w_{d,n}})$. We repeat this process for every word $w_{d,n} \in \mathbf{d}$ and obtain $\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_{\mathcal{V}})$. Here, we first elaborate tree-structured topic decoder, then describe other log-likelihood terms.

4.3.1 ***Hyperbolic Tree-Structured Topic Decoder.*** Given the topic proportion $\mathbf{z}_d$ of a document d, we first evaluate its probability distribution over paths and levels on the topic tree (Fig. 2(c-d)).

Path distribution. A path on the tree consists of a sequence of topics. They belong to similar semantics with different specificity. Root topic shows general concept, leaf topics discuss specific sub-concepts. Different paths cover different semantics. For a tree with depth S, the probability of path r with topics $r = \{k_s\}_{s=1}^{S}$ is

$$p(r) = p(\{k_s\}_{s=1}^{S}) = p(k_S|k_{S-1}) \cdots p(k_2|k_1)p(k_1). \quad (20)$$

Topic $k_1 = k_{\text{root}}$ is the root topic, and topic k_S is a leaf topic. Topic k_{s-1} is the parent of topic k_s. $p(k_1) = 1$, since a path must go through the root topic. We now define the conditional probability $p(k_s|k_{s-1})$, i.e., standing at parent topic k_{s-1}, what is the probability of selecting one of its children $k_s \in \text{Child}(k_{s-1})$. We have

$$p(k_s|k_{s-1}) = \frac{(1 + h_{\mathbb{P}^{H,c}}(\mathbf{z}_d, \mathbf{t}_{k_s})^2)^{-1}}{\sum_{k'_s \in \text{Child}(k_{s-1})}(1 + h_{\mathbb{P}^{H,c}}(\mathbf{z}_d, \mathbf{t}_{k'_s})^2)^{-1}}. \quad (21)$$

$\mathbf{t}_{k_s} \in \mathbb{P}^{H,c}$ is H-dimensional hyperbolic topic embedding of topic k_s. We parameterize it by $\mathbf{t}_{k_s} = \exp_{\mathbf{0}}^{c}(\mathbf{t}_{k_s}^{E})$ where $\mathbf{t}_{k_s}^{E} \in \mathbb{R}^{H}$ is learnable Euclidean parameter and lies in the tangent space of the origin, $\mathbf{t}_{k_s}^{E} \in \mathcal{T}_{\mathbf{0}}\mathbb{P}^{H,c}$. $h_{\mathbb{P}^{H,c}}(\cdot,\cdot)$ is geodesic distance. Eq. 21 evaluates the probability over topic k_{s-1}'s children. The lower the distance between document and one child, the higher the probability, the more likely the path goes through this child topic. Putting Eq. 21 into Eq. 20, we obtain probability of one path. We repeat this process for every path, and obtain path distribution $\boldsymbol{\pi}_d = [p(r_1), p(r_2), ...]^{\top}$. Path distribution is document-specific, since different documents contain diverse topics, and select different paths for content generation.

Level distribution. A path contains multiple topics, each with a different depth or level. After sampling a path, we aim to sample a level on the selected path to determine the specific topic for content generation. The level distribution depends on the specificity of the document. A document with general concepts tends to draw low-depth topics, a document with concrete content likely samples high-depth topics. For a tree with depth S, we have

$$p(s) = \frac{(1 + h(s)^2)^{-1}}{\sum_{s'=1}^{S}(1 + h(s')^2)^{-1}} \quad (22)$$
$$\text{where } h(s)^2 = \min\{h_{\mathbb{P}^{H,c}}(\mathbf{z}_d, \mathbf{t}_{k_s})^2 | \forall k_s \text{ on depth } s\}.$$

We first take min pooling for geodesic distances between a document and all the topics on the same level or depth. We then normalize the probability. The reason of min pooling is to select the most representative topic on each level for calculation. Level distribution

$\boldsymbol{\delta}_d = [p(s_1), p(s_2), ...]^\top$ is also document-specific, since a corpus of documents may contain texts with different degrees of specificity.

After sampling a path r and a level s, we narrow down to one topic k. Given r and s, the probability of topic k is $p(r) \times p(s)$. Since there are multiple paths going through the same topic k, the overall probability of this topic is $p(k) = p(s) \sum_{r':k \in r'} p(r')$, i.e., the summation of all the paths going through k. Finally, for a document d, we calculate the probability of every topic and obtain its hierarchical topic distribution $\boldsymbol{\theta}_d = [p(k_1), p(k_2), ..., p(K)]^\top$. Here we assume there are totally K topics on the tree.

Textual content log-likelihood. Following previous topic models [8, 27, 34], we generate the observed words of document d by

$$\begin{aligned} \log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_{\mathcal{V}}) &= \sum_{w_{d,n} \in \mathbf{d}} \log p(w_{d,n}|\mathbf{z}_d, \mathbf{z}_{w_{d,n}}) \\ &= \sum_{w_{d,n} \in \mathbf{d}} \log f(\boldsymbol{\beta}\boldsymbol{\theta}_d) \propto \frac{1}{N_d} \sum_{w_{d,n} \in \mathbf{d}} \log f(\boldsymbol{\beta}\boldsymbol{\theta}_d). \end{aligned} \tag{23}$$

$f(x) = \text{softmax}(x) = \frac{e^x}{\sum_{x'} e^{x'}}$ is softmax function. $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{V}| \times K}$ is Euclidean parameter, representing topic-word distribution, used to semantically interpret the topics. Here we divide by document length N_d to avoid long documents dominating the log-likelihood.

Topic tree regularizer. One requirement is to make sure that child topic k_s indeed inherits a sub-concept of its parent k_{s-1}, but not other parents k'_{s-1}, so that topics on one path indeed belong to similar semantics. To do so, we design a topic tree regularizer, forcing child k_s to be closer to its parent k_{s-1} than other topics.

$$\mathcal{L}_{\text{reg}} = \frac{1}{2} \sum_{i \neq j} (h_{\mathbb{P}^{H,c}}(\mathbf{t}_{k_{s_i}}, \mathbf{t}_{k_{s_j}})^2 - g(k_{s_i}, k_{s_j}))^2. \tag{24}$$

Here $g(k_{s_i}, k_{s_j})$ is the length between topics k_{s_i} and k_{s_j} on the tree, i.e., the number of edges connecting k_{s_i} and k_{s_j}. For example, the length between a child and its parent is one, and the length between two siblings is two. Topics belonging to the same branch tend to have low lengths, while topics from different subtrees present high lengths. We will add this regularizer to the final objective function.

4.3.2 ***Other log-likelihood terms.*** So far, we have focused on textual content log-likelihood. We now turn to the discussion of other terms. For network structure log-likelihood, we have

$$\log p(e_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j}) = \log \phi(\mathbf{z}_{d_i}, \mathbf{z}_{d_j}) - \textstyle\sum_{m=1}^{M} \mathbb{E}_{d'_j \sim p_n(d)} \log \phi(\mathbf{z}_{d_i}, \mathbf{z}_{d'_j}). \tag{25}$$

$\phi(\mathbf{z}_{d_i}, \mathbf{z}_{d_j}) = (1 + e^{h_{\mathbb{P}^{H,c}}(\mathbf{z}_{d_i}, \mathbf{z}_{d_j})^2})^{-1}$ is Fermi-Dirac decoder [29]. $h_{\mathbb{P}^{H,c}}(\cdot,\cdot)$ is geodesic distance. We use negative sampling [28] with M negative samples. $p_n(d)$ is a noise distribution. To preserve both intra- and cross-layer graph structure, we similarly design another log-likelihood between documents and words by replacing $\mathbf{z}_{d_j}$ at Eq. 25 with topic proportions of d_i's words (Fig. 2(e)).

If document d's label exists, we define label log-likelihood by

$$\hat{\mathbf{y}} = \text{softmax}(f_{\text{MLP}}(\log_{\mathbf{0}}^c(\mathbf{z}_d))), \quad \log p(\mathbf{y}_d|\mathbf{z}_d) = \sum_{i=1}^{|\mathbf{y}_d|} y_{d,i} \log \hat{y}_{d,i}. \tag{26}$$

$f_{\text{MLP}}(\cdot)$ is multi-layer perceptron [3], $\mathbf{y}_d$ is one-hot label encoding.

To summarize, hyperbolic graph topic encoder captures network hierarchy by intra- and cross-layer encoding. Tree-structured topic decoder preserves semantic hierarchy by text content generation.

4.4 Continuously Updating the Topic Tree

Different corpora contain documents with different topic structures. Some corpora have documents of various topic specificities, leading to a deep topic tree, while others present a relatively flat tree. To match the semantic hierarchy of different corpora, we design a heuristic method to continuously update the tree during training. More complicated design is our future work.

We evaluate the proportion of words assigned to topic k by $\gamma_k = \frac{\sum_{d \in \mathcal{D}} N_d \theta_{d,k}}{\sum_{d \in \mathcal{D}} N_d}$. N_d is the number of words in d. $\boldsymbol{\theta}_d = [\theta_{d,k}]_{k=1}^K$ is the topic distribution of d. A high γ_k means topic k captures too many concepts, thus children should be added to split its concepts. A low γ_k indicates a topic with overly specific semantic and keeping it may cause overfitting. We remove it as well as its descendants, since descendants even capture more specific and redundant semantics.

4.5 KL Divergence and Objective Function

KL divergence. The KL divergence $\text{KL}[q(\mathbf{z}_d)||p(\mathbf{z}_d)]$ in the evidence lower bound Eq. 8 serves as prior regularizer, which pushes variational posterior $q(\mathbf{z}_d)$ to a predefined prior $p(\mathbf{z}_d)$. Different from VAE [17], whose KL divergence is in Euclidean space, both our prior and variational posterior are in hyperbolic space. Thus our KL divergence does not have a closed-form solution. We instead use Monte Carlo sampling [3] to estimate KL divergence.

$$\begin{aligned} \text{KL}[q(\mathbf{z}_d)||p(\mathbf{z}_d)] &= q(\mathbf{z}_d) \log \frac{q(\mathbf{z}_d)}{p(\mathbf{z}_d)} = \mathbb{E}_{\mathbf{z}_d \sim q(\mathbf{z}_d)} \log \frac{q(\mathbf{z}_d)}{p(\mathbf{z}_d)} \\ &= \mathbb{E}_{\mathbf{z}_d \sim q(\mathbf{z}_d)} (\log q(\mathbf{z}_d) - \log p(\mathbf{z}_d)) \end{aligned} \tag{27}$$

where $\mathbf{z}_d \sim q(\mathbf{z}_d)$ is the output from our hyperbolic graph topic encoder (Eq. 18–19). In this paper, both prior and variational posterior are wrapped Gaussian, $q(\mathbf{z}_d) = \mathcal{WN}(\boldsymbol{\mu}_d, \Sigma_d)$ and $p(\mathbf{z}_d) = \mathcal{WN}(\mathbf{0}, \mathbf{I})$, where $\boldsymbol{\mu}_d$ and Σ_d are obtained by our encoder (Eq. 17).

Now the problem becomes how to evaluate $\log \mathcal{WN}(\cdot,\cdot)$. Once we obtain it, we can evaluate $\log q(\mathbf{z}_d) - \log p(\mathbf{z}_d)$ to estimate KL divergence. Fortunately, existing work [33] provides the solution.

Theorem 4.1. *The logarithm of wrapped Gaussian is [33]*

$$\begin{aligned} \log \mathcal{WN}(\boldsymbol{z}; \boldsymbol{\mu}, \Sigma) &= \log \mathcal{N}(\boldsymbol{v}; \boldsymbol{0}, \log_{\boldsymbol{0}}^c(\Sigma)) \\ &\quad - (H-1) \log \Big(\frac{\frac{1}{\sqrt{|c|}} sinh(\sqrt{|c|}||\boldsymbol{u}||_2)}{||\boldsymbol{u}||_2} \Big). \end{aligned} \tag{28}$$

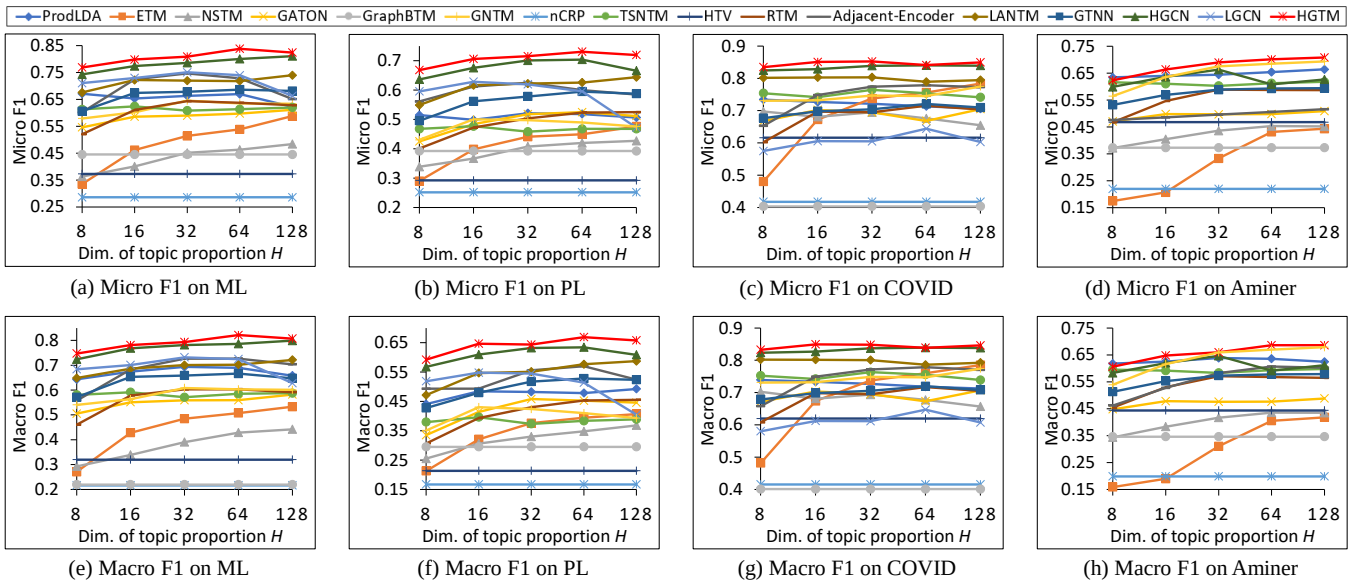
$\boldsymbol{u} = \log_{\boldsymbol{0}}^c(\boldsymbol{z})$, $\boldsymbol{v} = PT_{\boldsymbol{\mu} \to \boldsymbol{0}}^c(\boldsymbol{u})$. H is the dimension of representation $\boldsymbol{z}$.

The theorem provides an explicit equation to calculate $\log \mathcal{WN}(\cdot,\cdot)$. We replace $\mathbf{z}$ in Eq. 28 with topic proportion of documents $\mathbf{z}_d$, the output from our encoder, to obtain $\log q(\mathbf{z}_d)$ and $\log p(\mathbf{z}_d)$. Finally, we calculate $\log q(\mathbf{z}_d) - \log p(\mathbf{z}_d)$ to estimate KL divergence. We investigate the effect of KL divergence at the Experiments section.

Objective function. We have elaborated every necessary component. Hyperbolic graph topic encoder captures network hierarchy and outputs topic proportions of documents and words. Tree-structured topic decoder designs a topic tree to preserve semantic hierarchy. KL divergence is prior regularizer. Putting all components together, we obtain the final objective function for minimization.

$$\mathcal{L} = -\mathcal{L}_{\text{ELBO}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}. \tag{29}$$

We add topic tree regularizer $\mathcal{L}_{\text{reg}}$ at Eq. 24. Algo. 1 at Appendix summarizes the training process of our model.



(a) Micro F1 on ML (b) Micro F1 on PL (c) Micro F1 on COVID (d) Micro F1 on Aminer

(e) Macro F1 on ML (f) Macro F1 on PL (g) Macro F1 on COVID (h) Macro F1 on Aminer

Figure 3: Unsupervised document classification when varying the dimension of topic proportions H from 8 to 128.

Table 2: Dataset statistics.

Name	#Documents	#Links	Vocabulary	#Labels
ML	3,087	8,573	2,885	7
PL	2,597	7,754	3,106	9
COVID	1,500	5,706	5,083	5
Aminer	114,741	265,345	10,018	10
Web	445,657	565,502	10,015	N.A.

5 EXPERIMENTS

The main objective of experiments is to evaluation the quality of the learned topic proportions of documents.

Datasets. Table 2 shows five datasets. Cora [26] is a corpus of papers with abstract as content and citations as links. We follow [54] and created two independent datasets, Machine Learning (**ML**) and Programming Language (**PL**). **Aminer** [35] is another citation network. **COVID** is a Coronavirus news corpus. Since no links are observed, we created κNN links using $tf-idf$ similarity. **Web** [20] is a Web page hyperlink network. Each page is a news with hyperlinks to related pages. Appendix A.2 shows dataset preprocessing.

Baselines. We consider 5 classes of baselines. *i*) **Flat topic models**, ProdLDA [34], ETM [8], NSTM [52], GATON [42], GraphBTM [53], GNTM [32]. These *unsupervised* models treat all topics equally without semantic hierarchy. By comparison, we highlight the benefit of topic tree to differentiate documents with different topic specificity. *ii*) **Hierarchical topic models**, nCRP [12], TSNTM [15], HTV [31]. They model semantic hierarchy in Euclidean space in an *unsupervised* way, but do not consider network hierarchy. *iii*) **Topic models for document graphs** consider both text and network links for *unsupervised* training, RTM [7], Adjacent-Encoder [44], LANTM [38], GTNN [39]. They are all in Euclidean space. By comparison, we show the utility of using hyperbolic space to better preserve network hierarchy. *iv*) **Hyperbolic graph neural networks** derive node embeddings on graphs in hyperbolic space. Strictly speaking, they are not topic models, nor baselines. For completeness, we still compare to HGCN [6] and LGCN [50]. *v*) **Text classification models** learn text embeddings with *label supervision*. They are not topic models. We mainly compare to GNN models, TextGCN [43], HyperGAT [9], HINT [41]. HINT designs a topic tree in Euclidean space for text classification.

We set $L = 2$ convolutional steps. $\lambda_{\text{text}} = \lambda_{\text{reg}} = 5$. M at Eq. 25 is 5. We initialize the topic tree with three levels, and each topic has three children. For the supervised version, $\lambda_{\text{label}} = 5$. For ProdLDA and RTM, they perform the best with 1 and 2 as Dirichlet parameter, respectively. Each result is obtained by 5 independent runs.

5.1 Quantitative Evaluation

5.1.1 ***Document classification.*** As in LDA [5], we use document classification to evaluate the quality of topic proportions. We split 80% documents for training (of which 10% for validation), 20% for testing. During training, we only observe training documents and links within them. During testing, we infer topic proportions of test documents and classify them. We conduct two classification tasks, corresponding to unsupervised and supervised versions.

Unsupervised training. We set $\lambda_{\text{label}} = 0$. Labels are never involved for training. After convergence, we follow previous works [44] and train an external κNN classifier ($\kappa = 5$) using topic proportions of training documents and predict the labels of test documents. Fig. 3 shows Micro and Macro F1 with different dimensions of topic proportions. LANTM, LGCN, and TextGCN cannot run on large dataset Aminer even on a machine with 256GB, thus are excluded.

Supervised training. Labels are involved for supervised training. For a fair comparison, we pick the best baseline from each class of baselines, then manually add a multi-layer perceptron $f_{\text{MLP}}(\cdot)$ as classifier to create their supervised version. These baselines are GATON, TSNTM, GTNN, HGCN. For completeness, we also show the results of other unsupervised models. Table 3 shows the results.

Analysis. For both tasks, TSNTM performs better than flat models, due to semantic hierarchy to differentiate documents. By using

Table 3: Supervised document classification with Micro F1 (left) and Macro F1 (right) at $H = 16$. Results are in percentage.

Category	Model	Micro F1 score				Macro F1 score			
		ML	PL	COVID	Aminer	ML	PL	COVID	Aminer
Flat topic models (GATON is converted to *supervised* version)	ProdLDA	65.3±1.0	49.8±2.5	72.7±1.7	69.1±0.1	67.4±1.4	48.4±1.8	73.3±1.7	67.5±0.1
	ETM	46.2±1.2	39.8±0.8	67.2±1.8	20.7±0.9	42.8±1.6	32.1±1.2	67.4±1.7	19.0±0.9
	NSTM	40.1±1.9	36.7±2.3	68.1±1.7	40.5±0.2	33.9±2.0	30.6±1.4	68.1±1.8	38.3±0.3
	GATON	72.8±0.7	61.6±1.3	80.3±1.9	57.9±0.4	70.5±1.2	53.6±2.0	80.1±2.0	54.8±0.5
	GraphBTM	44.5±1.0	39.2±0.6	40.3±0.3	37.3±0.4	31.9±0.6	29.5±1.2	40.1±0.4	34.6±0.2
	GNTM	60.2±3.1	50.0±1.9	73.2±2.2	63.6±0.6	56.4±3.4	43.0±1.6	73.2±2.1	61.6±0.7
Hierarchical topic models (TSNTM is converted to *supervised* version)	nCRP	28.6±1.7	25.2±2.5	41.7±4.4	20.5±0.8	21.6±1.8	16.7±2.5	41.5±4.5	15.8±0.3
	TSNTM	72.8±1.5	63.3±0.5	84.1±1.3	71.5±0.1	68.6±1.3	56.1±0.8	84.0±1.2	67.3±0.1
	HTV	37.3±4.2	29.2±5.4	61.6±4.3	46.8±0.5	32.0±4.1	21.3±4.7	61.9±4.7	44.4±0.1
Topic models for document graphs (GTNN is converted to *supervised* version. LANTM cannot run on large dataset Aminer even on a machine with 256GB memory)	RTM	61.0±0.9	47.2±1.4	69.7±2.5	54.7±0.9	57.8±1.2	39.3±0.8	69.8±2.3	53.0±0.8
	Adjacent-Encoder	72.5±1.1	61.2±1.0	74.8±2.4	64.5±0.3	68.3±1.0	49.3±0.6	69.8±2.3	62.7±0.4
	LANTM	72.2±0.7	61.7±1.1	80.3±1.7	N.A.	68.6±1.0	54.6±1.2	80.2±1.7	N.A.
	GTNN	75.0±0.7	61.3±1.0	77.8±2.6	63.1±0.2	73.1±0.9	53.3±1.6	77.0±3.6	61.1±0.2
Hyperbolic GNNs (HGCN is converted to *supervised* version. LGCN cannot run on large dataset Aminer)	HGCN	82.6±1.3	70.3±1.0	86.0±0.5	67.6±1.0	81.2±1.3	65.9±1.3	85.8±0.5	65.5±1.0
	LGCN	73.0±2.2	62.8±2.6	60.5±5.0	N.A.	70.1±2.8	54.8±3.2	61.2±4.8	N.A.
Text classification (designed with label *supervision*. TextGCN cannot run on large dataset Aminer)	TextGCN	78.3±0.7	67.5±0.7	83.7±0.5	N.A.	76.0±0.8	61.4±1.1	79.6±0.5	N.A.
	HyperGAT	80.0±0.4	65.8±2.5	84.3±1.2	**74.2±1.5**	78.9±0.5	60.2±2.5	81.3±0.8	**72.0±1.1**
	HINT	69.5±1.1	55.4±2.3	85.7±1.5	66.5±0.5	64.8±3.9	44.3±3.2	85.8±1.5	59.5±0.3
Our proposed model (*supervised* version)	HGTM	**83.8±0.5**	**72.2±1.4**	**86.3±1.7**	70.0±0.3	**82.6±0.7**	**67.4±2.0**	**86.2±1.9**	68.5±0.3

Table 4: Link prediction AUC (in percentage) at $H = 16$.

Model	ML	PL	COVID	Aminer	Web
ProdLDA	82.5±0.2	78.9±0.4	80.1±0.8	93.4±0.1	82.4±0.0
ETM	70.3±1.6	72.5±1.1	87.2±0.9	63.0±1.2	79.4±0.0
NSTM	65.1±0.7	64.7±0.8	71.0±1.6	67.4±0.8	67.0±0.8
GATON	75.9±1.5	64.5±0.5	70.2±1.2	82.2±0.8	87.6±0.1
GraphBTM	69.2±2.5	68.9±1.0	70.3±1.3	71.2±1.4	N.A.
GNTM	79.3±0.8	73.2±0.3	76.8±0.4	91.8±0.3	86.3±0.0
nCRP	58.0±0.9	60.1±3.1	70.8±0.8	70.7±2.0	57.2±0.0
TSNTM	77.8±1.5	75.5±0.9	70.8±0.8	90.4±0.7	87.4±0.8
HTV	69.9±1.9	68.3±4.6	86.0±2.0	85.5±0.3	86.1±0.8
RTM	75.7±0.5	69.0±0.4	69.8±0.2	81.7±0.3	78.4±0.1
Adjacent-Encoder	81.0±1.1	80.8±1.7	79.8±0.6	92.6±0.3	73.2±0.0
LANTM	78.7±0.7	82.2±0.3	93.6±0.3	N.A.	N.A.
GTNN	76.6±0.9	73.7±1.2	84.3±1.0	84.6±0.4	74.3±0.2
HGCN	89.7±0.4	90.3±0.3	94.8±0.3	94.2±0.1	90.5±0.2
LGCN	89.2±0.4	90.8±0.5	93.4±0.5	N.A.	N.A.
TextGCN	76.5±0.5	68.2±0.4	87.1±0.4	N.A.	N.A.
HyperGAT	82.0±0.8	77.5±1.0	87.1±0.4	90.0±0.0	N.A.
HINT	71.7±1.4	69.7±1.4	86.6±0.2	89.8±0.1	N.A.
HGTM	**89.9±0.8**	**91.3±0.3**	**95.7±0.2**	**95.9±0.2**	**91.3±0.1**

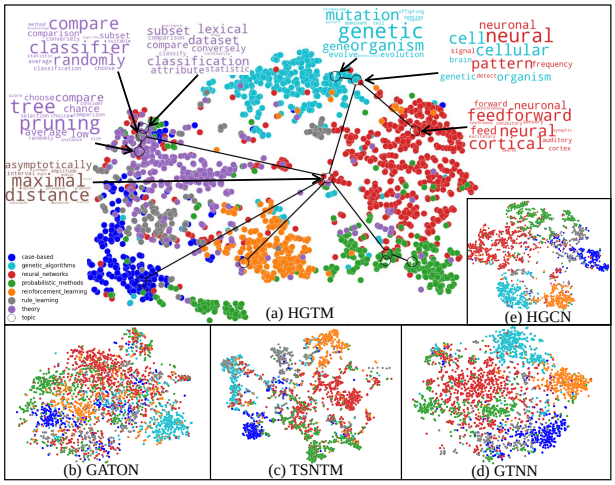


Figure 4: Visualization on ML dataset.

hyperbolic space for network, HGCN preserves network hierarchy and is the best baseline. Compared to HGCN, we further model semantic hierarchy with topic tree, improving the results. Compared to hierarchical models, we achieve improvement due to network hierarchy. HyperGAT performs better on Aminer, possibly because it has word-word connections. We are still better than other baselines.

5.1.2 ***Link Prediction.*** As in RTM [7], we predict links on the document network. During training, we observe 80% training documents and links within them. During testing, we predict links within 20% testing documents. Following previous works [38, 44], the probability of a link for Euclidean baselines is $p(e_{ij}) \propto e^{-||\mathbf{z}_{d_i} - \mathbf{z}_{d_j}||^2}$, while the probability for hyperbolic baselines is Fermi-Dirac decoder at Eq. 25. We compare the predicted probability with the ground-truth adjacency with AUC as metric. Table 4 shows the results. TextGCN and HyperGAT cannot run on Web with no labels.

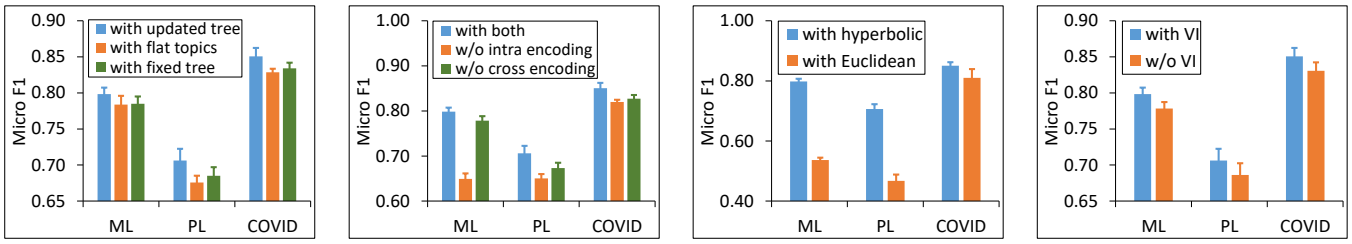
Our model predicts links more accurately than baselines, due to the modeling of both semantic and network hierarchy. Compared to hierarchical topic models with topic tree, we showcase the advantage of modeling network hierarchy. Compared to hyperbolic GNNs, we verify the benefit of semantic hierarchy.

5.2 Topic Analysis

5.2.1 ***Topic Coherence.*** An important property of topic models is semantic interpretability. In our model, decoding parameter $\boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{V}| \times K}$ is topic-word distribution. Each column is the distribution of a topic over the vocabulary, and the highest values on that column are the keywords of the topic. As in ProdLDA [34], we evaluate the coherence of keywords by an external corpus, Google Web 1T 5-gram Version 1 [10], with NPMI as metric. Table 5 (left) shows the results. We exclude TextGCN, HyperGAT, HGCN, LGCN, since they are not topic models and cannot evaluate coherence. Our model generates more coherent words than flat models, since we construct a topic tree to model topic relationship. Connected topics on the tree complement each other to improve coherence. TSNTM performs slightly better on Web, possibly because it models pretrained word embeddings. Our model is still better than most baselines on Web.

Table 5: Topic coherence NPMI (left, in percentage, higher is better) and perplexity (right, lower is better) at $H = 16$.

Category	Model	Topic Coherence NPMI (higher is better)					Perplexity (lower is better)				
		ML	PL	COVID	Aminer	Web	ML	PL	COVID	Aminer	Web
Flat topic models	ProdLDA	10.9±0.7	12.1±0.7	12.0±0.7	9.3±0.5	21.2±0.2	7.97±0.00	7.99±0.00	7.82±0.00	8.18±0.00	8.34±0.00
	ETM	7.1±0.2	8.7±0.1	8.2±0.7	5.4±0.3	16.4±0.6	7.96±0.00	7.94±0.00	7.80±0.00	8.31±0.00	8.52±0.00
	NSTM	17.2±0.7	19.2±0.7	22.0±0.6	15.5±0.3	24.0±0.3	7.83±0.00	7.80±0.00	8.38±0.00	9.00±0.00	8.93±0.00
	GATON	17.4±1.0	5.4±1.1	13.8±1.2	19.4±1.5	4.8±1.1	8.37±0.02	8.38±0.03	8.42±0.00	9.25±0.03	8.33±0.00
	GraphBTM	5.1±0.5	7.0±0.4	10.4±0.1	8.2±0.3	N.A.	7.09±0.01	7.04±0.02	7.87±0.00	7.92±0.01	N.A.
	GNTM	12.1±0.3	15.4±0.7	13.8±0.8	17.3±0.4	23.8±0.3	6.91±0.01	6.83±0.01	7.69±0.01	7.81±0.00	7.79±0.00
Hierarchical topic models	nCRP	2.2±0.1	2.2±0.1	3.0±0.1	0.2±0.1	2.8±0.0	6.94±0.02	6.87±0.02	7.69±0.05	7.99±0.02	7.71±0.04
	TSNTM	12.1±0.6	15.1±0.8	14.1±0.8	17.6±0.8	**26.6±2.3**	6.92±0.01	6.83±0.01	7.64±0.04	7.85±0.01	**7.35±0.03**
	HTV	10.8±1.0	13.3±1.8	16.6±2.5	17.2±0.6	26.5±0.9	6.95±0.02	6.83±0.03	7.62±0.04	7.97±0.00	7.44±0.01
Topic models for document graphs (LANTM cannot run on large datasets Aminer and Web even on 256GB machine)	RTM	7.1±0.3	9.3±0.2	16.2±0.5	10.8±0.3	20.9±0.4	7.46±0.05	7.52±0.05	8.98±0.04	8.89±0.01	10.28±0.19
	Adjacent-Encoder	9.9±0.9	11.3±0.9	13.8±0.4	11.4±0.2	15.2±0.1	7.65±0.05	7.62±0.04	6.96±0.00	8.71±0.02	8.26±0.01
	LANTM	5.4±0.3	7.2±0.8	8.6±0.3	N.A.	N.A.	8.63±0.00	8.48±0.00	8.48±0.00	N.A.	N.A.
	GTNN	7.2±0.6	5.8±0.6	13.5±2.7	12.6±0.5	7.9±1.6	7.75±0.02	7.73±0.01	7.96±0.00	9.39±0.01	8.26±0.01
Text classification (cannot run on Web with no labels)	HINT	6.6±2.2	8.6±2.4	11.6±3.0	12.1±3.3	N.A.	8.45±0.08	8.51±0.28	8.84±0.12	10.04±0.54	N.A.
Our proposed model	HGTM	**19.0±2.6**	**21.9±2.8**	**23.3±3.1**	**20.5±1.4**	25.0±1.7	**6.89±0.02**	**6.81±0.00**	**7.60±0.01**	**7.78±0.01**	7.71±0.01



(a) Effect of topic tree structure (b) Effect of intra and cross encoding (c) Effect of hyperbolic modeling (d) Effect of variational inference

Figure 5: Ablation analysis of our model.

5.2.2 ***Perplexity.*** We evaluate perplexity [5], $e^{-\frac{\log p(\mathcal{D}_{\text{test}})}{\sum_{d' \in \mathcal{D}_{\text{test}}} N_{d'}}}$, for 20% test documents. Since perplexity is exponential, we instead report its power, $-\frac{\log p(\mathcal{D}_{\text{test}})}{\sum_{d' \in \mathcal{D}_{\text{test}}} N_{d'}}$. Lower is better. Table 5 (right) shows that benefiting from semantic hierarchy, TSNTM performs well among baselines. Compared to it, we further consider network hierarchy to improve the performance, since topological structure also reveals the centrality and the hierarchy of documents.

5.2.3 ***Topic Visualization.*** We use t-SNE [36] to learn 2D topic proportions for documents for visualization at Fig 4. Every topic has keywords, but we only show some topics for clarity. Our model and HGCN present similar separation between categories based on visual observation. However, we further learn topic embeddings and keywords of each topic for semantic interpretability.

5.3 Model Analysis

Effect of topic tree structure. We designs a latent topic tree to preserve semantic hierarchy. To verify its effectiveness, we conduct two experiments. *i*) To investigate the hierarchical tree structure, we remove the topic tree and treat all topics equally. *ii*) To test the advantage of the continuously updated structure, we fix the topic tree during training. Fig. 5(a) shows that changing the decoding topics to a flat structure hurts the results, since topics cannot preserve the semantic hierarchy, thus deteriorating topic proportions. Fixing the topic tree also influences the results, since the predefined tree may not be suited for the corpus, leading to semantic mismatch.

Effect of intra- and cross-layer encoding. We remove each encoding respectively from the complete model. With both encodings, we perform the best at Fig. 5(b), revealing the advantage of both encodings to capture network hierarchy. Intra-layer encoding is more informative, since disregarding it leads to the worst results.

Effect of hyperbolic space modeling. Using hyperbolic space for network link reconstruction can better preserve network hierarchy than Euclidean space. To verify network hierarchy is indeed better preserved, we replace all hyperbolic operations with the Euclidean counterparts, while keeping all necessary components. The only difference is the modeling spaces. Fig. 5(c) contrasts the performance. Hyperbolic space achieves better results, since its exponentially growing volume can better model hierarchically expanding network than Euclidean space with polynomial growth.

Effect of variational inference. Our model is built on variational inference with Monte Carlo sampling and KL divergence. To evaluate its effect, we remove it and report the results at Fig. 5(d). Our model with variational components classifies documents more accurately. This is because Monte Carlo sampling introduces a random noise to topic proportions at each training epoch, improving model robustness. The model without variational components does not have sampling or KL regularizer, thus may suffer overfitting.

6 CONCLUSION

We propose HGTM, a hyperbolic graph topic modeling network that learns interpretable document representations. We design intra- and cross-layer topic encoding to capture network hierarchy, and a continuously updated topic tree to preserve semantic hierarchy.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

REFERENCES

[1] Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. 2020. Constant curvature graph convolutional networks. In *International Conference on Machine Learning*. PMLR, 486–496.

[2] Haoli Bai, Zhuangbin Chen, Michael R Lyu, Irwin King, and Zenglin Xu. 2018. Neural relational topic models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 27–36.

[3] Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 4. Springer.

[4] David M Blei, Thomas L Griffiths, and Michael I Jordan. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)* 57, 2 (2010), 1–30.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems* 32 (2019).

[7] Jonathan Chang and David Blei. 2009. Relational topic models for document networks. In *Artificial intelligence and statistics*. PMLR, 81–88.

[8] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics* 8 (2020), 439–453.

[9] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be More with Less: Hypergraph Attention Networks for Inductive Text Classification. In *Proceedings of the 2020 EMNLP*. 4927–4936.

[10] Stefan Evert. 2010. Google web 1t 5-grams made easy (but not for the computer). In *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop*. 32–40.

[11] Zhe Gan, Changyou Chen, Ricardo Henao, David Carlson, and Lawrence Carin. 2015. Scalable deep Poisson factor analysis for topic modeling. In *International Conference on Machine Learning*. PMLR, 1823–1832.

[12] Thomas Griffiths, Michael Jordan, Joshua Tenenbaum, and David Blei. 2003. Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems* 16 (2003).

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[14] Ricardo Henao, Zhe Gan, James Lu, and Lawrence Carin. 2015. Deep Poisson factor modeling. *Advances in Neural Information Processing Systems* 28 (2015).

[15] Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2020. Tree-structured neural topic model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 800–806.

[16] Yoon Kim. 2014. Convolutional neural networks for sentence classification. EMNLP.

[17] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[18] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[19] Tuan MV Le and Hady W. Lauw. 2014. Probabilistic latent document network embedding. In *2014 IEEE International Conference on Data Mining*. IEEE, 270–279.

[20] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 497–506.

[21] Wei Li and Andrew McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*. 577–584.

[22] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 2873–2879.

[23] Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems* 32 (2019).

[24] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8409–8416.

[25] Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. 2020. Differentiating through the fréchet mean. In *International Conference on Machine Learning*. PMLR, 6393–6403.

[26] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163.

[27] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*. PMLR, 1727–1736.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[29] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems* 30 (2017).

[30] John Paisley, Chong Wang, David M Blei, and Michael I Jordan. 2014. Nested hierarchical Dirichlet processes. *IEEE transactions on pattern analysis and machine intelligence* 37, 2 (2014), 256–270.

[31] Dang Pham and Tuan Le. 2021. Neural Topic Models for Hierarchical Topic Detection and Visualization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 35–51.

[32] Dazhong Shen, Chuan Qin, Chao Wang, Zheng Dong, Hengshu Zhu, and Hui Xiong. 2021. Topic modeling revisited: A document graph-based neural network perspective. *Advances in Neural Information Processing Systems* 34 (2021), 14681–14693.

[33] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. 2019. Mixed-curvature Variational Autoencoders. In *International Conference on Learning Representations*.

[34] Akash Srivastava and Charles Sutton. 2017. Autoencoding Variational Inference for Topic Models. In *5th International Conference on Learning Representations*.

[35] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[38] Yiming Wang, Ximing Li, and Jihong Ouyang. 2021. Layer-assisted neural topic modeling over document networks. In *International Joint Conference on Artificial Intelligence*. 3148–3154.

[39] Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021*. 3055–3065.

[40] Bo Xiong, Shichao Zhu, Nico Potyka, Shirui Pan, Chuan Zhou, and Steffen Staab. 2022. Pseudo-Riemannian Graph Convolutional Networks. *Advances in Neural Information Processing Systems* 36 (2022).

[41] Hanqi Yan, Lin Gui, and Yulan He. 2022. Hierarchical interpretation of neural text classification. *Computational Linguistics* 48, 4 (2022), 987–1020.

[42] Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. 2020. Graph attention topic modeling network. In *Proceedings of The Web Conference 2020*. 144–154.

[43] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.

[44] Ce Zhang and Hady W. Lauw. 2020. Topic modeling on document networks with adjacent-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6737–6745.

[45] Delvin Ce Zhang and Hady W. Lauw. 2021. Semi-supervised semantic visualization for networked documents. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*. Springer, 762–778.

[46] Delvin Ce Zhang and Hady W. Lauw. 2022. Dynamic topic models for temporal document networks. In *International Conference on Machine Learning*. PMLR, 26281–26292.

[47] Delvin Ce Zhang and Hady W. Lauw. 2022. Meta-Complementing the Semantics of Short Texts in Neural Topic Models. *Advances in Neural Information Processing Systems* 35 (2022), 29498–29511.

[48] Delvin Ce Zhang and Hady W. Lauw. 2022. Variational Graph Author Topic Modeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2429–2438.

[49] Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Ye. 2021. Hyperbolic graph attention network. *IEEE Transactions on Big Data* 8, 6 (2021), 1690–1701.

[50] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. 2021. Lorentzian graph convolutional networks. In *Proceedings of the Web Conference 2021*. 1249–1261.

[51] He Zhao, Lan Du, Wray Buntine, and Mingyuan Zhou. 2018. Dirichlet belief networks for topic structure learning. *Advances in neural information processing systems* 31 (2018).

[52] He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. 2020. Neural Topic Model via Optimal Transport. In *ICLR*.

[53] Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. GraphBTM: Graph enhanced autoencoded variational inference for biterm topic model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*.

[54] Shenghuo Zhu, Kai Yu, Yun Chi, and Yihong Gong. 2007. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 487–494.

Algorithm 1 Training Process of HGTM

Input: Corpus C with documents $\mathcal{D}$ and network links $\mathcal{E}$, number of convolutional steps L, number of negative samples M, a predefined tree structure.

Output: Topic proportions $\mathbf{Z}_{\mathcal{D}}$ and $\mathbf{Z}_{\mathcal{W}}$, and topic-word distribution $\boldsymbol{\beta}$.

Initialize model parameters.
while not converged **do**
  //intra-layer topic encoding
  for $l = 1, 2, ..., L$ **do**
    Simulate intra-layer topic encoding by Eq. 10–14.
  end for
  //cross-layer topic encoding
  for $l = 1, 2, ..., L$ **do**
    Simulate cross-layer topic encoding by Eq. 15–16.
  end for
  Hyperbolic reparameterization by Eq. 17–19.
  //probabilistic decoding
  Obtain path distribution by Eq. 20–21.
  Obtain level distribution by Eq. 22.
  Evaluate objective function with log-likelihood terms (Eq. 8 and Eq. 25), KL divergence (Eq. 27), and tree-structured regularizer (Eq. 24).
  //optimization
  Minimize objective function Eq. 29.
  //update topic tree structure
  Update topic tree based on Sec. 4.4.
end while

A REPRODUCIBILITY SUPPLEMENT

A.1 Pseudo-Code of Training Process

We summarize the training process of our model at Algo. 1.

A.2 Dataset Preprocessing

Here we introduce the details of dataset preprocessing. Code and datasets are submitted, and will be released upon publication.

Cora[2] is publicly available dataset with academic papers as textual content and citations as links. We created two independent datasets, Machine Learning (**ML**) and Programming Language (**PL**). For both datasets, after removing stop words and punctuations, we kept the most frequent 3,000 words as the vocabulary.

COVID[3] is a publicly available coronavirus news corpus from multiple publishers. Each document is a news article and is associated with a category. We selected five categories, *economy, business, and finance*, *education*, *health*, *labour*, and *sports*. For each category, we randomly selected 300 news articles, forming a corpus of 1,500 articles in total. Similarly, after removing stop words and punctuations, we kept the most frequent 5,000 words as the vocabulary. Since we did not observe links connecting these news articles, we instead compared documents' $tf-idf$ similarity and induced links by κNN ($\kappa = 5$), resulting in 5,706 links in total.

[2]http://people.cs.umass.edu/~mccallum/data/cora-classify.tar.gz
[3]https://aylien.com/coronavirus-news-dataset/

Table 6: Categories and venues of Aminer dataset

Category	Venues
Computational Linguistics	ACL, EMNLP, NAACL, COLING, EACL
Databases and Information Systems	SIGMOD, VLDE, ICDE, CIKM, IPM
Data Mining and Analysis	KDD, WWW, ICDM, TKDE, SIGIR
Computer Vision and Pattern Recognition	CVPR, ICCV, ECCV, TPAMI, TIP
Artificial Intelligence	NeurIPS, ICML, AAAI, IJCAI, JMLR
Computer Graphics	TOG, TVCG, SIGGRAPH, CGA, TVS
Theoretical Computer Science	STOC, SODA, FOCS, JOC, JACM
Software Systems	ICSE, ASE, FSE, TSE, PLDI
Computer Networks and Wireless Communication	SIGCOMM, INFOCOM, TWC, CM, JNCA
Computing Systems	TPDS, ISCA, TJSC, ICDCS, ATC

Aminer[4] is another academic corpus with abstract as document content and citations as links. We used *ACM-Citation-network V8* as the raw dataset. Since we did not discover any categories of these academic papers, we labeled documents based on their publication venues. Specifically, we used Google Scholar Metrics[5] as ground-truth categories. We selected 10 computer science categories. For each category, we selected 5 representative conferences or journals, resulting in totally 50 venues. Table 6 summarizes the details of these venues. Again, after removing stop words and punctuations, we maintained the most frequent 10,000 words as vocabulary.

Web[6] is a web page hyperlink network. Each page is a news article containing the most frequent phrases and quotes. Each page has hyperlinks to other related pages. After removing stop words and punctuatinos, we kept documents with links and more than 30 words, resulting in 445,657 documents and 565,505 links in total. We did not observe any ground-truth categories of these documents.

A.3 Experiment Environment

All the experiments were conducted on Linux server with a Tesla K80 GPU with 11441MiB. Its operating system is CentOS Linux 7 (Core). We implemented our proposed model HGTM using Python 3.6 as programming language and TensorFlow 1.15.0 as deep learning library. Other frameworks include NumPy 1.17.4, sklearn 0.23.2, and scipy 1.5.2.

[4]http://www.arnetminer.org/citation
[5]https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng
[6]https://snap.stanford.edu/data/memetracker9.html