
Dynamic Scale-free Graph Embedding via Self-attention

Dingyang Duan^{1,2}, Daren Zha^{1,2}, Xiao Yang^{3,4},
Jiahui Shen^{1,2,*} and Nan Mu^{1,2}

¹*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*

²*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*

³*Aerospace Internet of Things Technology Co., Ltd, Beijing, China*

⁴*China Aerospace Times Electronics Co., Ltd, Beijing, China*

E-mail: duandingyang@iie.ac.cn; zhadaren@iie.ac.cn; yangxiao@casciot.com; shenjiahui@iie.ac.cn; munan@iie.ac.cn

**Corresponding Author*

Received 01 October 2022; Accepted 30 January 2023;
Publication 14 April 2023

Abstract

Graph neural networks (GNNs) have recently become increasingly popular due to their ability to learn node representations in complex graphs. Existing graph representation learning methods mainly target static graphs in Euclidean space, whereas many graphs in practical applications are dynamic and evolve continuously over time. Recent work has demonstrated that real-world graphs exhibit hierarchical properties. Unfortunately, many methods typically do not account for these latent hierarchical structures. In this work, we propose a dynamic network in hyperbolic space via self-attention, referred to as DynHAT, which leverages both the hyperbolic geometry and attention mechanism to learn node representations. More specifically, DynHAT captures hierarchical information by mapping the structural graph onto hyperbolic space, and time-varying dynamic evolution by flexibly weighting historical representations. Through extensive experiments on three real-world

Journal of Web Engineering, Vol. 22_1, 55–78.

doi: 10.13052/jwe1540-9589.2214

© 2023 River Publishers

datasets, we show the superiority of our model in embedding dynamic graphs in hyperbolic space and competing methods in a link prediction task. In addition, our results show that embedding dynamic graphs in hyperbolic space has competitive performance when necessitating low dimensions.

Keywords: Dynamic graphs, hyperbolic space, self-attention, representation learning.

1 Introduction

A network graph contains a set of interconnected nodes, where each pair of nodes has a large amount of relational information. Graph neural networks (GNNs) are widely used to model complex network graphs due to their ability to learn node representations. The basic idea is to map each node to a vector in a low-dimensional representation space, while capturing the structure and properties of the network. Researchers have developed various graph mining algorithms for node classification, label recommendation, anomaly detection, and link prediction tasks. Despite the great potential for graph representation learning, embedding graphs into low-dimensional space is not simple.

Current graph neural networks face the following key challenges [41]:

1. **Structure-preserving.** To learn node representations, graph embedding should preserve the structural similarity of the network, which also means that nodes similar in the original structure space should also be similarly represented in the learned vector space. However, the similarity at the structural level is not only reflected in the local structure, but also in the global structure. Therefore, graph embedding should preserve both local and global structure.
2. **Content-preserving.** In addition to structural information, various nodes and edges are attached with valuable attribute content. Nodes with similar properties should also have similar node representations. Moreover, many real-world graphs, such as social networks where graph structure continuously evolves over time, typically exhibit scale-free or hierarchical structure [6]. Many models in Euclidean space underestimate the inherent hierarchical properties when embedding such scale-free graphs. Learning representations of dynamic structures is an essential ingredient in the study of real-world dynamic networks, as it describes the evolution of the network, which is very valuable [39].

Most of the existing studies of dynamic graph models can be divided into two different approaches: one is based on a discrete-time approach, where

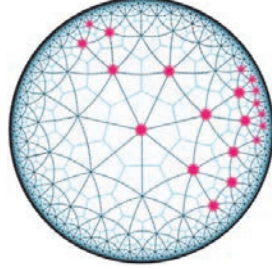


Figure 1 Tree-like structure in hyperbolic space; the volume grows exponentially.

the evolution of a dynamic graph can be described by a sequence of static graphs with fixed timestamps. There are also continuous-time methods that model evolution at a finer time granularity to encompass different events in real-time [32]. Essentially, both approaches are primarily designed for graphs in Euclidean space, which are used by existing GCNs and have highly distorted embedding [4,5,26]. In such graphs, which can be approximated by a tree-like structure, the graph volume expands exponentially. For example, in an n -ary tree, the number of nodes with a distance less than r grows as n^r . Specifically, the expressive capacity of hyperbolic space is much larger than that of Euclidean space (Figure 1 is an example). In a two-dimensional Euclidean space, the length of a circle and the area of a disk of radius r are given as $2\pi r$ and πr^2 , respectively. However, in a two-dimensional hyperbolic space with curvature $-c^2$, $c > 0$, the growth of volume is exponential, length as $2\pi \sinh(cr)$ and $2\pi(\cosh(cr) - 1)$, contrasted with Euclidean space growing only polynomially. On the other hand, it has also been pointed out in numerous works [27, 29] that such a tree with a hierarchical structure can be embedded into hyperbolic space by low distortion. In contrast, Euclidean space cannot represent a tree with low distortion [18].

Previous work on network representation learning has focused on static graphs that either represent existing connections at a fixed moment in time (i.e., graph snapshots) or represent time-varying node interactions that are aggregated into a single static graph [1]. Several challenges arise when developing methods for embedding dynamic graphs due to the complex structure of time-varying graphs. This requires capturing their dynamic behavior, including adding or removing nodes and edges [3], labels and other properties that change over time [17]. In this case, it is common to build a recurrent neural network (RNN) that summarizes historical snapshots via hidden state, for example [10, 11], which mainly focuses on mining the pattern of graph evolution. However, the major disadvantage of an RNN

is the vanishing gradient and it scales poorly with increasing time steps. The attention mechanism considerably improves the efficiency and accuracy of perceptual information processing [21]. These are called self-attention when a definite sequence is used as both input and context, and they were originally designed to facilitate RNNs to capture long-range dependencies. Since changes in graphs can be periodic and frequent, self-attention is able to draw context from all past graph snapshots to adaptively assign interpretable weights for previous time steps [8].

Inspired by the insight mentioned above, we propose a novel neural architecture named a dynamic network in hyperbolic space via self-attention, referred to as DynHAT, to learn latent node representations on dynamic graphs [8]. The model fully leverages the hyperbolic geometry to capture hierarchical information and evolving behaviors over multiple time steps by flexibly weighting historical representations. The main contributions of this work are summarized below:

- We propose a novel hyperbolic temporal graph embedding model, named DynHAT, to learn temporal regularities, topological dependencies, and implicitly hierarchical organization.
- DynHAT learns node representations by a joint attention mechanism along two aspects: hyperbolic structure attention and Euclidean temporal attention. The hyperbolic structure attention extracts nodes information according to their importance to the center node, while the Euclidean temporal attention captures the most relevant historical contexts from long-range variations.
- Experimental results on three real-world datasets demonstrate consistent performance improvement in link prediction task. Moreover, we conduct experiments on multi-head attention to demonstrate the effectiveness of multiple subspaces in adequately capturing the evolution of nodes. Finally, we provide experimental evidence that our model can achieve the same effect using only 16 dimensions in hyperbolic space rather than using 64 dimensions in Euclidean space.

2 Related Works

In this section, we introduce dynamic graph embedding, as well as hyperbolic representation learning methods.

Static network representation models have been intensively studied to support applications such as node classification and link prediction in the last

few years. However, various real-world networks exhibit dynamic behavior, where not only the node and edge properties may change as time goes by, but also the topology may evolve continuously over time, including addition and removal of nodes and edges. Naturally, there are two ways to embed such a dynamic graph, namely, discrete-time approaches, which can be viewed as a sequence of network snapshots with fixed timestamps; and the continuous-time approach in a stream, where its lifespan is a continuous set, and thus evolution is modeled at a finer temporal granularity to encompass different events in real time [36, 38]. Correspondingly, there are two typical variants of the network data models that embed temporal graphs. Computationally, dynamic graph models assuming discrete-time domains are easier to manipulate [1]. Following this, our proposed model is based on a discrete-time approach.

DynGEM [11] employs a deep auto-encoder as its core to learn the graph and uses an RNN to model the relations over time. Dyngraph2vec [10] also utilizes an auto-encoder structure to learn node embedding and capture temporal dynamics by RNN methods. DynamicTriad [44], a semi-supervised model, describes how open triads evolve into closed triads along graph evolution. Considering DynamicTriad [44] may ignore distinct temporal behaviors, DySAT [28] applies the attention mechanism [33], learning structural and temporal attention to adaptively obtain useful information for embedding. DySAT [28] has a clear architecture: decoupling graph evolution into two layers, the structural layer and the temporal layer. Different with RNN methods, DySAT [28] can capture the most relevant historical contexts through attention mechanism. DySAT [28] outperforms state-of-the-art performance baselines on real-world datasets.

Essentially, these methods are designed primarily for the graphs in Euclidean space in that it is natural of our intuition-friendly and visible three-dimensional space [40, 43]. Many real-world graphs, such as protein interaction and social networks, often exhibit scale-free or hierarchical structure [2]. In such situations, Euclidean space could not provide the most powerful and adequate geometrical representations. [40] points out two benefits of hyperbolic geometry. The first one is that hyperbolic space exhibits minimal distortion and is well-suited for modeling tree-like graphs. The other one is that even with a low-embedding dimension space, hyperbolic models are surprisingly able to produce high-quality representation compared with Euclidean space. Inspired by this insight, the study of tree-like graph data in hyperbolic space has received increasing attention, such as the hyperbolic graph convolutional neural network (HGCM) [4], hyperbolic graph

neural networks (HGNN) [19], the Poincare embedding model [20] and the hyperbolic graph attention network (HGAT) [43]. [20] proposed a Poincare embedding model for learning hierarchical representations of symbolic data. HGCN [4] is a generalization of inductive GCNs in hyperbolic geometry that benefits from the expressiveness of both graph neural networks and hyperbolic embedding. HGAT [43] employs the framework of gyrovector space to implement graph processing in hyperbolic space and design an attention mechanism [34] based on hyperbolic proximity. An increasing number of studies report that hyperbolic models are more competitive than Euclidean models in a broad spectrum of applications, including NLP [7, 31] and image processing works [15].

Although these models can technically be applied to dynamic graphs, they cannot capture the temporal evolution of the nodes and edges of a dynamic graph. Recently, hyperbolic dynamic graph models have been proposed where the graph is considered to be dynamic. HTGN [39] proposes a hyperbolic gated RNN to capture evolving behaviors. HVGNN [30] devises a temporal GNN based on a theoretically grounded time encoding approach. [35] presents temporal random walks to obtain training samples from continuous-time networks. However, these models fail to capture the long-range variations and are commonly deficient when the nodes exhibit considerable evolutionary behaviors.

Our proposed framework fills this gap. Specifically, our model [8] computes node representations by joint self-attention in hyperbolic space along two dimensions: topology information and temporal evolution. It can take full advantage of hyperbolic geometry’s exponential capacity and hierarchical awareness and effectively capture temporal evolutionary patterns.

3 Preliminaries

In this section, we first define dynamic graphs. Then, we introduce some fundamentals of hyperbolic geometry.

3.1 Problem Formulation

In this work, we formally define the problem of dynamic graph representation learning. A dynamic graph is defined as a series of observed static graph snapshots, $\mathbb{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_T\}$ where T is equal to the number of time steps. Each snapshot $\mathcal{G}_t = (V_t, A_t) \in \mathcal{G}$ is a weighted and undirected network snapshot recorded at time t , where V_t is the set of vertices and A_t is the

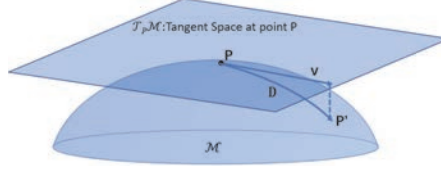


Figure 2 The Riemannian manifold \mathcal{M} and the tangent space $\mathcal{T}_x\mathcal{M}$ at point P . Tangent vector v is the projection of P' . D is the geodesic between P and P' .

corresponding adjacency matrix at time step t . It is important to mention that in the continuous-time approach T is a continuous set, which is different from ours. Computationally, we use the discrete-time approach following [11, 28] because the discrete-time domain is easier to manipulate. We construct one-hot embedding as the initial node features from these vertices.

3.2 Hyperbolic Geometry

Hyperbolic space commonly refers to manifolds with constant negative curvature [16]. Here, we will briefly introduce the fundamental concepts about hyperbolic geometry.

Riemannian manifold: A smooth manifold equipped with a Riemannian metric, which is a family of inner products on the associated tangent space: $\mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \rightarrow \mathbb{R}$, is called a Riemannian manifold.

Tangent space: The tangent space $\mathcal{T}_x\mathcal{M}$ of a manifold \mathcal{M} at a point $x \in \mathcal{M}$ is defined as the first-order approximation of the n -dimension vector space \mathcal{M} (see Figure 2). Since the tangent space of a point on a Riemannian manifold always is Euclidean, it allows us to perform graph operations in hyperbolic space.

There are five isometric models that one can work with [9], including the Poincare ball model, the Lorentz model (also named hyperboloid model) and the Klein model, which show different features. Similarly to [20], we choose to work in the Poincare ball model.

Poincare ball model: With negative curvature $-c$ ($c \geq 0$) the Poincare ball model $(\mathbb{H}^{n,c}, g_{\mathbb{H}})$ is an open n -dimensional ball, defined by the manifold $\mathbb{H}^{n,c} = \{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\|^2 \leq 1\}$, equipped with the following Riemannian metric:

$$g_x^{\mathbb{H}} = (\lambda_x^c)^2 g^E, \quad \text{where } \lambda_x^c = \frac{2}{1 - c\|\mathbf{x}\|^2}, g^E = \mathbf{I}_n. \quad (1)$$

If $c = 0$, then the Poincare ball equals the Euclidean space. Here, we fix $c = 1$ to keep things simple.

Exponential map: The input of the proposed model is the node feature, whose norm could live in Euclidean space. To make the node feature available in hyperbolic space, we use an exponential map to project the feature into the hyperbolic space [43]. For $a \in \mathcal{T}_{\mathbf{x}'}\mathbb{H}$, then the exponential map is formulated as:

$$\exp_{\mathbf{x}'}(a) = \mathbf{x}' \oplus \left(\tanh\left(\frac{\lambda_{\mathbf{x}'}\|a\|}{2}\right) \frac{a}{\|a\|} \right). \quad (2)$$

\oplus denotes the Mobius addition [24, 39], for any $u, v \in \mathbb{H}$:

$$u \oplus v := \frac{(1 + 2\langle u, v \rangle + \|v\|^2)u + (1 - \|u\|^2)v}{1 + 2\langle u, v \rangle + \|u\|^2\|v\|^2}. \quad (3)$$

Logarithmic map: As the inverse operation of the exponential map, for a point $b \in \mathbb{H}$, the logarithmic map is given by:

$$\log_{\mathbf{x}'}(b) := \frac{2}{\lambda_{\mathbf{x}'}} \operatorname{arctanh}(\|-\mathbf{x}' \oplus b\|) \frac{-\mathbf{x}' \oplus b}{\|-\mathbf{x}' \oplus b\|}. \quad (4)$$

Distance: The distance between two points $u, v \in \mathbb{H}$ is given as:

$$d_{\mathbb{H}}(u, v) = \operatorname{arcosh} \left(1 + 2 \frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)} \right). \quad (5)$$

The boundary of the Poincare ball is not a part of the manifold, but represents the infinitely distant point [20].

4 Proposed Model

The overall framework of the proposed model (DynHAT) [8] is illustrated in Figure 3. DynHAT has two primary modules: *hyperbolic structure attention (HSA)*; *Euclidean temporal attention (HSA)*, which benefit from the expressiveness of both hyperbolic embeddings and temporal evolutionary embeddings. More specifically, our model can be summarized as two procedures:

1. Given the original input node feature at each time snapshot, the projection into hyperbolic space is used as the latent representation. This

procedure preserves both local and global geometric information. Similar to other models in hyperbolic space [4, 13, 43], the HSA module aims to learn the attention coefficient based on latent representation using hyperbolic similarity and to aggregate representations which capture its evolving neighborhoods in hyperbolic space. This process is presented in Algorithm 1.

2. These sequences of node representations are then fed as input to the temporal attention, which are defined in Euclidean space since the operations in hyperbolic space are generally more complex and time-consuming [8]. Due to superior self-attention, this unit fuses the final embeddings by figuring out the importance of the graph snapshots at each time step. The output of the temporal module comprises the set of final dynamic node representations. Furthermore, we endow our model with expressivity to capture dynamic graph evolution from different latent perspectives through multi-head attention following [28, 33].

Then, we consider the link prediction task and train our model using the cross-entropy loss function. Finally, we feed the aggregated representations to a loss function for the downstream task. Specifically, we formulate the learning objective to maximize the probability of having connected nodes and minimize the probability of no interconnected nodes. We solve the optimization problem using hyperbolic distance, following [13, 20, 39], as:

$$\theta' \leftarrow \arg \min \mathbf{L}(\theta). \tag{6}$$

The loss function $\mathbf{L}(\theta)$ is based on binary cross-entropy which is defined as:

$$\mathbf{L}(\theta) = \sum_{t=1}^T \sum_{v \in V} \left(\sum_{u \in \mathcal{N}^t} -\log(p(h_u^t, h_v^t)) - \sum_{u' \in P_n^t(v)} \log(1 - p(h_{u'}^t, h_v^t)) \right). \tag{7}$$

p is the probability which could be inferred by the Fermi–Dirac function [4, 39], which is defined as:

$$p := \frac{1}{1 + e^{d_{\mathbb{H}}(u,v)-2}}. \tag{8}$$

For the minimization of Equation (6), we optimize our model with Adam optimizer and set the learning rate to 0.01.

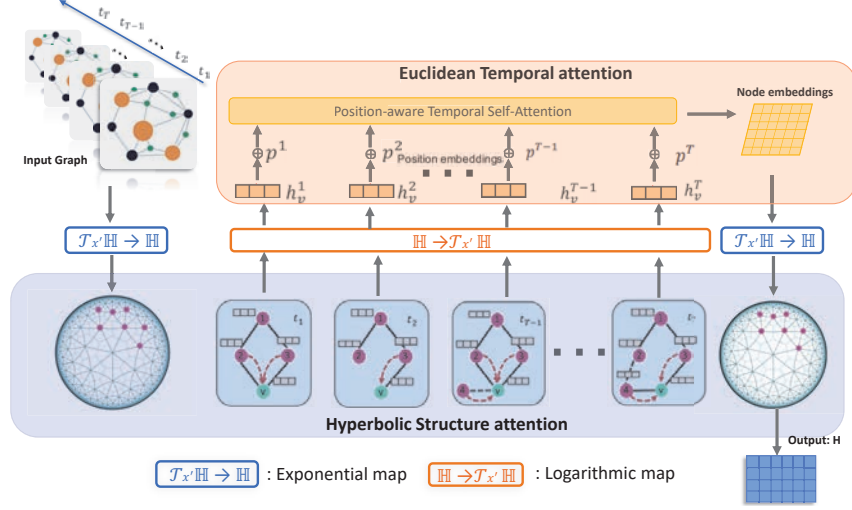


Figure 3 Architecture of DynHAT.

Algorithm 1 Process of hyperbolic structure attention

Input: The set of graph snapshots and node features f ;

Output: The sequences of node representations;

- 1: Project the feature into the hyperbolic space, $\mathbf{x}_i^{\mathcal{H}} = \exp_{x'}(\mathbf{x}_i^E)$;
 - 2: Then transform $\mathbf{x}_i^{\mathcal{H}}$ into a higher-level latent representation $m_i^{\mathcal{H}} = W \otimes^c \mathbf{x}_i^{\mathcal{H}} \oplus^c b$;
 - 3: Compute the attention coefficient α_{ij} ;
 - 4: Calculate the node embedding through a self-attentional aggregation of neighboring node;
 - 5: Feed the output embeddings back to the Euclidean space $X_t^E = \log_{x'}(\mathbf{x}_i^{\mathcal{H}})$;
 - 6: **return** Node embeddings X_t^E in each time step;
-

Algorithm 2 Process of Euclidean temporal attention

Input: Node embeddings X_t^E in each time step; Position embedding of each time step;

Output: The final node representations;

- 1: The position embeddings are combined with the output of the HSA module $X_t^1 + p^1, X_t^2 + p^2, \dots, X_t^T + p^T$;
 - 2: Transform the (query, key, value) to different space through linear projection matrices $X_v W_q, X_v W_k, X_v W_v$ respectively;
 - 3: Then we construct a mask matrix M_{ij} to encoder temporal order;
 - 4: Calculate the node embedding Z_v via self-attention layer, feeding with M_{ij} and sequence of node combined with position;
 - 5: Feed the output embeddings back to the hyperbolic space $H^{\mathcal{H}} = \exp_{x'}^c(Z_v)$;
 - 6: **return** The final Node embeddings;
-

5 Experiments and Analysis

We evaluate our proposed model on the link prediction task. To evaluate the effectiveness of our model, we have conducted experiments on three widely used datasets and corroborated our approach’s effectiveness in the link prediction task [8].¹ *Enron* [25] and *UCI* [22] datasets are communication networks whose links denote the user interactions. *MovieLens* [14] contains movie metadata information and user attribute information. In this paper, we use a subset of *MovieLens*, which consists of user–tag interactions where the links denote the users with tags on specific movies. Their detailed descriptions are summarized in Table 1. Note that the duration of each snapshot will affect the total number of snapshots. The proper granularity is more beneficial to capture evolving patterns [37]. Details of the total number of nodes and links on *Enron*, in different time slices corresponding to the 300-day time interval, are provided in Table 2. On *UCI* and *MovieLens*, we construct for a 20-day time granularity and a 90-day time granularity, respectively, as shown in Tables 3 and 4.

Table 1 Summary of the datasets

Dataset	Enron	UCI	MovieLens
Nodes	184	1899	20,537
Edges*	2216	13,838	43,760
Start date	1980-01-01	2004-04-15	2005-12-23
End date	2002-06-22	2004-10-26	2009-01-05
Time duration	8207 days	193 days	1108 days

* Edge counts denotes the total edges across all time steps.

Table 2 Detail of 300-day time interval on *Enron*

Time Snapshot	Time Period	Nodes	Edges
slice 1	19800101–19981112	43	174
slice 2*	19981113–19981123	43	8
slice 3	19981124–19990919	58	1370
slice 4	19990920–20000713	120	15630
slice 5	20000714–20010509	152	61545
slice 6	20010510–20020306	184	45811
slice 7	20020306–20220622	184	871

* Due to the nodes not being evenly distributed, the number of nodes is 0 from 19800102 to 19981112.

¹Source code: <https://github.com/duandingyang/dynhat.git>

Table 3 Detail of 20-day time interval on UCI

Time Snapshot	Time Period	Nodes	Edges
Slice 1	20040415–20040504	743	10589
Slice 2	20040505–20040524	1383	24789
Slice 3	20040525–20040613	1698	14031
Slice 4	20040614–20040703	1736	1822
Slice 5	20040704–20040723	1765	2242
Slice 6	20040724–20040812	1794	1685
Slice 7	20040813–20040901	1830	1848
Slice 8	20040902–20040921	1855	1354
Slice 9	20040922–20041011	1890	1121
Slice 10	20041012–20041026	1899	354

Table 4 Detail of 90-day time interval on Movielens

Time Snapshot	Time Period	Nodes	Edges
Slice 1	20051223–20060322	6058	16704
Slice 2	20060323–20060620	7588	6995
Slice 3	20060621–20060918	8825	5217
Slice 4	20060919–20061217	10030	6542
Slice 5	20061218–20070317	11529	11799
Slice 6	20070318–20070615	12766	9035
Slice 7	20070616–20070913	15193	10452
Slice 8	20070913–20071213	16097	4670
Slice 9	20071213–20080311	17168	6155
Slice 10	20080312–20080609	18036	4551
Slice 11	20080609–20080907	19494	6076
Slice 12	20080908–20081206	20265	4998
Slice 13	20081207–20090105	20537	2386

We compare our model with the following methods, including static and dynamic graph embedding methods. Node2vec [12] is a static embedding method to generate vector representations of nodes on a graph. GAT [34] leverages masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations. In terms of the dynamic graph, the EvolveGCN [23] model is proposed to compute a separate GCN model for each time step, and DySAT [28] employs GAT [34] as a static layer and self-attention mechanism to capture the temporal graph evolution. As for the hyperbolic model, HGCN [4] leverages the expressiveness of GCNs and hyperbolic geometry to learn node representations.

5.1 Link Prediction

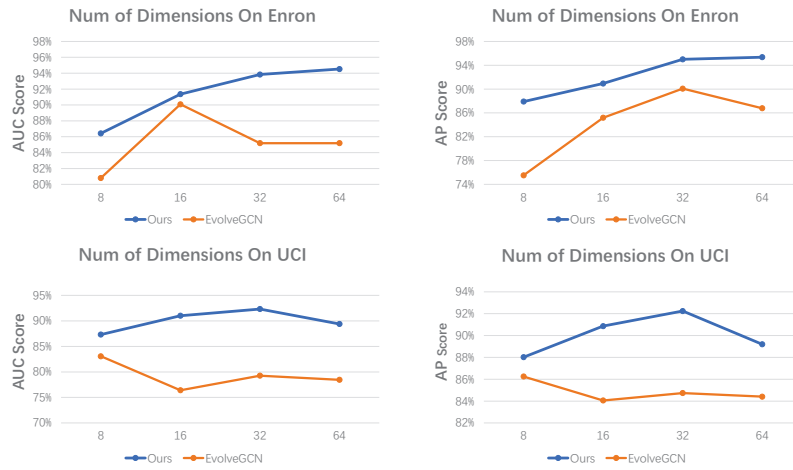
Link prediction is one commonly adopted way to evaluate the quality of node representations. In this paper, we conduct experiments on single-step and multi-step link prediction. Using node representations trained on graph snapshots up to step t , the single-step link prediction aims to predict the links between nodes at time step $t + 1$. Take the Movielens dataset as an example, and we use the latest node representations at time step t to classify user–tag interactions into links and non-links at step $t + 1$. For the 90-day time granularity, the data from 20051223 to 20090105 is split into 13 time snapshots. The slices from 1 to 12 are given as input to our model, and the last slice is taken as test data. For multi-step link prediction, we utilize the latest node representations, for example slice 7, to predict the links at multiple future time steps 8, 9, 10, 11, 12, 13 ($\{t + 1, \dots, t + \Delta, \Delta = 6\}$), which can exhibit evolutionary behaviors. We use the average precision (AP) and area under the ROC curve (AUC) metric to evaluate link prediction performance.

The results of single-step link prediction are presented in Table 5. Our results indicate that the proposed model achieves gains of about 3–5% AUC and AP, comparing to the best baseline across all datasets. On the one hand, the runner-up goes to the other temporal graph embedding model, which confirms the importance of temporal regularity in dynamic graph modeling. On the other hand, for the hyperbolic model in the static graph, our model consistently outperforms HGCN. Despite being in Euclidean space, DySAT achieves competitive performance in dynamic embedding approaches across several datasets, falling short of the DynHAT in AUC and AP scores by about 3%. One possible explanation is that joint structural and temporal modeling with expressive aggregators, such as multi-head attention, may have a significant impact in higher performance on link prediction. The performance gap between DynHAT and DySAT suggests the substantial benefit of hyperbolic geometry. Second, HGCN also has relatively good performance despite being agnostic to temporal information, which indicates further improvements to DynHAT on transforming the embeddings from Euclidean space to hyperbolic space.

Figure 4 shows the experiments on the multi-step link prediction task. We compare the performance of DynHAT against EvolveGCN on multi-step link prediction over $t + \Delta$ time steps, where Δ is equal to 6. The time granularity of the Enron and UCI are 90-day and 15-day respectively. Enron has 17 snapshots and UCI has 15 snapshots. Take multi-step 2 as an example, we use snapshots from slice 1 to slice 15 on Enron and slice 13 on UCI as training data, and the last two snapshots as test data. Specifically, we

Table 5 AUC (left) and AP (right) scores of single-step link prediction result

Dataset	AUC			AP		
	Enron	UCI	MovieLens	Enron	UCI	MovieLens
GAT	84.98	82.11	84.3	84.87	81.42	86.32
HGCN	86.76	84.31	87.42	84.45	85.28	86.32
EvolveGCN	88.62	84.68	84.71	88.81	86.73	85.72
DySAT	88.71	86.55	89.41	89.60	86.81	89.77
DynHAT	92.39	91.01	92.25	94.07	90.86	92.39

**Figure 4** Performance of DynHAT with different models on multi-step link prediction.

observe that the performance of the EvolveGCN model decline to a dramatic degree, while our model retains a stable result over time. Despite EvolveGCN being designed in an RNN to capture the dynamism of the graph sequence by evolving the GCN parameters, it exhibits large variations. This exemplifies the temporal attention module’s capacity to capture the most relevant historical context. Additionally, we observe more drastic drops on Enron than UCI in performance over time for all models. A possible explanation might be that Enron has a more extended period (8270 days) than UCI (193 days). Nevertheless, DynHAT has achieved better performance than EvolveGCN.

5.2 Analysis Number of Attention Heads

The attention mechanism has been adequately utilized in our proposed architecture, where not only does the structural procedure represent the topological

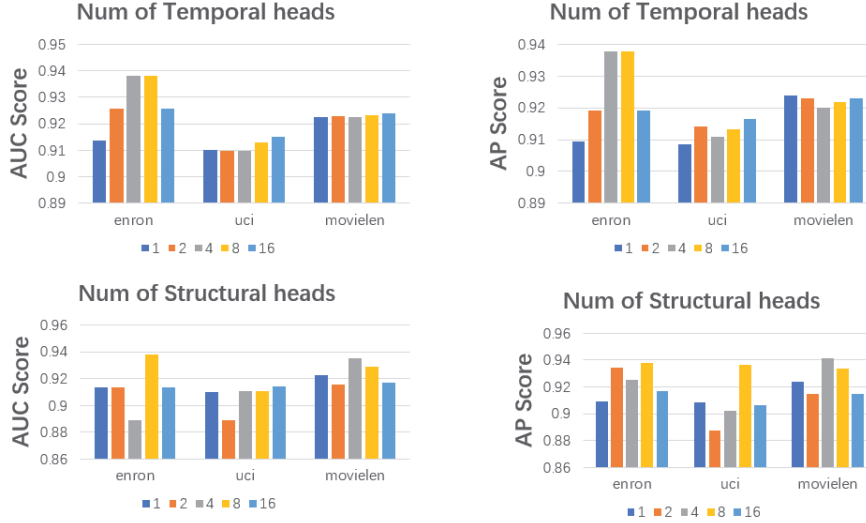


Figure 5 Experimental results under different numbers of attention heads.

dependencies at each time step in hyperbolic space, but also the temporal procedure captures the graph evolution across multiple time spans by flexibly weighting historical context. Multi-head attention, which projects the historical context to multiple subspaces based on learnable parameters, gives expressivity to our model with different latent perspectives. The performance impact of different numbers of heads is not negligible. Thus, in our experiments, we adjust the number of structural and temporal heads respectively in the range 1, 2, 4, 8 and 16 to validate the effectiveness of multi-head attention. In this experiment, we use the 16-dimension embedding space on each dataset. When we verify the temporal heads, we set the number of structural heads to a fixed value of 1. The time granularity of the Enron, UCI and Movielens datasets are 130-day, 20-day and 90-day respectively.

From the results in Figure 5, it is observed that our model benefits from multi-head attention on both structural and temporal modules. In addition, if we do not use the multi-head attention mechanism (head = 1), the model degrades to a single attention function, capturing a single facet of graph evolution, especially on Enron. One possible reason is that Enron has a long duration, which could benefit more from temporal attention heads. A large or small number of heads cannot specifically lead to a better or worse generalization.

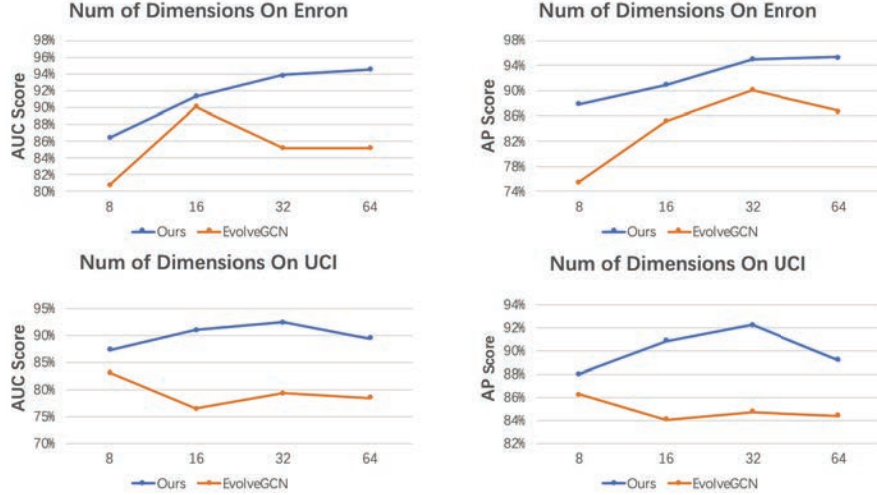


Figure 6 Experimental results under different embedding dimensions.

5.3 Analysis of Dimension

In Sections 1 and 3, we can conclude that hyperbolic space has a larger expressive capacity than Euclidean space. Therefore, the number of dimensions required in hyperbolic space is much lower than in Euclidean space in order to fully capture all the hierarchical information in the dataset. We perform the experiments in the embedding dimension of the range 8, 16, 32, and 64 to validate the performance of the low-dimensional embedding.

The results of our experiments are shown in Figure 6. Our model performs very well on both Enron and UCI datasets, especially in low dimension, owing to the representational efficiency of the hyperbolic approach. However, as we enlarge the dimension to more than a threshold, the performance of both models (DynHAT and EvolveGCN) declines to varying degrees. Increasing the dimension will increase the complexity of the model. Model complexity can lead to a too complex model that does not generalize well to new data, while overfitting can cause the model to perform well on the training data but poorly on new data. Our model has a significant advantage when the embedding dimension is smaller than 32. These results verify the claim in Section 1 that graphs can be modeled into low-dimensional hyperbolic space naturally. This allows us to learn graph representations of high quality, which is crucial for solving low-memory scenarios.

6 Limitations

In graph representation learning, models based on hyperbolic space have gained momentum in the context of machine learning due to their ability to model data with latent hierarchies. However, defining dynamic graphs in hyperbolic space involves a large number of cumbersome and complicated numerical operations, which are constantly costly compared to Euclidean space [24, 39]. In particular, the amount of graph data in the real-world is huge and it is difficult to embed such graphs in time. In addition, hyperbolic space is preferable to Euclidean space in the context of the small latent size. When the latent size is large enough, the hyperbolic space loses a bit of its competitiveness [42]. Moreover, not all real-world graph data follow a power-law distribution, and hence hyperbolic embedding cannot provide corresponding low-distortion embedding. We hope to trigger exciting future research related to the better development of other non-Euclidean deep learning approaches.

7 Conclusion

In this work, we make an effort toward investigating the graph neural network in hyperbolic space and we do a more detailed description of our proposed model DynHAT by conducting a more in-depth experiment. More specifically, our model computes dynamic node representations through two joint modules: hyperbolic structure attention (HSA) and Euclidean temporal attention (ETA). HSA leverages the superiority of hyperbolic mechanism and ETA captures the most relevant historical contexts through efficient self-attention. Experimental results show the superiority of DynHAT for link prediction on several real-world datasets. In addition, we analyze the benefits of multi-head attention in capturing the graph evolution from multiple latent facets. We show that our method has competitive performance in low dimensions.

References

- [1] Claudio DT Barros, Matheus RF Mendonça, Alex B Vieira, and Artur Ziviani. A survey on embedding dynamic graphs. *ACM Computing Surveys (CSUR)*, 55(1):1–37, 2021.
- [2] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

- [3] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 346–359. Springer, 2011.
- [4] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32:4868–4879, 2019.
- [5] Wei Chen, Wenjie Fang, Guangda Hu, and Michael W Mahoney. On the hyperbolicity of small-world and treelike random graphs. *Internet Mathematics*, 9(4):434–491, 2013.
- [6] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.
- [7] Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 59–69, 2018.
- [8] Dingyang Duan, Daren Zha, Xiao Yang, Nan Mu, and Jiahui Shen. Dynamic network embedding in hyperbolic space via self-attention. In *Web Engineering: 22nd International Conference, ICWE 2022, Bari, Italy, July 5–8, 2022, Proceedings*, page 189–203, Berlin, Heidelberg, 2022. Springer-Verlag.
- [9] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.
- [10] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyn-graph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187:104816, 2020.
- [11] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*, 2018.
- [12] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [13] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2018.

- [14] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- [15] Valentin Khruikov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020.
- [16] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.
- [17] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 387–396, 2017.
- [18] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
- [19] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.
- [20] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.
- [21] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.
- [22] Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley. Patterns and dynamics of users’ behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, 2009.
- [23] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370, 2020.
- [24] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [25] Carey E. Priebe, Youngser Park, David J. Marchette, John M. Conroy, John Grothendieck, and Allen L. Gorin. Statistical inference on attributed random graphs: Fusion of graph features and content: An

- experiment on time series of enron graphs. *Computational Statistics & Data Analysis*, 54(7):1766–1776, 2010.
- [26] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical review E*, 67(2):026112, 2003.
- [27] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.
- [28] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 519–527, 2020.
- [29] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.
- [30] Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and S Yu Philip. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4375–4383, 2021.
- [31] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 583–591, 2018.
- [32] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, volume 1050, page 4, 2018.
- [35] Lili Wang, Chenghan Huang, Weicheng Ma, RuiBo Liu, and Soroush Vosoughi. Hyperbolic node embedding for temporal networks. *Data Mining and Knowledge Discovery*, 35(5):1906–1940, 2021.
- [36] Guotong Xue, Ming Zhong, Jianxin Li, Jia Chen, Chengshuai Zhai, and RuoChen Kong. Dynamic network embedding survey. *Neurocomputing*, 472:212–223, 2022.

- [37] Hansheng Xue, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Yu Lin. Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, page 282–298, Berlin, Heidelberg, 2020. Springer-Verlag.
- [38] Menglin Yang, Ziqiao Meng, and Irwin King. Featurenorm: L2 feature normalization for dynamic graph embedding. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 731–740. IEEE, 2020.
- [39] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1975–1985, 2021.
- [40] Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. Hyperbolic graph neural networks: A review of methods and applications. *arXiv e-prints*, pages arXiv–2202, 2022.
- [41] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.
- [42] Sixiao Zhang, Hongxu Chen, Xiao Ming, Lizhen Cui, Hongzhi Yin, and Guandong Xu. Where are we in embedding spaces? In *Association for Computing Machinery, KDD '21*, page 2223–2231, New York, NY, USA, 2021.
- [43] Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Fanny Ye. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 2021.
- [44] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Biographies



Dingyang Duan received his Master of Engineering degree from Beijing Jiaotong University in 2016. He is currently a Ph.D. candidate in the School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China. He focuses on machine learning and deep learning. His current research interests are graph neural networks and knowledge graphs.



Daren Zha received his bachelor's degree from the University of Science and Technology of China and his Ph.D. degree from the University of Chinese Academy of Sciences in 2010. He has presided over a number of national key R&D programs on cyberspace security key special projects. He has published more than 20 papers in refereed journals and conferences. His current research interests include intelligent data processing related to cyberspace security, Big Data storage and cryptographic engineering.



Xiao Yang received her Master of Engineering degree from Peking University in 2019. She focuses on machine learning and deep learning. Her current research interests are graph neural networks and computer vision.



Jiahui Shen received her Ph.D. degree from the University of Chinese Academy of Sciences in 2020. Her current research focuses on machine learning and knowledge graphs.



Nan Mu received his Ph.D. degree from the University of Chinese Academy of Sciences in 2021. His research interest includes knowledge Graphs and graph neural networks.

