

# Towards Effective, Efficient and Unsupervised Social Event Detection in the Hyperbolic Space

Xiaoyan Yu<sup>1</sup>, Yifan Wei<sup>2</sup>, Shuaishuai Zhou<sup>3</sup>, Zhiwei Yang<sup>4</sup>, Li Sun<sup>5</sup>, Hao Peng<sup>2</sup>, Liehuang Zhu<sup>1\*</sup>, Philip S. Yu<sup>6</sup>

<sup>1</sup>School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

<sup>2</sup>Beihang University, Beijing, China <sup>3</sup>Kunming University of Science and Technology, Kunming, China

<sup>4</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

<sup>5</sup>North China Electric Power University, Beijing, China <sup>6</sup>University of Illinois at Chicago, Chicago, USA  
 {xiaoyan.yu, liehuangz}@bit.edu.cn, {weiyifan, penghao}@buaa.edu.cn, psyu@uic.edu

## Abstract

The vast, complex, and dynamic nature of social message data has posed challenges to social event detection (SED). Despite considerable effort, these challenges persist, often resulting in inadequately expressive message representations (ineffective) and prolonged learning durations (inefficient). In response to the challenges, this work introduces an unsupervised framework, **HyperSED (Hyperbolic SED)**. Specifically, the proposed framework first models social messages into semantic-based message anchors, and then leverages the structure of the anchor graph and the expressiveness of the hyperbolic space to acquire structure- and geometry-aware anchor representations. Finally, HyperSED builds the partitioning tree of the anchor message graph by incorporating differentiable structural information as the reflection of the detected events. Extensive experiments on public datasets demonstrate HyperSED’s competitive performance, along with a substantial improvement in efficiency compared to the current state-of-the-art unsupervised paradigm. Statistically, HyperSED boosts incremental SED by an average of 2%, 2%, and 25% in NMI, AMI, and ARI, respectively; enhancing efficiency by up to 37.41 times and at least 12.10 times, illustrating the advancement of the proposed framework. Our code is publicly available at <https://github.com/XiaoyanWork/HyperSED>.

## Introduction

Social Event Detection (SED), a task aimed at identifying noteworthy occurrences on social media (Cordeiro and Gama 2016), remains challenging due to its *large scale*, *complex interrelations*, and *high dynamism* (Fedoryszak et al. 2019). The SED task is pivotal for various downstream applications, such as crisis management (Pohl, Bouchachia, and Hellwagner 2012), public opinion monitoring (Karamouzas, Mademlis, and Pitas 2022), etc. Current attempts frame SED as a problem of learning representations of social messages and clustering them into events (Cao et al. 2021; Ren et al. 2022).

Recent advancements in modeling social messages into graphs have yielded promising results across supervised

(Cao et al. 2021), self-supervised (Ren et al. 2022), and unsupervised (Cao et al. 2024) paradigms. These methods typically construct the social message graph based on message attributes (Peng et al. 2021) and employ graph neural networks (GNNs), optimized through contrastive learning (Cao et al. 2021), reinforcement learning (Peng et al. 2022), and pseudo-label generation (Ren et al. 2022) to learn clustering-friendly message representations. Notably, Cao et al. (2024) introduced an unsupervised algorithm based on structural entropy (Li and Pan 2016) minimization, achieving state-of-the-art performance without any supervision signal.

However, existing works have neglected some or all of the potential challenges posed by the characteristics of social message data. **Firstly, the large scale of social messages leads to resource-intensive learning.** Social messages are often short texts that users post in response to specific events. Among messages describing the same event, certain messages may exhibit high content similarity, even nearly identical semantics. When handling such data, current methods (Ren et al. 2022; Cao et al. 2021) consider all messages for learning, leading to resource wastage. **Secondly, the complex interrelations between messages** require a stereoscopic representation. To uncover the relationships among messages, current methods (Cao et al. 2024; Ren et al. 2023) learn message representations within the Euclidean space. However, in real-world scenarios, messages often exhibit nested and hierarchical patterns (Wang et al. 2024). When confronted with such data patterns, the Euclidean space lacks the expressiveness required to uncover profound interrelations among them. **Thirdly, the dynamic nature** of novel events continuously emerging necessitates SED systems to react efficiently. New events occur continuously, regularly introducing new topics, trends, and relationships, making it costly and laborious to obtain message labels or total event count. Efficiently detecting new events from a large-scale message pool without supervision is crucial for SED systems.

To address the aforementioned challenges, this work introduces a novel unsupervised SED framework, HyperSED (**Hyperbolic SED**). Given the vast amount of social messages, we establish the notion of the Semantic-based Anchor Message Graph (SAMG), where semantically related messages are seen as a singular node within the anchor graph.

\*Corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

To uncover the complex interrelations among messages, we represent the anchor and model the graph structure within the hyperbolic space to derive structure- and geometry-aware anchor representations. Finally, we harness the power of differentiable structural information to construct the partitioning tree of the SAMG bottom-up, which formulates the detected events efficiently and effectively. Extensive experiments on two publicly available datasets demonstrate HyperSED’s efficiency and effectiveness for SED. Further evaluation and analyses highlight HyperSED’s advancement and applicability as a solid framework. In summary, our contributions are as follows:

- We introduce HyperSED, an effective and efficient unsupervised framework for social event detection, which exploits the benefits of graph learning in the hyperbolic space and structural information to achieve event detection.
- We propose a semantic-based anchor message graph, simplifying the vast amount of social messages from a semantic viewpoint and uncovering the complex interrelations among messages within the hyperbolic space.
- Extensive experiments on public datasets demonstrate HyperSED’s competitive performance in unsupervised online and offline SED scenarios, marked by significant enhancements in efficiency.

## Related Work

This section includes the overview of literatures related to Social Event Detection (SED) and Structural Entropy (SE).

**Social Event Detection** Social Event Detection (SED) is a long-standing and challenging task (Atefah and Khreich 2015). The vast amount of social messages, covering a wide range of topics (Ren et al. 2023), poses significant difficulties in learning message representations and categorizing events. Earlier SED methods fall in incremental clustering (Zhao, Mitra, and Chen 2007; Feng et al. 2015), topic modeling (Xing et al. 2016; Wang et al. 2016), and community detection (Fedoryszak et al. 2019; Liu et al. 2020b,a). Recent SED methods utilize GNNs’ capabilities to model social messages and their relationships (Cao et al. 2021; Ren et al. 2022; Peng et al. 2022, 2021) by extracting key attributes. They employ contrastive learning to pull related messages closer and push unrelated messages apart, optimizing supervised (Cao et al. 2021) or self-supervised (Ren et al. 2022) SED. However, these methods require supervision signals like a predefined number of events. HISEvent, proposed by Cao et al. (2024), is an unsupervised SED algorithm that hierarchically minimizes two-dimensional structural entropy, requiring no supervision signals. Our work retains the merits of prior approaches and addresses their limitations, achieving effective and efficient unsupervised event detection.

**Structural Entropy** Information entropy (Shannon 1948) measures the amount of information in unstructured data but cannot capture information in graph structures. Li and Pan (2016) proposed structural entropy to consider the structural information of graphs. This theory has been applied in various fields, such as graph pooling (Wu et al. 2022), adversarial attacks (Liu et al. 2019), contrastive learning (Wu et al.

2023), and graph structural learning (Zou et al. 2023). In the context of SED, Cao et al. (2024) proposed minimizing two-dimensional structural entropy to achieve unsupervised event detection. However, these methods utilize the discrete algorithms of this theory and cannot be combined with trainable networks. This limitation was overcome when Sun et al. (2024) reformulated the structural entropy into differentiable structural information for deep graph clustering. This has allowed optimizing the partitioning tree proposed by Li and Pan (2016).

## Preliminaries

This section outlines the essential concepts and definitions, including Differentiable Structural Information and Hyperbolic Space, and then presents the problem formulation.

### Differentiable Structural Information

Differentiable Structural Information (DSI, (Sun et al. 2024)) is an equivalent reformulation of structural information (Li and Pan 2016). This reformulation employs a layer-wise assignment approach, enabling the partitioning tree to be differentiable and thus optimizable.

**Definition 1** (Partitioning Tree (Li and Pan 2016; Cao et al. 2024)). Given a weighted graph  $G = (\mathcal{V}, \mathcal{E})$  with weight function  $\omega$ , its partitioning tree  $\mathcal{T}$  satisfies the following:

1. For the root node  $\lambda$ ,  $T_\lambda = \mathcal{V}$ . For every node  $\alpha \in \mathcal{T}$ , a subset  $T_\alpha \subseteq \mathcal{V}$  is associated with it. Each leaf node  $\gamma$  is associated with a single node in  $G$ ,  $T_\gamma = \{v\}, v \in \mathcal{V}$ .
2. The height of tree  $\mathcal{T}$  is  $h(\mathcal{T}) = H$ ,  $h(\lambda) = H$ ,  $h(\gamma) = 0$ , and  $h(\alpha^-) = h(\alpha) + 1$ , where  $\alpha^-$  denote the parent of  $\alpha$ .
3. For each node  $\alpha \in \mathcal{T}$ , denote its children as  $\beta_1, \beta_2, \dots$ ; thus,  $(T_{\beta_1}, T_{\beta_2}, \dots)$  is a partition of  $T_\alpha$ .

**Definition 2** ( $H$ -Dimensional Structural Information (Li and Pan 2016; Sun et al. 2024)). For a partitioning tree  $\mathcal{T}$  with height  $H$ , the formula of structural information of  $G$  at the  $h$ -th layer of  $\mathcal{T}$  is defined as:

$$\begin{aligned} \mathcal{H}^{\mathcal{T}}(G; h) = & -\frac{1}{\text{vol}(G)} \sum_{k=1}^{N_h} (\text{vol}^h(T_k) - \sum_{(i,j) \in \mathcal{E}} S_{ik}^h S_{jk}^h \omega_{ij}) \\ & \cdot \log_2 \frac{\text{vol}^h(T_k)}{\text{vol}^{h-1}(T_{k^-})}, \end{aligned} \quad (1)$$

where  $\text{vol}(\cdot)$  calculate the volume of corresponding node set and  $\omega_{ij}$  is the weight between node  $i$  and  $j$ . For the  $k$ -th node in height  $h$ ,  $S_{ik}^h = \mathbb{I}(i \in T_k)$ . Finally, the  $H$ -dimensional structural information of  $G$  is:

$$\mathcal{H}^{\mathcal{T}}(G) = \sum_{h=1}^H \mathcal{H}^{\mathcal{T}}(G; h). \quad (2)$$

### Hyperbolic Space

The hyperbolic space, unlike the ‘flat’ Euclidean space, is a curved space with negative curvature  $\kappa$ , which quantifies the extent to which the manifold deviates from flatness. We apply the Poincaré ball model (Ungar 2001) of the hyperbolic space, which excels in social network analysis due

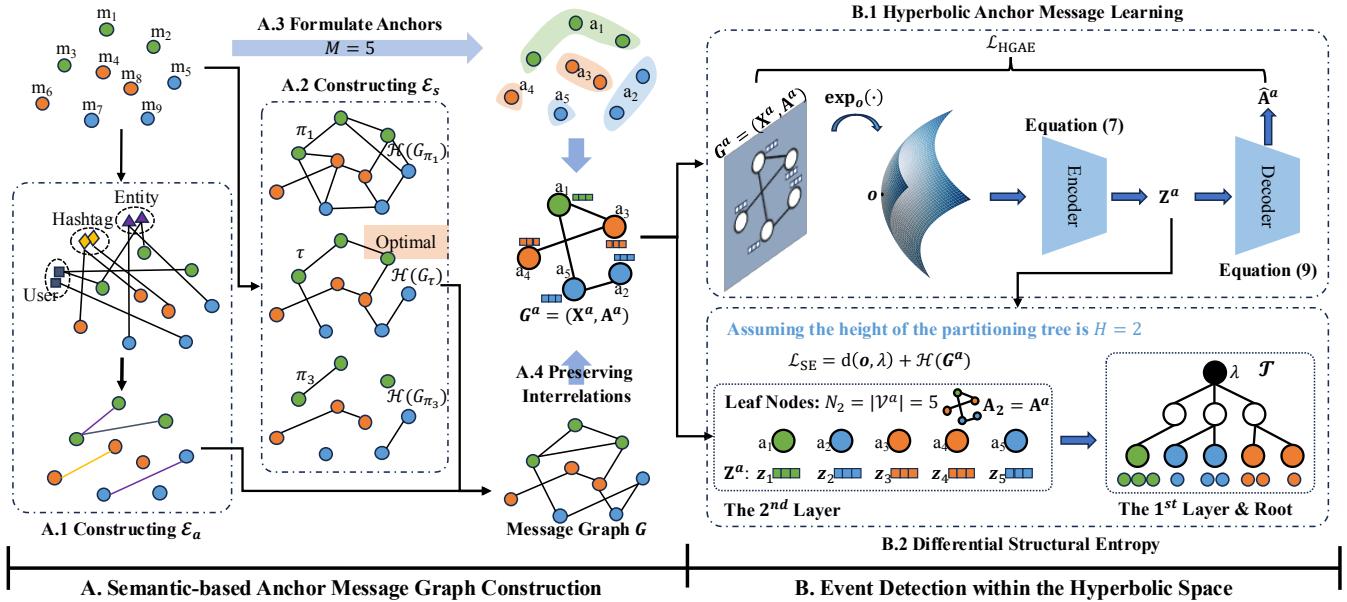


Figure 1: The overall framework of HyperSED.

to its efficient and compact representation of hierarchical and nested structures (Wang et al. 2024; Papadopoulos, Psomas, and Krioukov 2014). The  $d$ -dimensional Poincaré ball model with constant negative curvature  $\kappa < 0$  is defined as (Ungar 2022)  $\mathbb{B}_\kappa^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 = -\frac{1}{\kappa}\}$ , where  $\|\cdot\|$  denotes Euclidean norm. The origin of the Poincaré ball is  $\mathbf{o} = (0, \dots, 0) \in \mathbb{R}^d$ . The distance function (Ungar 2022) is given by  $d_{\mathbb{B}_\kappa}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{-\kappa}} \tanh^{-1}(\|\mathbf{u}\|)$ , where  $\mathbf{u} = (-\mathbf{x}) \oplus_\kappa \mathbf{y}$  and  $\oplus_\kappa$  is the Möbius addition (Ungar 2001):

$$\mathbf{x} \oplus_\kappa \mathbf{y} = \frac{(1 - 2\kappa \mathbf{x}^\top \mathbf{y} - \kappa \|\mathbf{y}\|^2)\mathbf{x} + (1 + \kappa \|\mathbf{x}\|^2)\mathbf{y}}{1 - 2\kappa \mathbf{x}^\top \mathbf{y} + \kappa^2 \|\mathbf{x}\|^2 \|\mathbf{y}\|^2}. \quad (3)$$

The exponential and logarithmic maps of Poincaré ball model are defined as (Petersen 2006):

$$\exp_\mathbf{x}^\kappa(\mathbf{v}) = \mathbf{x} \oplus_\kappa \left( \frac{\mathbf{v}}{\sqrt{-\kappa} \|\mathbf{v}\|} \tanh\left(\frac{\sqrt{-\kappa} \lambda_x^\kappa \|\mathbf{v}\|}{2}\right) \right), \quad (4)$$

$$\log_\mathbf{x}^\kappa(\mathbf{y}) = \frac{2}{\sqrt{-\kappa} \lambda_x^\kappa} \tanh^{-1}\left(\frac{\sqrt{-\kappa} \|\mathbf{u}\|}{2}\right) \frac{\mathbf{u}}{\|\mathbf{u}\|}. \quad (5)$$

For more details about the hyperbolic space, please refer to the Appendix.

## Problem Formulation

We technically approach the task of Social Event Detection (SED) as a graph learning and clustering problem and outline the formulation as follows:

**Input.** A sequence of social messages  $m_1, m_2, \dots$ , and  $m_N$ , where  $N$  represents the total number of messages.

**Output.**  $K$  clusters, each corresponding to a detected event.

**Goal.** The objective is to construct social message graphs and uncover the correlation between messages to derive

an optimal partitioning tree  $\mathcal{T}$  without a predetermined number of event clusters. The obtained partitioning tree reflects the detected events.

## Methodology

The proposed framework, HyperSED, as shown in Figure 1, comprises two stages: anchor message graph construction and event detection.

### Anchor Message Graph Construction

We begin by constructing social message graphs from individual messages. Subsequently, an anchor message graph is formulated based on the semantic content of the messages while retaining message interrelations.

**Message Graph Construction.** In the message graph  $G = (\mathcal{V}, \mathcal{E}_a \cup \mathcal{E}_s)$ , we treat all messages as nodes  $\mathcal{V} = \{m_1, \dots, m_N\}$ , and consider two types of edges: attribute edges  $\mathcal{E}_a$  and semantic edges  $\mathcal{E}_s$ . Attribute edges convey the co-occurrence of attributes among social messages, such as user  $u$ , hashtag  $h$ , and entity  $e$  (Ren et al. 2022). Semantic edges serve to compensate for potential correlations that may be overlooked due to the absence of co-occurring attributes. For each message  $m_i$ , we extract its attributes  $a_i = \{u_i, h_i, e_i\}$ . An attribute edge is added in  $G$  between  $m_i$  and  $m_j$  if they share a common attribute, i.e.,  $\mathcal{E}_a = \{(m_i, m_j) \mid a_i \cap a_j \neq \emptyset\}$ , as shown in Figure 1A.1. For semantic edges, Cao et al. (2024) identify  $k$  stable neighbors for each node via minimizing the one-dimensional structural entropy (1DSE). However, this practice incurs significant computational costs and may not always yield results. Therefore, we design an efficient algorithm that determines an optimal similarity threshold  $\tau$ , preserving edges with weights surpassing it, as shown in Figure 1A.2. First,

all message nodes are interconnected by edge weights corresponding to their similarity. The similarity is calculated with cosine similarity between their initial message embeddings (in this work obtained with SBERT (Reimers and Gurevych 2019)). Then, the algorithm computes the 1DSE of graphs across varying thresholds through:

$$\mathcal{H}^{(1)}(G) = - \sum_{i=1}^{|V|} \frac{d_i}{\text{vol}(G)} \log \frac{d_i}{\text{vol}(G)}. \quad (6)$$

The optimal threshold is determined by searches for a relatively stable graph structure and preserving more edges as:

$$\tau = \arg \min_{\pi \in \Pi} \left| \mathcal{H}^{(1)}(G_\pi) - \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \mathcal{H}^{(1)}(G_\pi) \right|, \quad (7)$$

where  $\Pi$  is the search space of thresholds and  $\mathcal{H}^{(1)}(G_\pi)$  is the 1DSE of  $G$  under the threshold  $\pi$ . The semantic edges are defined as  $\mathcal{E}_s = \{(m_i, m_j) | \text{sim}(m_i, m_j) \geq \tau\}$ . So far, all edges have been established, resulting in the creation of the message graph  $G = (\mathcal{V}, \mathcal{E}_a \cup \mathcal{E}_s) = (\mathbf{X}, \mathbf{A})$ , where  $\mathbf{X}$  is the embedding matrix and  $\mathbf{A}$  denotes the adjacency matrix with  $A_{ij} = \max(\text{sim}(\mathbf{x}_i, \mathbf{x}_j), 0)$ , s.t.  $(m_i, m_j) \in \mathcal{E}_a \cup \mathcal{E}_s$ .

**Anchor Message Graph Construction.** With a substantial size of social messages covering diverse topics, many exhibit semantic similarities or near-identical content. Current methods for SED generally consider all messages within the message graph (Cao et al. 2024; Ren et al. 2023), which makes the graph structure complex and the learning process resource-intensive. This work addresses this by constructing a Semantic-based Anchor Message Graph (**SAMG**), which regards semantically similar messages as a single anchor node while preserving their relationships through edges. The essence of constructing the anchor graph is to reduce the size of the message graph while preserving the integrity of the information conveyed by the original messages. To achieve this goal, we derive  $M$  anchor nodes, denoted as  $\{a_1, \dots, a_M\}$ , based on the semantic similarities between message embeddings  $\mathbf{X}$ . Each anchor comprises a set of semantically related messages, illustrated in Figure 1A.3. The embedding  $\mathbf{x}_u^a$  of the anchor node  $a_u$  is computed as the arithmetic mean of the message representations within the anchor, given by  $\mathbf{x}_u^a = \frac{1}{|a_u|} \sum \{\mathbf{x}_i | m_i \in a_u\}$ . The adjacency matrix  $\mathbf{A}^a$  of the anchor graph is defined as  $\mathbf{A}^a = (\mathbf{C})^\top \mathbf{A} \mathbf{C}$ , where  $\mathbf{C} \in \mathbb{R}^{N \times M}$  and  $C_{iu} = \mathbb{I}(m_i \in a_u)$ . It signifies that the edge weight in the anchor graph preserves the connections between messages not assigned to the same anchor while accounting for the interrelations of messages from an anchor perspective. Thus, the SAMG  $G_a = (\mathbf{X}^a, \mathbf{A}^a)$  is obtained, with each node serving as a representative of a group of semantically related messages and edges as the correlations between the messages of different anchors.

**Remark.** SBERT effectively captures message semantics. By applying clustering techniques to these representations, the resulting anchors are believed to group semantically similar messages effectively.

## Event Detection within the Hyperbolic Space

Given the SAMG, we take a three-step effort to achieve unsupervised social event detection, as shown in Figure 1B. The primary objective is to capture intricate relationships among anchors and obtain the optimal partitioning tree of the graph.

**Step 1: Learning the SAMG with graph autoencoder in the hyperbolic space (**HGAE**).** This step aims to acquire structure- and geometry-aware anchor representations by learning the anchor graph in the hyperbolic space. The structure-aware representation captures the relational dependencies and patterns among anchors. The geometry-aware representation reflects the spatial and topological correlations among anchors within the hyperbolic space, indicating their stereoscopic distances. To achieve this, we first define the hyperbolic graph autoencoder (**HGAE**). Inspired by prior works (Kipf and Welling 2016; Chami et al. 2019; Park et al. 2021; Sun et al. 2024), we first map the Euclidean anchor  $(a_i)$  embedding  $\mathbf{x}_i^a \in \mathbb{R}^d$  into the hyperbolic space by  $\mathbf{h}_i = \exp_o^\kappa(\mathbf{x}_i^a) \in \mathbb{B}_\kappa^d$  (Equation 4), where  $d$  is the embedding's dimension. The graph encoder in the hyperbolic space, which is basically a hyperbolic convolutional layer (Chami et al. 2019), is defined as:

$$\mathbf{z}_i = \text{PConv}(\mathbf{h}_i) = \exp_o^\kappa \left( \sum_{j=1}^N \omega_{ij} (\mathbf{W} \log_o^\kappa(\mathbf{h}_i) + b) \right), \quad (8)$$

where  $\mathbf{z}_i$  is the latent representation of the anchor  $a_i$  and  $\omega_{ij}$  is derived from the attention mechanism (Vaswani et al. 2017):

$$\omega_{ij} = \frac{\exp \left( -\frac{1}{\sqrt{|M|}} d_{\mathbb{B}}^2(\mathbf{h}_i, \mathbf{h}_j) \right)}{\sum_{j=1}^M \exp \left( -\frac{1}{\sqrt{|M|}} d_{\mathbb{B}}^2(\mathbf{h}_i, \mathbf{h}_j) \right)}. \quad (9)$$

The decoder, which reconstructs the adjacency matrix, defines the reconstruction in the hyperbolic space using the Fermi-Dirac distribution (Krioukov et al. 2010; Park et al. 2021):

$$\hat{A}_{ij} = \sigma \left( \exp \left( \frac{d_{\mathbb{B}}^2(\mathbf{z}_i, \mathbf{z}_j) - q}{t} \right) + 1 \right)^{-1}, \quad (10)$$

where  $\hat{\mathbf{A}}$  is the reconstructed anchor adjacency matrix,  $\sigma$  is the activation function, and  $q$  and  $t$  are hyperparameters. Finally, the reconstruction loss of the affinity matrix is (Kipf and Welling 2016):

$$\mathcal{L}_{\text{HGAE}} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{H}, \mathbf{A}^a)} [\log p(\hat{\mathbf{A}}^a | \mathbf{Z})]. \quad (11)$$

**Step 2: Bottom-up Learning of Tree Node Embeddings.**

This step seeks to build a partitioning tree of the SAMG. Following Definition 1, where each graph node corresponds to a leaf node in the partitioning tree, the partitioning tree can be built by iteratively assigning parent nodes layer-by-layer, starting from the leaf nodes (Sun et al. 2024). DSI designs an assignment scheme that assigns the  $i$ -th node on the  $h$ -th layer to the  $j$ -th node on the  $(h-1)$ -th layer by:

$$\mathbf{C}^h = \text{Softmax} \left( \mathbf{A}^h \sigma \left( \text{PConv}(\mathbf{Z}_a^h) \right) \right) \in \mathbb{R}^{N_h \times N_{h-1}}, \quad (12)$$

where  $N_h$  and  $N_{h-1}$  are the number of nodes on the  $h$ -th and the  $(h-1)$ -th layer, respectively. Softmax is for row normalization. Upon establishing correspondence, the embedding and adjacency matrix of the  $(h-1)$ -th layer is required, as the construction of the  $(h-2)$ -th layer relies on this information (as implied by Equation 12). Similar to building the anchor graph, the embedding of the nodes on the  $(h-1)$ -th layer is defined with the arithmetic mean of all the embeddings of nodes assigned to it. Only in this case, in the hyperbolic space, the arithmetic mean can be seen as finding the point in the hyperbolic space that is the closest to all nodes. This process can be formalized as the Fréchet mean (Lou et al. 2020):

$$\mathbf{z}_j^{h-1} = \arg \min_{\mathbf{z}_j^{h-1}} \sum_{i=1}^{N^h} c_{ij} d_{\mathbb{B}}^2(\mathbf{z}_j^{h-1}, \mathbf{z}_i^h). \quad (13)$$

The adjacency matrix of nodes on the  $(h-1)$ -th layer can be derived from it on the  $h$ -th layer, that is  $\mathbf{A}^{h-1} = (\mathbf{C}^h)^\top \hat{\mathbf{A}}^h \mathbf{C}^h$ . We recursively utilize Equation 12 and Equation 13 to build the partitioning tree from leaf nodes to the root node in hyperbolic space.

**Step 3: Optimization of the partitioning tree.** In essence, the optimization objective encompasses two primary aspects: first, find the proper position to put the root node of the partitioning tree, and second, minimize the structural entropy of the SAMG. For the first aspect, a straightforward approach is to position the root node at the origin  $\mathbf{o}$  of the Poincaré ball. This choice is significant as the origin serves as a reference point for mapping from Euclidean to hyperbolic space, where all other points exhibit symmetric positions relative to it. Regarding the second aspect, which involves minimizing the H-dimensional structural entropy corresponding to Equation 2. The loss function is given by:

$$\mathcal{L}_{SE} = d_{\mathbb{B}}(\mathbf{o}, \mathbf{z}^0) + \sum_{h=1}^H H^T(G; h), \quad (14)$$

where  $\mathbf{z}^0$  is the embedding of the root node. The total loss of the training is  $\mathcal{L} = \mathcal{L}_{HGAE} + \mathcal{L}_{SE}$ .

**Remark.** The overall methodology can be seen as starting from individual messages, layer-by-layer, to the top of the partitioning tree. Only the immediate parent layer of messages is chosen based on their semantic relatedness rather than the layer-wise assignment and the value of structural entropy. This strategy maintains the benefits of SE while notably enhancing efficiency through learning with anchors.

## Time Complexity of HyperSED

First, the time complexity for the message graph construction including construct attribute edges:  $O(|\mathcal{E}_a|)$ , calculating the similarity between messages:  $O(N^2)$ , and constructing the semantic edges:  $O(|\Pi| \cdot |\mathcal{E}_s|)$ . Second, constructing the SAMG has the time complexity of  $O(N)$ . Third, the DSI is computed level-wisely, so the time complexity at level  $h$  is  $O(N_h|\mathcal{E}|)$ . Fourth, the time complexity for constructing the partitioning tree is  $O(HM)$ . Therefore, the overall time complexity comes to  $O(|\mathcal{E}_a| + N^2 + N + M|\mathcal{E}|)$ , as  $N_h \ll M, H$  denote the height of  $\mathcal{T}$ , and  $|\Pi|$  is the search size of threshold, which are small constant numbers.

## Experiments

We conduct experiments and analyses to evaluate the **effectiveness** and **efficiency** of the proposed framework. Additionally, we present *ablation studies* and *parameter analyses* to demonstrate the advancement of the proposed framework.

Table 1: Offline results. The best result is marked in **bold**, the runner-up is marked in *italic*, ‘\*’ marks results acquired with the ground truth event numbers, the proposed method is marked in **orange** background.

Datasets	English Twitter			French Twitter		
	NMI	AMI	ARI	NMI	AMI	ARI
BERT*	0.42	0.11	0.01	0.26	0.10	0.01
SBERT*	<i>0.83</i>	<i>0.73</i>	0.17	<i>0.66</i>	<i>0.59</i>	0.10
EventX	0.72	0.19	0.05	0.56	0.16	0.03
KPGNN*	0.79	0.52	<i>0.22</i>	0.56	0.44	0.15
QSGNN*	0.72	0.53	<i>0.22</i>	0.58	0.44	0.16
HISEvent	0.47	0.42	0.17	0.48	0.41	<i>0.30</i>
HyperSED	<b>0.84</b>	<b>0.77</b>	<b>0.44</b>	<b>0.68</b>	<b>0.62</b>	<b>0.62</b>
promotion	↑ 0.01	↑ 0.04	↑ 0.22	↑ 0.02	↑ 0.03	↑ 0.32

## Experimental Setups

**Datasets.** We conduct experiments on two publicly available datasets: the English Twitter dataset (68,841 messages, 503 events) (McMinn, Moshfeghi, and Jose 2013) and the French Twitter dataset (64,516 messages, 257 events) (Mazoyer et al. 2020). Data splitting for offline and online evaluation consistent with prior studies (Cao et al. 2021, 2024). Specifically, for offline evaluation, the data is divided into training, validation, and test sets with a ratio of 7:1:2; for online evaluation, the data is segmented into message blocks based on time intervals: the first block encompassed messages from the first seven days (weekly), followed by daily blocks. Detailed statistics regarding the message blocks are shown in Table 4.

**Baselines.** We compare HyperSED with two types of SED baselines. (1) Non-Graph-based methods: **BERT** (Devlin et al. 2019) and **SBERT** (Reimers and Gurevych 2019), which are Transformer-based pre-trained language models (PLMs) known for their strong text representation capabilities; **EventX** (Liu et al. 2020a) leverages community detection for SED. (2) Graph-based methods: **KPGNN** (Cao et al. 2021) construct homogeneous graphs and utilize Graph Attention Networks (GATs) for supervised SED. **QSGNN** (Ren et al. 2022) employs pseudo-labels and active learning for self-supervised SED. **HISEvent** (Cao et al. 2024) achieves unsupervised partitioning through hierarchical minimization of two-dimensional structural entropy. We set the hyperparameter  $n$  to 200 to mitigate occasional deadlock issues. All baselines, except EventX and HISEvent, require predefined event numbers. For implementation details, please refer to the Appendix.

**Evaluation Metrics.** In line with previous SED studies (Cao et al. 2024; Ren et al. 2022), we assess the performance of all methods using three common clustering metrics: Normalized Mutual Information (**NMI**) (Estévez et al. 2009),

Table 2: Online results of Twitter English. The best result is marked in **bold**, the runner-up is marked in *italic*, '\*' marks results acquired with the ground truth event numbers, the proposed method is marked in orange background.

Blocks	M <sub>1</sub>			M <sub>2</sub>			M <sub>3</sub>			M <sub>4</sub>			M <sub>5</sub>			M <sub>6</sub>			M <sub>7</sub>		
Metrics	NMI	AMI	ARI																		
BERT*	0.14	0.11	0.01	0.38	0.32	0.43	0.22	0.15	0.05	0.26	0.18	0.03	0.28	0.23	0.08	0.32	0.20	0.08	0.20	0.14	0.01
SBERT*	0.40	0.38	0.03	0.85	0.84	0.70	0.89	0.88	0.67	0.80	0.78	0.36	0.87	0.87	0.79	0.86	0.83	0.56	0.63	0.60	0.09
EventX	0.36	0.06	0.01	0.68	0.29	0.45	0.63	0.18	0.09	0.63	0.19	0.07	0.59	0.14	0.04	0.70	0.27	0.14	0.51	0.13	0.02
KPGNN*	0.39	0.37	0.07	0.79	0.78	0.76	0.76	0.74	0.58	0.67	0.64	0.29	0.73	0.71	0.47	0.82	0.79	0.72	0.55	0.51	0.12
QSGNN*	0.43	0.41	0.07	0.81	0.80	0.77	0.78	0.76	0.59	0.71	0.68	0.29	0.75	0.73	0.48	0.83	0.80	0.73	0.57	0.54	0.12
HISEvent	0.38	0.37	0.09	0.90	0.89	0.88	0.90	0.89	0.79	0.77	0.76	0.52	0.83	0.82	0.63	0.89	0.88	0.84	0.64	0.63	0.36
HyperSED	<b>0.84</b>	<b>0.84</b>	<b>0.96</b>	0.84	0.83	0.77	0.85	0.84	<b>0.81</b>	0.77	0.75	<b>0.60</b>	0.84	0.82	0.71	0.86	0.83	0.76	<b>0.85</b>	<b>0.85</b>	<b>0.89</b>
promotion	↑0.41	↑0.43	↑0.87	↓0.06	↓0.06	↓0.11	↓0.05	↓0.05	↑0.02	↓0.03	↓0.03	↑0.08	↓0.03	↓0.05	↓0.08	↓0.03	↓0.05	↓0.08	↑0.21	↑0.22	↑0.53
Blocks	M <sub>8</sub>			M <sub>9</sub>			M <sub>10</sub>			M <sub>11</sub>			M <sub>12</sub>			M <sub>13</sub>			M <sub>14</sub>		
Metrics	NMI	AMI	ARI																		
BERT*	0.29	0.15	0.04	0.30	0.20	0.09	0.30	0.20	0.07	0.29	0.20	0.06	0.20	0.15	0.04	0.25	0.18	0.04	0.22	0.16	0.05
SBERT*	<b>0.87</b>	<b>0.85</b>	0.62	0.86	0.84	0.47	0.88	0.86	0.61	0.81	0.79	0.37	0.84	0.83	0.60	0.72	0.69	0.21	0.77	0.75	0.36
EventX	0.71	0.21	0.09	0.67	0.19	0.07	0.68	0.24	0.13	0.65	0.24	0.16	0.61	0.16	0.07	0.58	0.16	0.04	0.57	0.14	0.10
KPGNN*	0.80	0.76	0.60	0.74	0.71	0.46	0.80	0.78	0.70	0.74	0.71	0.49	0.68	0.66	0.48	0.69	0.67	0.29	0.69	0.65	0.42
QSGNN*	0.79	0.75	0.59	0.77	0.75	0.47	0.82	0.80	0.71	0.75	0.72	0.49	0.70	0.68	0.49	0.68	0.66	0.29	0.68	0.66	0.41
HISEvent	0.82	0.81	0.68	<b>0.89</b>	<b>0.88</b>	0.65	<b>0.91</b>	<b>0.90</b>	<b>0.87</b>	0.85	0.84	0.66	<b>0.87</b>	<b>0.87</b>	0.82	0.75	0.74	0.39	0.83	0.82	0.71
HyperSED	<b>0.87</b>	<b>0.85</b>	<b>0.69</b>	0.86	0.84	<b>0.81</b>	0.86	0.85	0.74	<b>0.86</b>	<b>0.84</b>	<b>0.91</b>	0.78	0.76	0.62	<b>0.82</b>	<b>0.81</b>	<b>0.93</b>	<b>0.84</b>	<b>0.83</b>	<b>0.80</b>
promotion	0.00	0.00	↑0.01	↓0.03	↓0.04	↑0.16	↓0.05	↓0.05	↓0.13	↑0.01	0.00	↑0.25	↓0.09	↓0.11	↓0.20	↑0.07	↑0.07	↑0.54	↑0.01	↑0.01	↑0.09
Blocks	M <sub>15</sub>			M <sub>16</sub>			M <sub>17</sub>			M <sub>18</sub>			M <sub>19</sub>			M <sub>20</sub>			M <sub>21</sub>		
Metrics	NMI	AMI	ARI																		
BERT*	0.24	0.17	0.03	0.28	0.19	0.07	0.22	0.16	0.04	0.20	0.14	0.04	0.28	0.22	0.08	0.32	0.19	0.07	0.20	0.15	0.03
SBERT*	0.71	0.68	0.17	0.80	0.77	0.50	0.79	0.78	0.42	<b>0.81</b>	<b>0.80</b>	0.52	<b>0.86</b>	<b>0.85</b>	0.56	<b>0.84</b>	<b>0.80</b>	0.54	0.72	0.70	0.26
EventX	0.49	0.07	0.01	0.62	0.19	0.08	0.58	0.18	0.12	0.59	0.16	0.08	0.60	0.16	0.07	0.67	0.18	0.11	0.53	0.10	0.09
KPGNN*	0.58	0.54	0.17	0.79	0.77	0.66	0.70	0.68	0.43	0.68	0.66	0.47	0.73	0.71	0.51	0.72	0.68	0.51	0.60	0.57	0.20
QSGNN*	0.59	0.55	0.17	0.78	0.76	0.65	0.71	0.69	0.44	0.70	0.68	0.48	0.73	0.70	0.50	0.73	0.69	0.51	0.61	0.58	0.21
HISEvent	0.69	0.67	0.27	0.87	0.86	0.83	0.77	0.76	0.56	0.74	0.73	0.64	0.85	0.84	0.60	0.82	0.80	0.67	0.73	0.73	0.46
HyperSED	<b>0.81</b>	<b>0.80</b>	<b>0.93</b>	<b>0.90</b>	<b>0.89</b>	<b>0.88</b>	<b>0.83</b>	<b>0.82</b>	<b>0.89</b>	<b>0.80</b>	<b>0.79</b>	<b>0.71</b>	<b>0.86</b>	<b>0.85</b>	<b>0.84</b>	<b>0.83</b>	<b>0.80</b>	<b>0.66</b>	<b>0.80</b>	<b>0.78</b>	<b>0.86</b>
promotion	↑0.10	↑0.12	↑0.66	↑0.03	↑0.03	↑0.05	↑0.04	↑0.04	↑0.33	↓0.01	↓0.01	↑0.07	0.00	0.00	↑0.24	↓0.01	0.00	↓0.01	↑0.07	↑0.05	↑0.40

Table 3: Online results of Twitter French. The best result is marked in **bold**, the runner-up is marked in *italic*, '\*' marks results acquired with the ground truth event numbers, the proposed method is marked in orange background.

Blocks	M <sub>1</sub>			M <sub>2</sub>			M <sub>3</sub>			M <sub>4</sub>			M <sub>5</sub>			M <sub>6</sub>			M <sub>7</sub>			M <sub>8</sub>		
Metrics	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI			
BERT*	0.13	0.11	0.04	0.11	0.09	0.04	0.07	0.05	0.01	0.07	0.05	0.01	0.12	0.08	0.02	0.09	0.06	0.01	0.14	0.12	0.05			
SBERT*	0.59	0.58	0.21	0.64	0.63	0.32	0.64	0.64	0.32	0.58	0.57	0.21	<b>0.74</b>	<b>0.73</b>	0.41	0.70	0.69	0.36	<b>0.66</b>	0.29	0.76	0.75	0.48	
EventX	0.34	0.11	0.02	0.37	0.12	0.02	0.39	0.01	0.11	0.39	0.14	0.06	0.53	0.24	0.13	0.44	0.15	0.08	0.41	0.12	0.02	0.54	0.21	0.09
KPGNN*	0.54	0.54	0.17	0.56	0.55	0.18	0.55	0.15	0.55	0.55	0.55	0.17	0.58	0.57	0.21	0.59	0.57	0.21	0.63	0.61	0.30	0.58	0.57	0.20
QSGNN*	0.57	0.56	0.18	0.58	0.57	0.19	0.58	0.17	0.56	0.58	0.57	0.18	0.61	0.59	0.23	0.60	0.59	0.21	0.64	0.63	0.30	0.57	0.55	0.19
HISEvent	0.74	0.74	0.58	0.73	0.73	0.60	<b>0.72</b>	<b>0.72</b>	0.52	0.67	0.66	0.48	<b>0.74</b>	<b>0.73</b>	0.56	<b>0.80</b>	<b>0.79</b>	0.59	<b>0.82</b>	<b>0.82</b>	0.75	<b>0.75</b>	0.69	0.38
HyperSED	<b>0.77</b>	<b>0.77</b>	<b>0.89</b>	<b>0.75</b>	<b>0.75</b>	<b>0.77</b>	<b>0.71</b>	<b>0.81</b>	<b>0.70</b>	<b>0.70</b>	<b>0.59</b>	<b>0.68</b>	<b>0.65</b>	<b>0.57</b>	<b>0.78</b>	<b>0.77</b>	<b>0.86</b>	<b>0.67</b>	<b>0.66</b>	<b>0.75</b>	<b>0.79</b>	<b>0.78</b>	<b>0.64</b>	
promotion	↑0.03	↑0.03	↑0.31	↑0.02	↑0.02	↑0.17	↓0.01	↓0.01	↑0.25	↑0.03	↑0.04	↑0.11	↓0.06	↓0.08	↑0.01	↓0.02	↓0.02	↓0.12	↓0.12	↓0.16	↓0.03	↓0.04	↑0.50	
Blocks	M <sub>9</sub>			M <sub>10</sub>			M <sub>11</sub>			M <sub>12</sub>			M <sub>13</sub>			M <sub>14</sub>			M <sub>15</sub>			M <sub>16</sub>		
Metrics	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI	ARI			
BERT*	0.10	0.07	0.01	0.15	0.10	0.04	0.13	0.09	0.03	0.14	0.11	0.05	0.10	0.08	0.02	0.15	0.12	0.05	0.10	0.06	0.02	0.11	0.07	0.05
SBERT*	0.63	<b>0.62</b>	0.22	0.72	0.70	0.36	<b>0.72</b>	0.70	0.31	0.76	0.75	0.53	0.63	0.62	0.32	0.70	0.69	0.36	<b>0.66</b>	0.29	0.76	0.75	0.48	
EventX	0.45	0.16	0.07	0.52	0.19	0.07	0.48	0.18	0.06	0.51	0.20	0.09	0.44	0.15	0.06	0.52	0.22	0.11	0.49	0.22	0.11	0.39	0.10	0.01
KPGNN*	0.48	0.46	0.10	0.57	0.56	0.18	0.54	0.53	0.16	0.55	0.56	0.17	0.60	0.60	0.28	0.66	0.65	0.43	0.60	0.58	0.25	0.52	0.50	0.13
QSGNN*	0.52	0.46	0.13	0.60	0.58	0.19	0.60	0.59	0.20	0.61	0.59	0.20	0.59	0.58	0.27	0.68	0.67</							

Table 4: Graph construction time, running time, and overall time of unsupervised SED methods on French Twitter.

Blocks	Offline	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	M <sub>6</sub>	M <sub>7</sub>	M <sub>8</sub>	M <sub>9</sub>	M <sub>10</sub>	M <sub>11</sub>	M <sub>12</sub>	M <sub>13</sub>	M <sub>14</sub>	M <sub>15</sub>	M <sub>16</sub>	
Statistics	# Messages	12,902	5,356	3,186	2,644	3,179	2,662	4,200	3,454	2,257	3,669	2,385	2,802	2,927	4,884	3,065	2,411	1,107
	# Events	241	22	19	15	19	27	26	23	25	31	32	31	29	28	26	25	14
Construction Time (s)	HISEvent	1,250	533	299	243	323	258	437	334	214	371	229	281	281	506	298	241	95
	HyperSED	896	171	60	40	61	44	108	68	30	78	33	46	52	139	59	38	7
Running Time (s)	HISEvent	25,372	6,800	1,129	897	1,727	556	2,358	1,713	422	1,444	492	1,130	628	3,144	1,001	662	218
	HyperSED	65	25	58	11	12	17	17	16	17	21	16	24	17	25	64	11	6
Overall Time (s)	HISEvent	26,622	7,333	1,428	1,140	2,050	814	2,795	2,047	636	1,815	721	1,411	909	3,650	1,299	903	313
	HyperSED	961	196	118	51	73	61	125	84	47	99	49	70	69	164	123	49	13
Speed-up (times ×)		27.70	37.41	12.10	22.35	28.08	13.34	22.36	24.37	13.53	18.33	14.71	20.16	13.17	22.26	10.56	18.43	24.08

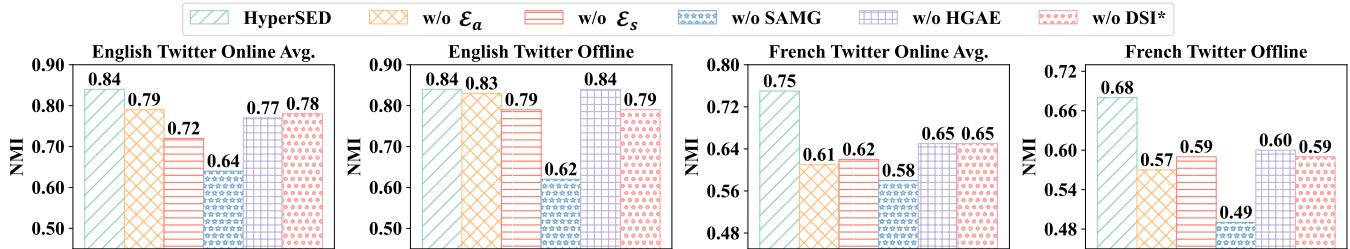


Figure 2: Ablation results for English and French Twitter. Online results are the average results across all blocks.

and the proposed HyperSED on the French Twitter dataset.

**HyperSED:** HyperSED exhibits remarkably short running times, even with the largest amount of messages (offline setting), which takes up only 65 seconds, whereas HISEvent takes 25,372 seconds. Compared to its running time, its graph construction time is longer, especially in the offline setting; however, consider the data scale, is all reasonable.

**Comparison:** In contrast, both the graph construction time and running time of HISEvent significantly lag behind HyperSED across all blocks. Overall, HyperSED demonstrates an average speedup (online setting) of 21.04 times in total time over HISEvent, with a maximum improvement of 37.41 times. Efficiency is paramount for SED systems as they must promptly respond to new messages and events. In this context, the proposed HyperSED outperforms existing unsupervised SED methods to a significant extent.

### Ablation Study

The results of the ablation study are presented in Figure 2. With the absence of DSI, the method requires the total event number as the supervision signal.

**Maximum Influence:** The largest performance drop is observed in the absence of SAMG. This is because, in scenarios of large-scale messages, the model without SAMG needs to learn all messages and their relationships, which is susceptible to noise interference. SAMG not only reduces the node size but also simplifies their relationships, which mitigates the influences caused by noise to a great extent.

**Minimum Influences:** The least performance drop is observed when any graph edge  $\mathcal{E}_a$  or  $\mathcal{E}_s$  is absent. This is due to SAMG solely preserving edges between messages not belonging to the same anchor. In most cases, messages sharing high similarity or attribute co-occurrence are housed within the same anchor, thereby limiting its impact. Nonetheless, the construction of these edges remains indispensable to the overall performance of HyperSED.

### Parameter Sensitivity Analysis

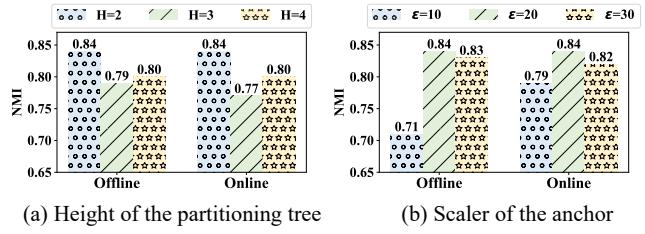


Figure 3: Results of parameter sensitivity analysis for English Twitter. Online results are the average results across all blocks.

We analyze the parameter sensitivity concerning the dimension of structural entropy (height of the partitioning tree)  $H = \{2, 3, 4\}$  and the number of anchor nodes  $M = N/\epsilon, \epsilon = \{10, 20, 30\}$ . The results on the English Twitter dataset are shown in Figure 3. The different parameters show minor fluctuations, all of which demonstrate reasonable performance without significant deviations.

### Conclusion

This work proposes HyperSED, an unsupervised SED framework that models social messages as anchors and learns their interrelations within the hyperbolic space. HyperSED learned semantically representative anchor nodes and simplified the relations among messages by incorporating SAMG. A substantial reduction in the detection runtime is achieved through the collaboration of SAMG and learning in the hyperbolic space. Experiments on public datasets demonstrate the effectiveness and efficiency of HyperSED, achieving competitive performance and enhancing efficiency significantly.

## Acknowledgments

We are sincerely grateful to all the reviewers and chairs for dedicating their valuable time and providing insightful suggestions. This work is supported by the Yunnan Provincial Major Science and Technology Special Plan Projects (No.202302AD080003). Hao Peng is supported by NSFC through grant 62322202.

## References

- Atefah, F.; and Khreich, W. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1): 132–164.
- Bachmann, G.; Bécigneul, G.; and Ganea, O. 2020. Constant curvature graph convolutional networks. In *International conference on machine learning*, 486–496. PMLR.
- Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; and Yu, P. S. 2021. Knowledge-preserving incremental social event detection via heterogeneous gnns. In *Proceedings of the Web Conference 2021*, 3383–3395.
- Cao, Y.; Peng, H.; Yu, Z.; and Yu, P. S. 2024. Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8255–8264.
- Chami, I.; Ying, Z.; Ré, C.; and Leskovec, J. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32: 1–20.
- Cordeiro, M.; and Gama, J. 2016. Online social networks event detection: a survey. *Solving Large Scale Learning Tasks. Challenges and Algorithms: Essays Dedicated to Katharina Morik on the Occasion of Her 60th Birthday*, 1–41.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.
- Estévez, P. A.; Tesmer, M.; Perez, C. A.; and Zurada, J. M. 2009. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2): 189–201.
- Fedoryszak, M.; Frederick, B.; Rajaram, V.; and Zhong, C. 2019. Real-time event detection on social data streams. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2774–2782.
- Feng, W.; Zhang, C.; Zhang, W.; Han, J.; Wang, J.; Aggarwal, C.; and Huang, J. 2015. STREAMCUBE: Hierarchical spatio-temporal hashtag clustering for event exploration over the Twitter stream. In *2015 IEEE 31st international conference on data engineering*, 1561–1572. IEEE.
- Karamouzas, D.; Mademlis, I.; and Pitas, I. 2022. Public opinion monitoring through collective semantic analysis of tweets. *Social Network Analysis and Mining*, 12(1): 91–112.
- Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 1–3.
- Krioukov, D.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; and Boguná, M. 2010. Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82(3): 036106.
- Li, A.; and Pan, Y. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6): 3290–3339.
- Liu, B.; Han, F. X.; Niu, D.; Kong, L.; Lai, K.; and Xu, Y. 2020a. Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(3): 1–28.
- Liu, Y.; Liu, J.; Zhang, Z.; Zhu, L.; and Li, A. 2019. REM: From structural entropy to community structure deception. *Advances in Neural Information Processing Systems*, 32: 1–11.
- Liu, Y.; Peng, H.; Li, J.; Song, Y.; and Li, X. 2020b. Event detection and evolution in multi-lingual social streams. *Frontiers of Computer Science*, 14: 1–15.
- Lou, A.; Katsman, I.; Jiang, Q.; Belongie, S.; Lim, S.-N.; and De Sa, C. 2020. Differentiating through the fréchet mean. In *International conference on machine learning*, 6393–6403. PMLR.
- Mazoyer, B.; Cagé, J.; Hervé, N.; and Hudelot, C. 2020. A french corpus for event detection on twitter. 1–8.
- McMinn, A. J.; Moshfeghi, Y.; and Jose, J. M. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 409–418.
- Papadopoulos, F.; Psomas, C.; and Krioukov, D. 2014. Network mapping by replaying hyperbolic growth. *IEEE/ACM Transactions on Networking*, 23(1): 198–211.
- Park, J.; Cho, J.; Chang, H. J.; and Choi, J. Y. 2021. Unsupervised hyperbolic representation learning via message passing auto-encoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5516–5526.
- Peng, H.; Li, J.; Song, Y.; Yang, R.; Ranjan, R.; Yu, P. S.; and He, L. 2021. Streaming social event detection and evolution discovery in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(5): 1–33.
- Peng, H.; Zhang, R.; Li, S.; Cao, Y.; Pan, S.; and Yu, P. S. 2022. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 980–998.
- Petersen, P. 2006. *Riemannian geometry*, volume 171. Springer.
- Pohl, D.; Bouchachia, A.; and Hellwagner, H. 2012. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st international conference on world wide web*, 683–686.
- Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In

- Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.
- Ren, J.; Jiang, L.; Peng, H.; Cao, Y.; Wu, J.; Yu, P. S.; and He, L. 2022. From known to unknown: Quality-aware self-improving graph neural network for open set social event detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1696–1705.
- Ren, J.; Peng, H.; Jiang, L.; Liu, Z.; Wu, J.; Yu, Z.; and Yu, P. S. 2023. Uncertainty-guided boundary learning for imbalanced social event detection. *IEEE Transactions on Knowledge and Data Engineering*, 1–15.
- Shannon, C. E. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423.
- Sun, L.; Huang, Z.; Peng, H.; Wang, Y.; Liu, C.; and Yu, P. S. 2024. LSEnet: Lorentz Structural Entropy Neural Network for Deep Graph Clustering. In *International Conference on Machine Learning*, 1–26. PMLR.
- Ungar, A. 2022. *A gyrovector space approach to hyperbolic geometry*. Springer Nature.
- Ungar, A. A. 2001. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2): 135–147.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30: 1–11.
- Vinh, N. X.; Epps, J.; and Bailey, J. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, 1073–1080.
- Wang, R.; Zhang, Y.; Li, J.; Liu, S.; Sun, D.; Wang, T.; Wang, T.; Chen, Y.; Kara, D.; and Abdelzaher, T. 2024. MetaHKG: Meta Hyperbolic Learning for Few-shot Temporal Reasoning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 59–69.
- Wang, Y.; Liu, J.; Huang, Y.; and Feng, X. 2016. Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs. *IEEE Transactions on Knowledge and Data Engineering*, 28(7): 1919–1933.
- Wu, J.; Chen, X.; Shi, B.; Li, S.; and Xu, K. 2023. Sega: Structural entropy guided anchor view for graph contrastive learning. In *International Conference on Machine Learning*, 37293–37312. PMLR.
- Wu, J.; Chen, X.; Xu, K.; and Li, S. 2022. Structural entropy guided graph hierarchical pooling. In *International conference on machine learning*, 24017–24030. PMLR.
- Xing, C.; Wang, Y.; Liu, J.; Huang, Y.; and Ma, W.-Y. 2016. Hashtag-based sub-event discovery using mutually generative LDA in Twitter. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2666–2672.
- Zhao, Q.; Mitra, P.; and Chen, B. 2007. Temporal and information flow based event detection from social text streams. In *Proceedings of the 22nd national conference on Artificial intelligence-Volume 2*, 1501–1506.
- Zou, D.; Peng, H.; Huang, X.; Yang, R.; Li, J.; Wu, J.; Liu, C.; and Yu, P. S. 2023. Se-gsl: A general and effective graph structure learning framework through structural entropy optimization. In *Proceedings of the ACM Web Conference 2023*, 499–510.

## Structural Entropy

**Definition 3** (H-dimensional Structural Entropy (Li and Pan 2016)). Given a weighted graph  $G = (\mathcal{V}, \mathcal{E})$  with weight function  $w$  and a partitioning tree  $\mathcal{T}$  of  $G$  with height  $H$ , the *structural information* of  $G$  with respect to each non-root node  $\alpha$  of  $\mathcal{T}$  is defined as

$$\mathcal{H}^{\mathcal{T}}(G; \alpha) = -\frac{g_{\alpha}}{\text{vol}(G)} \log_2 \frac{\text{vol}(T_{\alpha})}{\text{vol}(T_{\alpha^-})}. \quad (15)$$

In the *partitioning tree*  $\mathcal{T}$  with root node  $\lambda$ , each tree node  $\alpha$  is associated with a subset of  $\mathcal{V}$ , denoted as module  $T_{\alpha}$ , and the immediate predecessor of it is written as  $\alpha^-$ . The module of the leaf node is a singleton of the graph node. The scalar  $g_{\alpha}$  is the total weights of graph edges with exactly one endpoint in module  $T_{\alpha}$ . Then, the  $H$ -dimensional structural information of  $G$  by  $\mathcal{T}$  is given as,

$$\mathcal{H}^{\mathcal{T}}(G) = \sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} \mathcal{H}^{\mathcal{T}}(G; \alpha). \quad (16)$$

Traversing all possible partitioning trees of  $G$  with height  $H$ ,  $H$ -dimensional structural entropy of  $G$  is defined as

$$\mathcal{H}^H(G) = \min_{\mathcal{T}} \mathcal{H}^{\mathcal{T}}(G), \quad \mathcal{T}^* = \arg \min_{\mathcal{T}} \mathcal{H}^{\mathcal{T}}(G), \quad (17)$$

where  $\mathcal{T}^*$  is the optimal tree of  $G$  which encodes the self-organization and minimizes the uncertainty of the graph.

## Hyperbolic Space

### Riemannian Manifold

Riemannian manifold is a real and smooth manifold  $\mathbb{M}$  equipped with Riemannian metric tensor  $g_x$  on the tangent space  $\mathcal{T}_x \mathbb{M}$  at point  $x$ . A Riemannian metric assigns to each  $x \in \mathbb{M}$  a positive-definite inner product  $g_x : \mathcal{T}_x \mathbb{M} \times \mathcal{T}_x \mathbb{M} \rightarrow \mathbb{R}$ , which induces a norm defined as  $|\cdot| : v \in \mathcal{T}_x \mathbb{M} \mapsto \sqrt{g_x(v, v)} \in \mathbb{R}$ .

An exponential map at point  $x \in \mathbb{M}$  is denoted as  $\exp_x(\cdot) : u \in \mathcal{T}_x \mathbb{M} \mapsto \exp_x(u) \in \mathbb{M}$ . It takes a tangent vector  $u$  in the tangent space at  $x$  and transforms  $x$  along the geodesic starting at  $x$  in the direction of  $u$ .

A logarithmic map at point  $x$  is the inverse of the exponential map  $x$ , which maps the point  $y \in \mathbb{M}$  to a vector  $v$  in the tangent space at  $x$ .

### Poincaré ball model

The Riemannian metric tensor at  $x$  is  $g_x^{\mathbb{B}_{\kappa}} = (\lambda_x^{\kappa})^2 g_x^{\mathbb{E}}$ , where  $\lambda_x^{\kappa} = \frac{2}{1+\kappa\|x\|^2}$  and  $g_x^{\mathbb{E}} = \mathbf{I}$  is the Euclidean metric.

## Technical Details

### Implementation Details

The experiments are implemented using the PyTorch framework and run on a machine with eight NVIDIA Tesla A100 (40G) GPUs. The baseline methods are implemented with their original configurations. For HISEvent, we set the hyperparameter  $n$  to 200 to mitigate occasional deadlock issues and reduce computational runtime. For SBERT, we adopt ‘all-MiniLM-L6-v2’ to obtain the initial message representation for English Twitter (384 dimensions)

and ‘distiluse-base-multilingual-cased-v1’ for French Twitter (512 dimensions). We set the training epochs to 200, with an early stop patience of 50, the learning rate to  $1e-3$ , and the dropout rate to 0.4. For message construction, the threshold search space is set to  $\Pi = [0.4, 0.6]$  and the search step to 0.05. For SAMG, the anchor number is determined based on the number of messages as  $M = N/\epsilon$ , where  $\epsilon = \{20, 30, 40\}$ . For the HGAE, the number of layers is set to 2, the hidden feature size is set to 128, and the output feature size is set to 64. For DSI, the number of layers is set to 2, the hidden feature size to 64, the output feature size to 3, the height of the encoding tree to 2, and  $N_{h-1} = 500$ . For BERT, we adopt ‘base-multilingual-cased’ for both languages (768 dimensions). For BERT and SBERT, PLMs are utilized to obtain message representations, followed by k-means clustering to derive results. Baselines requiring a pre-defined number of events for clustering are configured with the ground truth number (a necessity for evaluation but impractical in real-world scenarios), marked in the result tables with ‘\*’. We report the average score over 5 individual runs.

### Efficiency Analysis on the English Twitter

The results for graph construction time, running time, and overall time on the English Twitter Dataset are presented in Figure 5. Consistent with the results on the French Twitter dataset, HyperSED’s progress demonstrates a significant lead in efficiency compared to the unsupervised SED baseline HISEvent. Specifically, in message block M1, HyperSED surpasses HISEvent by a substantial margin of 38.32 times and, at a minimum, outperforms HISEvent by 7.81 times in message block M15. Even in the offline scenario with the highest message size, the running time is 1152 seconds, a 120 times improvement over HISEvent’s runtime.

### Case Study & Visualization

We conduct a case study on the message block  $M_{20}$  from the English Twitter dataset, utilizing the ‘w/o SAMG’ to construct the hyperbolic partitioning tree and visualize the results, as it provides a message-wise view. We apply stereographic projection  $\Psi$  (Bachmann, Bécigneul, and Ganea 2020) to the Poincaré ball model to obtain the corresponding Poincaré disc, which is preferable to visualization. The visualization figures are shown in Figure 4, Figure 4(a) is the result of the message-wise partitioning tree, and Figure 4(b) is the ground-truth of the message nodes. The findings indicate that the learned clusters exhibit clear separation and closely match the ground-truth clusters. In the constructed partitioning tree, the identified event illustrated in Figure 4(a) appears to show fewer distinctions compared to those in the ground truth, as evidenced by the prevalence of dark pink nodes at the 7 o’clock position. Conversely, Figure 4(b) exhibits a range of colors, implying increased diversity. This disparity may arise from the similarity in their representations, where the close proximity of nodes to each other is visually discernible in the disc representation.

Table 5: Graph construction time, running time, and overall time of unsupervised SED methods on English Twitter.

Blocks		Offline	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	M <sub>6</sub>	M <sub>7</sub>	M <sub>8</sub>	M <sub>9</sub>	M <sub>10</sub>
Statistics	# Messages	13769	8722	1.491	1835	2010	1834	1276	5278	1560	1363	1096
	# Events	488	41	30	33	38	30	44	57	53	38	33
Construction Time (s)	HISEvent	1290	836	126	156	175	158	104	521	132	109	91
	HyperSED	1030	504	19	25	28	24	11	177	17	12	9
Running Time (s)	HISEvent	14649	19471	471	327	379	254	234	5779	388	358	242
	HyperSED	122	26	8	8	14	8	8	23	8	7	7
Overall Time (s)	HISEvent	15939	20307	597	483	554	412	338	6300	520	467	333
	HyperSED	1152	530	27	33	42	32	19	200	25	19	16
Speed-up (times ×)		13.84	38.32	22.11	14.64	13.19	12.88	17.79	31.50	20.80	24.58	20.81
Blocks		M <sub>11</sub>	M <sub>12</sub>	M <sub>13</sub>	M <sub>14</sub>	M <sub>15</sub>	M <sub>16</sub>	M <sub>17</sub>	M <sub>18</sub>	M <sub>19</sub>	M <sub>20</sub>	M <sub>21</sub>
Statistics	# Messages	1232	3237	1972	2956	2549	910	2676	1887	1399	893	2410
	# Events	30	42	40	43	42	27	35	32	28	34	32
Construction Time (s)	HISEvent	105	290	183	271	252	70	249	165	115	69	214
	HyperSED	12	66	36	63	61	7	61	26	14	5	36
Running Time (s)	HISEvent	220	620	793	703	357	109	684	802	187	101	1471
	HyperSED	9	12	15	11	17	4	11	8	5	5	12
Overall Time (s)	HISEvent	325	910	976	974	609	179	933	967	302	170	1685
	HyperSED	21	78	51	74	78	11	72	34	19	10	48
Speed-up (times ×)		15.48	11.67	19.14	13.16	7.81	16.27	12.96	28.44	15.89	17.00	35.10

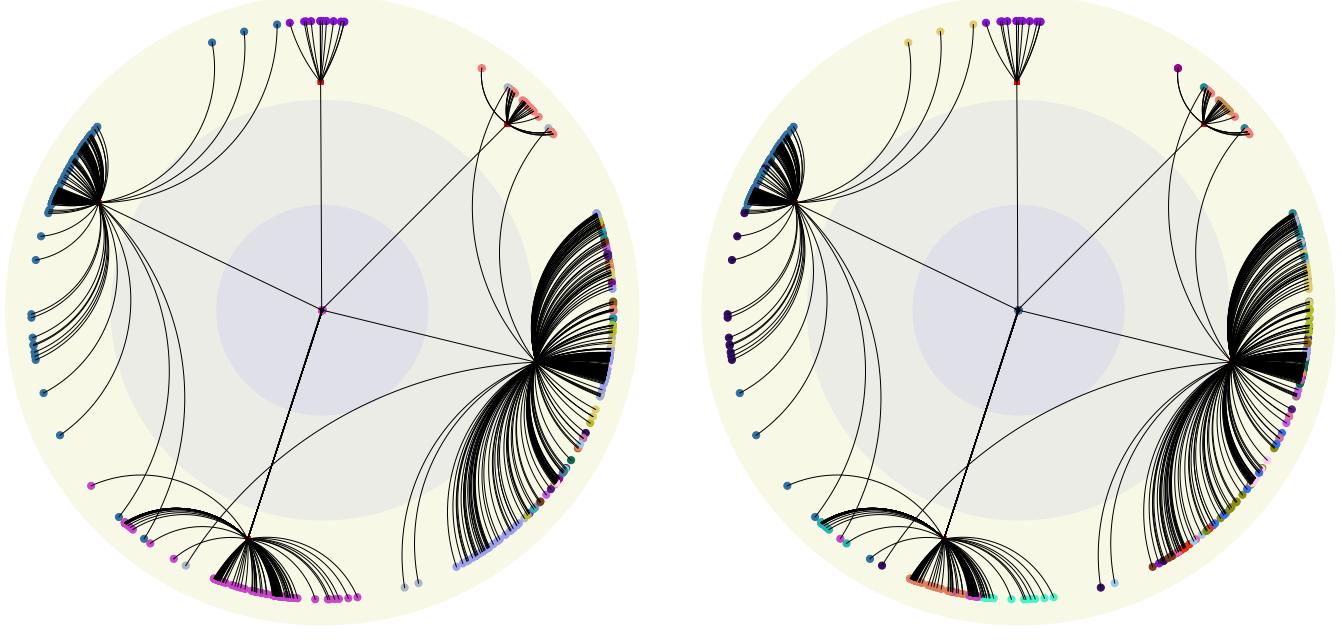


Figure 4: Caption