

THINK: Temporal Hypergraph Hyperbolic Network

Shivam Agarwal^{*♣}, Ramit Sawhney^{*◇}, Megh Thakkar[†], Preslav Nakov[†], Jiawei Han[♣], Tyler Derr[♣]

[♣]University of Illinois at Urbana-Champaign, [◇]Georgia Institute of Technology, [†]BITS Pilani

[†]Mohamed bin Zayed University of Artificial Intelligence, [♣]Vanderbilt University

{shivama2,hanj}@illinois.edu, ramit.sawhney@scheller.gatech.edu, preslav.nakov@mbzuai.ac.ae, tyler.derr@vanderbilt.edu

Abstract—Network-based time series forecasting is a challenging task as it involves complex geometric properties, higher-order relations, and scale-free characteristics. Previous work has modeled network-based series as oversimplified graphs or has ignored the power law dynamics of real-world temporal and dynamic networks, which could yield suboptimal results. With the aim to address these issues, here we propose THINK, a novel framework based on hypergraph learning that captures the hyperbolic properties of time-evolving dynamic hypergraphs. We design an elegant hyperbolic distance-aware hypergraph attention mechanism to better capture informative internal structural features on the Poincaré ball. Through quantitative and conceptual analysis on seven tasks across temporal, and time-evolving dynamic hypergraphs, we demonstrate THINK’s practicality in comparison to a variety of benchmarks spanning finance, health, and energy networks.

Index Terms—hyperbolic, hypergraphs, spatio-temporal forecasting.

I. INTRODUCTION

Network time-series forecasting is a critical problem with several applications, e.g., for financial predictions, for traffic prediction, and for forecasting the trends of rare diseases [1]–[3]. Making accurate predictions about the future is a challenging task, as both inter-series and intra-series dependencies need to be modeled simultaneously [4]. Recent work [5], [6] leveraged graph neural networks (GNNs) to model the inter-series relationships as a graph since GNNs can capture the symmetries in graph data [7]. However, most real-world networks innately comprise higher-order relations that go beyond pairwise connections [2], [8]. One way to model such higher-order relations is to use hypergraph generalizations of graphs [9], where a hyperedge can connect multiple nodes and thus can naturally express higher-order relations such as group behavior between multiple nodes [10].

In addition to higher-order correlations, several real-world temporal networks such as blockchain transaction networks, and stock networks exhibit a scale-free or a hierarchical structure [11], where a small change in a single network entity may cause a series of chain reactions leading to a “domino effect.” For example, changes in crude oil prices can lead to contagious effects on several groups of stocks across industries such as transportation, energy, utilities, etc. [12]. However, conventional (hyper)graph models are defined in the Euclidean space, and thus suffer from large distortions when representing such networks [13].

At the same time, typical characteristics of scale-free networks such as long-tail node degree distributions and the presence of influential hubs are handled well in the *hyperbolic* space [14]. However, existing hyperbolic graph neural networks [15] do not generalize to higher-order connections in hypergraphs.

With this in mind, here we leverage the hyperbolic space to encode scale-free temporal and time-evolving dynamic hypergraphs. In particular, we formulate the δ_{hg} hyperbolicity for hypergraphs (§II) based on the shortest path algorithms in hypergraphs and we build elegant distance-aware hyperbolic hypergraph aggregation operations (§III-B) that use hyperbolic distance to preserve the information during message propagation. We present a novel hyperbolic learning framework which we call *Temporal Hypergraph Hyperbolic Network* (THINK), which uses hyperbolic hypergraph and temporal operations (§III-C) to encode temporal, and time-evolving dynamic networks.

Our contributions can be summarized as:

- We formulate δ_{hg} hyperbolicity for hypergraphs, and we propose DHAN, a Distance-aware Hyperbolic Hypergraph Attention Network, to capture higher-order scale-free correlations in the hyperbolic space.
- We devise THINK, which combines hyperbolic temporal convolutions with DHAN in order to capture hyperbolic properties in the network and in the temporal domains.
- We show that THINK outperforms several state-of-the-art methods across seven tasks defined on spatio-temporal and dynamic networks. We further demonstrate THINK’s practicality on financial, health, and energy applications.

II. HYPERGRAPH HYPERBOLICITY & HYPERBOLIC SPACES

The δ_{hg} **Hyperbolicity** is a score that provides a degree of similarity of the hypergraph to the hyperbolic space. For a hypergraph $\mathcal{G}$, the δ_{hg} hyperbolicity can be computed using the Gromov product [16], which is defined as follows for a triple of nodes x , y , and z :

$$(y, z)_x = \frac{1}{2}(d(x, y) + d(x, z) - d(y, z)) \quad (1)$$

where the distance $d(x, y)$ is the shortest s -walk, called the s -distance between nodes x and y [17]. The s -walk is a higher-order walk in the hypergraph, where s controls the “width” in terms of edge overlap size. The s -walk between nodes x and y is a sequence of nodes that pairwise share at least s hyperedges (we provide the algorithm for s -distance in the Appendix).

^{*}Equal Contribution

TABLE I: Basic dataset statistics including δ_{hg} and δ_{rel} hyperbolicities of (i) *spatio-temporal networks* such as Chicken Pox spread (CPox), windmill energy (WMill), and stock networks of NYSE, NASDAQ, CSE, and TSE exchanges, and (ii) *dynamic networks* such as Twitter tennis mentions (DTT).

Dataset	# Timesteps	# Nodes	δ_{hg}	δ_{rel}
DTT [18]	120	1,000	1.0	-
CPox [19]	522	20	1.5	0.190
WMill [20]	17,472	319	1.0	0.025
NYSE [1]	1,245	1,737	0.5	0.087
TSE [21]	1,159	95	1.5	0.074
NASDAQ [2]	1,245	1,026	1.0	0.107
CSE [22]	1,293	85	1.5	0.176

We now define δ_{hg} as the minimal value greater than zero for which the following holds for any nodes x, y, z , and w :

$$(x, z)_w \geq \min((x, y)_w, (y, z)_w) - \delta_{\text{hg}} \quad (2)$$

In addition to δ_{hg} , we also compute the relative dataset-level δ_{rel} hyperbolicity using the Euclidean distance between temporal node features [23] (formally defined in the Appendix). Low values of δ_{hg} and δ_{rel} hyperbolicity indicate that the space has an underlying hyperbolic geometry, i.e., an approximate tree-like structure, and that the hyperbolic space would be well-suited to embed it [24]. Table I shows the hyperbolicity of various temporal and dynamic hypergraphs: we can see that the degree of hyperbolicity in these networks is generally high, i.e., low $\delta_{\text{hg}}, \delta_{\text{rel}}$, suggesting that scale-free hypergraphs can benefit from representations in the hyperbolic space.

The **hyperbolic space** is a non-Euclidean geometry with constant negative curvature. We implement the Poincaré ball model of the hyperbolic space. This model is defined as $(\mathcal{B}, g_x^{\mathcal{B}})$, where the manifold $\mathcal{B} = \{x \in \mathbb{R}^n : \|x\| < 1\}$ is endowed with the Riemannian metric $g_x^{\mathcal{B}} = \lambda_x^2 g^E$, where the conformal factor $\lambda_x = \frac{2}{1-\|x\|^2}$ and $g^E = \text{diag}[1, \dots, 1]$ is the Euclidean metric tensor. We denote the tangent space (Euclidean) centered at point x as $\mathcal{T}_x \mathcal{B}$. Since hyperbolic spaces are non-Euclidean, we leverage the formulations of Möbius gyrovector spaces, which allow us to generalize standard operations to hyperbolic geometry [15].

Möbius Addition & Hyperbolic Distance. Following [15], the Möbius addition $\oplus$ and the hyperbolic distance $d_{\mathcal{B}}(x, y)$ for points $x, y \in \mathcal{B}$ is given as follows:

$$x \oplus y = \frac{(1 + 2\langle x, y \rangle + \|y\|^2)x + (1 - \|x\|^2)y}{1 + 2\langle x, y \rangle + \|x\|^2\|y\|^2} \quad (3)$$

$$d_{\mathcal{B}}(x, y) = 2 \tanh^{-1}(\| -x \oplus y \|) \quad (4)$$

where, $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the inner product and the norm, respectively. We now define the mapping functions to project Euclidean vectors to the hyperbolic space, and vice versa.

Exponential & Logarithmic Map. For point $x \in \mathcal{B}$, the exponential map $\exp_x : \mathcal{T}_x \mathcal{B} \rightarrow \mathcal{B}$ and the logarithmic map $\log_x : \mathcal{B} \rightarrow \mathcal{T}_x \mathcal{B}$ for tangent vector v and point y are

$$\exp_x(v) = x \oplus \left(\tanh \left(\frac{\lambda_x \|v\|}{2} \right) \frac{v}{\|v\|} \right) \quad (5)$$

$$\log_x(y) = \frac{2}{\lambda_x} \tanh^{-1}(\| -x \oplus y \|) \frac{-x \oplus y}{\| -x \oplus y \|} \quad (6)$$

$$x \odot y = \tan \left(\frac{\|xy\|}{y} \arctan^{-1}(\|y\|) \right) \frac{\|xy\|}{\|y\|} \quad (7)$$

$$W \otimes x = \exp_o(W \log_o(x)) \quad (8)$$

Poincaré Fully Connected Layer ($\mathcal{F}^c(\cdot)$). Given the inputs $x \in \mathcal{B}^n$ with parameters $Z = \{z_k \in \mathcal{T}_o \mathcal{B}^n\}_{k=1}^m$ and bias terms $r = \{r_k \in \mathbb{R}\}_{k=1}^m$, we follow [25] to define $(\mathcal{F}^c(\cdot))$ as follows:

$$\mathcal{F}^c(x) = w(1 + \sqrt{1 + \|w\|^2})^{-1}, w = (\sinh(v_k(x)))_{k=1}^m \quad (9)$$

where $v_k(x)$ is the generalized linear transform defined as

$$v_k(x) = 2\|z_k\| \sinh^{-1}(\lambda_x \langle x, z_k \rangle \cosh(2r_k) - (\lambda_x - 1) \sinh(2r_k)) \quad (10)$$

Poincaré β -concatenation ($\beta\text{-cat}(\cdot)$). is used to join M inputs $\{x_i \in \mathcal{B}^{n_i}\}_{i=1}^M$ into a vector $y = \beta\text{-cat}(\{x_i\}_{i=1}^M) \in \mathcal{B}^n$, where $n = \sum_{i=1}^M n_i$. Following [25], the inputs are first scaled using coefficients $\beta_n = B(\frac{n}{2}, \frac{1}{2})$, where B denotes the beta distribution. These features are then concatenated in the tangent space as follows:

$$y = \exp_o(\beta_n \beta_{n1}^{-1} \log_o(x_1)^\top, \dots, \beta_n \beta_{nM}^{-1} \log_o(x_M)^\top)^\top \quad (11)$$

III. METHODOLOGY: THINK

Fig. 1 shows an overview of THINK. In the following subsections, we explain how we extract temporal features using hyperbolic temporal convolutions (§III-A). We then describe how we aggregate node features via distance-aware hyperbolic hypergraph aggregation (§III-B) followed by detailing how hyperbolic temporal and spatial components are combined to capture the temporal evolution of hypergraphs (§III-C).

A. Hyperbolic Temporal Convolution

The temporal dependencies of several common hypergraphs such as stock networks often show power-law distribution [26], which indicates the possible presence of hierarchical relations in time-series data. Thus, we implement a temporal convolution in the hyperbolic space $\tau\text{-conv}(\cdot)$ [25]. In contrast to the Euclidean space, the Poincaré ball model reflects the power law distribution with its radius [15], thus enabling it to better represent the hierarchical relations in a time-series data.

Given Euclidean temporal input node features X_E , we first project them to hyperbolic features X^l via the exponential mapping as $X^l = \exp_o(X_E)$. Let K denote the kernel size of the hyperbolic temporal convolution with input features $X^l \in \mathcal{B}^{N \times nK \times C}$ corresponding to N nodes for a historical lookback period $\tau = nK$ to the l^{th} layer. Here, C is the number of features per node. We concatenate the node features $\{x_{is} \in \mathcal{B}^{N \times C}\}_{s=1}^K$ in the receptive field of the kernel into features $x_i \in \mathcal{B}^{N \times CK}$ using a Poincaré β -concatenation, which is then operated on by a Poincaré fully connected layer as shown in Fig. 1. These transformations produce X^{l+1} :

$$X^{l+1} = \tau\text{-conv}(X^l) = \{\mathcal{F}^c(\beta\text{-cat}(\{x_{is}\}_{s=1}^K))\}_{i=1}^n \quad (12)$$

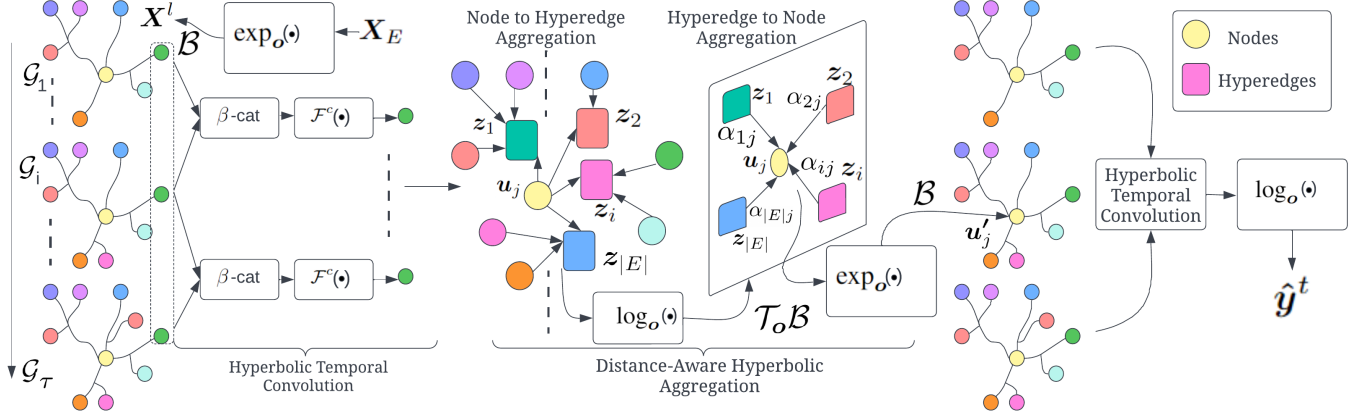


Fig. 1: Overview of our proposed THINK framework for distance-guided hyperbolic aggregation and interaction learning.

B. Distance-Aware Hyperbolic Aggregation

We use hypergraph convolutions in the hyperbolic rather than the Euclidean space [9]. As the hypergraph grows exponentially, we use the hyperbolic space, where we can obtain more robust embeddings [15]; in contrast, the polynomial growth of the graph volume in the Euclidean space leads to distortion in the embeddings for scale-free networks [14].

Next, we describe an attentive hyperbolic hypergraph convolution $\text{HA}(\cdot)$, which benefits from the expressiveness of both hypergraphs and hyperbolic embeddings [15]. We apply the attentive hyperbolic hypergraph convolution $\text{HA}(\cdot)$ on the hypergraph $\mathcal{G}(V, E)$, where $V = \{v_1, \dots, v_{|V|}\}$ is a set of nodes and $E = \{e_1, \dots, e_{|E|}\}$ is a set of hyperedges, and each hyperedge e_i connects a subset of nodes $\{v_j \in e_i\}$. As shown in Fig. 1, DHHAN is composed of two steps:

Node to Hyperedge Aggregation computes the hyperedge feature vector z_i by aggregating the hyperbolic features $\{u_k | v_k \in e_i\}$ of the nodes that belong to hyperedge e_i . In particular, we use the Einstein midpoint [27], which generalizes mean-based aggregation to the hyperbolic space:

$$z_i = \frac{1}{2} \otimes \left(\sum_{u_k | v_k \in e_i} \frac{\lambda_{u_k}}{\sum_{u_j | v_j \in e_i} (\lambda_{u_j} - 1)} u_k \right) \quad (13)$$

where, λ_u is the conformal factor, defined as $\lambda_u = \frac{2}{1 - \|u\|^2}$.

Hyperedge to Node Aggregation updates the hyperbolic node features u_j using the information from the hyperedges that contains the node v_j . We use self-attention aggregation to learn the varying importance of each relation $\{e_i | v_j \in e_i\}$ with respect to node v_j . We further develop a hyperbolic distance-aware self-attention mechanism that better preserves the hyperbolic properties while propagating information [28]. We use the hyperbolic distance $d_B(u_j, z_i)$ between features u_j and the features of hyperedges containing the node v_j , i.e., $\{z_i | v_j \in e_i\}$. The distance-guided attention learns the attention coefficient α_{ij} using the node's hyperbolic feature u_j and the aggregated hyperedge features $\{z_i | v_j \in e_i\}$:

$$\alpha_{ij} = \mathbf{a}^\top \otimes (u_j \oplus z_i) \odot d_B(u_j, z_i) \quad (14)$$

where $\cdot^\top$ denotes transposition and $\mathbf{a}$ is a trainable vector.

With the help of the attention coefficients α_{ij} , the hypergraph convolution $\text{HA}(\cdot)$ updates the hyperbolic node features $U = \{u_1, \dots, u_{|V|}\}$ to a new set of hyperbolic features $U' = \{u'_1, \dots, u'_{|V|}\}$, given by $U' = \text{HA}(\mathcal{G}, U)$, where

$$u'_j = \exp_o \left(\text{ReLU} \left(\sum_{i | v_j \in e_i} \alpha_{ij} \log_o(\mathcal{F}^c(z_i)) \right) \right) \quad (15)$$

We now generalize the above hyperbolic hypergraph convolution $\text{HA}(\cdot)$ to *time-varying input features* $Q^l \in \mathcal{B}^{\tau \times N \times C}$ and *time-evolving dynamic hypergraphs* $G = \{\mathcal{G}_i\}_{i=1}^\tau$ and denote it by $\text{HA}^\tau(\cdot)$. The generalized attentive hyperbolic hypergraph convolution layer $\text{HA}^\tau(\cdot)$ applies the same $\text{HA}(\cdot)$ layer to each snapshot $\mathcal{G}_i$ of the input and outputs features Q^{l+1} given by

$$Q^{l+1} = \text{HA}^\tau(Q^l, G) = \{\text{HA}(\mathcal{G}_t, Q_t^l \in \mathcal{B}^{N \times C})\}_{t=1}^\tau \quad (16)$$

C. THINK: End-to-End Framework

We position the DHHAN between two hyperbolic temporal convolutions. This design choice allows the propagation of spatially updated features along the time axis through temporal convolutions. As shown in Fig. 1, we operate a hyperbolic temporal convolution on features X_E followed by DHHAN and another hyperbolic temporal convolution that produces final outputs $\hat{y}^t = \text{THINK}(X_E, G)$, given by

$$\hat{y}^t = \log_o(\tau\text{-conv}((\text{HA}^\tau(\tau\text{-conv}(\exp_o(X_E))), G))) \quad (17)$$

IV. EXPERIMENTS

A. Spatio-temporal & Dynamic Hypergraph Tasks

In node regression, we aim to forecast each node's future value for a single time-step. We evaluate THINK on various node regression problems: wind energy forecasting [20], county-level chickenpox cases prediction [19], [20], and risk forecasting [35] on the Chinese stock exchange (CSE). We further apply THINK on spatio-temporal classification problems, which predict each node's single time-step future trend. We evaluate THINK on stock movement classification [2] (on NASDAQ) with targets being stock prices going up, going down, or staying neutral.

TABLE II: Performance comparison to baselines on spatio-temporal and time-evolving dynamic hypergraphs (mean of 25 runs). **Purple & pink** show the best and the second best results, respectively, and Clf denotes stock movement classification on NASDAQ. * indicates significant ($p < 0.01$) improvements over state-of-the-art methods under Wilcoxon’s Signed Rank Test.

Dataset	DTT	CPox	WMill	Risk	NYSE		TSE		Clf
δ_{hg}	1.0	1.5	1.0	1.5	0.5		1.5		1.0
δ_{rel}	-	0.190	0.025	0.176	0.087		0.074		0.107
Model	MSE($\downarrow$)	MSE($\downarrow$)	MSE($\downarrow$)	MSE($\downarrow$)	SR($\uparrow$)	NDCG($\uparrow$)	SR($\uparrow$)	NDCG($\uparrow$)	F1($\uparrow$)
GConvGRU [29]	$2.05 \pm 5e^{-3}$	$1.14 \pm 9e^{-4}$	$1.38 \pm 3e^{-3}$	$0.61 \pm 6e^{-3}$	$0.89 \pm 2e^{-3}$	$0.57 \pm 4e^{-3}$	$0.81 \pm 9e^{-4}$	$0.55 \pm 1e^{-3}$	$0.25 \pm 2e^{-4}$
EGCN-O [30]	$2.06 \pm 5e^{-3}$	$1.12 \pm 4e^{-3}$	$1.36 \pm 9e^{-4}$	$0.57 \pm 8e^{-4}$	$0.92 \pm 4e^{-3}$	$0.59 \pm 6e^{-3}$	$0.83 \pm 1e^{-3}$	$0.57 \pm 5e^{-3}$	$0.28 \pm 3e^{-3}$
DCRNN [31]	$2.05 \pm 4e^{-3}$	$1.12 \pm 6e^{-3}$	$1.28 \pm 4e^{-3}$	$0.45 \pm 2e^{-3}$	$0.98 \pm 5e^{-3}$	$0.69 \pm 9e^{-4}$	$0.92 \pm 3e^{-3}$	$0.64 \pm 8e^{-4}$	$0.32 \pm 2e^{-3}$
TGCN [32]	$2.04 \pm 1e^{-3}$	$1.11 \pm 2e^{-3}$	$1.27 \pm 5e^{-3}$	$0.43 \pm 3e^{-3}$	$1.02 \pm 1e^{-3}$	$0.71 \pm 6e^{-3}$	$0.93 \pm 2e^{-3}$	$0.67 \pm 1e^{-3}$	$0.34 \pm 4e^{-3}$
ST-TGCN [33]	$2.04 \pm 3e^{-3}$	$1.12 \pm 1e^{-3}$	$1.24 \pm 2e^{-3}$	$0.37 \pm 1e^{-3}$	$1.04 \pm 2e^{-3}$	$0.70 \pm 3e^{-3}$	$0.96 \pm 8e^{-4}$	$0.71 \pm 6e^{-3}$	$0.37 \pm 4e^{-3}$
DyGrAE [34]	$2.03 \pm 1e^{-3}$	$1.12 \pm 6e^{-3}$	$1.24 \pm 3e^{-3}$	$0.40 \pm 2e^{-3}$	$1.03 \pm 8e^{-4}$	$0.72 \pm 4e^{-3}$	$0.95 \pm 3e^{-3}$	$0.69 \pm 5e^{-3}$	$0.36 \pm 2e^{-3}$
RSR-I [1]	-	$1.13 \pm 2e^{-3}$	$1.23 \pm 5e^{-3}$	$0.39 \pm 4e^{-3}$	$1.05 \pm 1e^{-3}$	$0.75 \pm 6e^{-3}$	$0.99 \pm 1e^{-3}$	$0.72 \pm 9e^{-4}$	$0.38 \pm 6e^{-3}$
EGCN-H [30]	$2.04 \pm 4e^{-3}$	$1.11 \pm 9e^{-4}$	$1.21 \pm 2e^{-3}$	$0.39 \pm 8e^{-2}$	$1.03 \pm 3e^{-3}$	$0.72 \pm 6e^{-3}$	$0.96 \pm 1e^{-3}$	$0.71 \pm 2e^{-3}$	$0.37 \pm 4e^{-3}$
STHGCN [2]	$1.03 \pm 6e^{-3}$	$1.11 \pm 2e^{-3}$	$1.19 \pm 1e^{-3}$	$0.37 \pm 4e^{-3}$	$1.10 \pm 3e^{-3}$	$0.78 \pm 6e^{-3}$	$1.07 \pm 2e^{-3}$	$0.74 \pm 5e^{-3}$	$0.40 \pm 1e^{-3}$
TCONV + DHAN	$0.61 \pm 3e^{-3}^*$	$1.10 \pm 6e^{-3}$	$1.08 \pm 2e^{-3}^*$	$0.34 \pm 3e^{-3}^*$	$1.14 \pm 7e^{-3}^*$	$0.81 \pm 1e^{-3}^*$	$1.11 \pm 2e^{-3}^*$	$0.76 \pm 5e^{-3}$	$0.44 \pm 8e^{-3}^*$
THINK	$0.58 \pm 5e^{-3}^*$	$1.09 \pm 4e^{-3}^*$	$1.05 \pm 2e^{-3}^*$	$0.32 \pm 2e^{-3}^*$	$1.18 \pm 4e^{-3}^*$	$0.86 \pm 9e^{-4}^*$	$1.19 \pm 6e^{-3}^*$	$0.81 \pm 7e^{-4}^*$	$0.49 \pm 4e^{-3}^*$

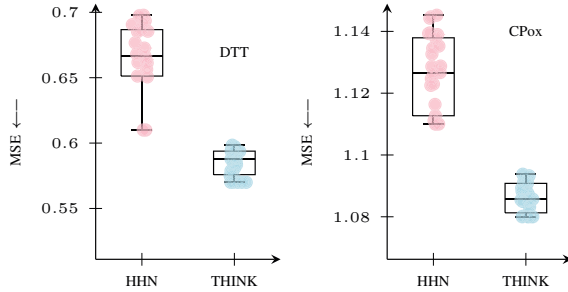


Fig. 2: Performance distribution with confidence intervals (over 25 runs) of hyperbolic hypergraph network (HHN, first, pink), and THINK (second, blue). We use the same hyperbolic temporal convolution for DTT and CPox datasets.

Finally, we apply THINK on stock ranking [2] over the Tokyo Stock Exchange (TSE) [21] and the New York Stock Exchange (NYSE) [1] markets. Following [1], we formulate stock prediction as a ranking problem, where our target is to learn a ranking function that maps a set of stocks to a ranking list. In the learned ranking list, stocks with higher ranking scores are expected to yield higher profits.

B. Evaluation Measures

We evaluate risk prediction, chickenpox cases forecasting, and windmill power via Mean Squared Error (MSE). We evaluate THINK on node classification using F1-score. We evaluate THINK’s profitability and ranking ability using Sharpe Ratio (SR) and Normalized Discounted Cumulative gain (NDCG). The Sharpe Ratio is a measure of the return R_a in excess of a risk-free return R_f , given by $SR = \frac{E[R_a - R_f]}{std[R_a - R_f]}$. Following [1], we adopt a daily-buy-hold trading strategy in which, on trading day t , we acquire a ranked list of stocks based on the predicted return ratio for every stock. From this list, we buy the top- k stocks, which are sold at the closing market price of the following day $t + 1$.¹

¹We release our code at: <https://github.com/shivamag125/ICDM22-THINK>

V. RESULTS AND ANALYSIS

A. Performance Comparison

We evaluate THINK on various spatio-temporal and dynamic hypergraph problems in Table II, and we observe that THINK is the new state of the art across most datasets, and that it can generalize to various downstream applications spanning across time-evolving dynamic and spatio-temporal hypergraphs. This improvement is due to two aspects: (i) hypergraph learning and (ii) hyperbolic geometry. First, THINK captures higher-order relations via hypergraphs instead of constraining them as pairwise edges in ordinary graphs (RSR-I, EGCN-H). Second, spatio-temporal learning can significantly benefit from hyperbolic geometry, especially in datasets that exhibit scale-free nature [15]. A common limitation of spatio-temporal (hyper)graph approaches (STHGCN, EGCN-H) is that they use the Euclidean space to encode scale-free properties of spatio-temporal and time-evolving dynamic networks, which leads to high distortion in their learned representations [14].

Impact of Hyperbolic Temporal Convolution. In order to further quantify the improvements due to hyperbolic learning in the temporal domain, we compare it to Euclidean learning in Table II. Specifically, we replace the hyperbolic temporal convolution with a Euclidean temporal convolution [3]. We observe significant ($p < 0.01$) improvements when using hyperbolic learning for representing the time-series data. This improvement empirically validates that hyperbolic learning in the temporal front equips THINK with geometrically appropriate inductive biases for better representing the power law dynamics of temporal data.

B. Impact of Distance-Aware Self-Attention

In order to contextualize the improvements from our distance-guided self-attention hyperbolic message propagation, we contrast THINK to a version without the enhancement of distance attention (HHN). The results are shown in Fig. 2, where we can see that the model without the enhancement of distance attention reduces the overall performance on all datasets and also increases the variance.

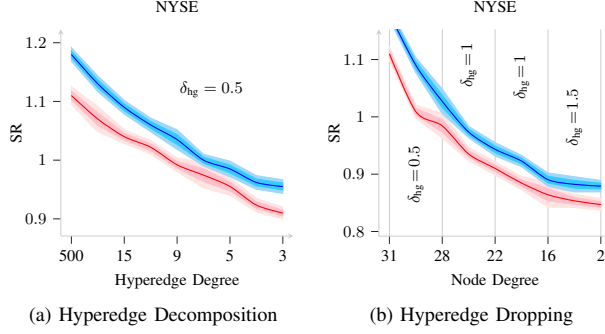


Fig. 3: Performance variation with (a) successive hyperedge decomposition and (b) hyperedge removal with error bounds (over 25 independent runs). The blue (top curve) and red (bottom curve) indicate THINK and Euclidean THINK (Euclidean temporal convolution + hypergraph attention), respectively.

These observations suggest that thanks to its hyperbolic distance-guided message propagation THINK can better capture the impact of a node on the overall representation of the group. Moreover, these observations tie up with those of [28], who showed that distance-aware aggregations preserve hyperbolic message aggregations in the tangent space using hyperbolic distances.

C. Impact of Hypergraph & Hyperbolic Learning

Impact of Hypergraph Learning. We compare the performance of representing relations as hyperedges vs. as ordinary pairwise edges. We decompose each hyperedge of degree n into $\binom{n}{2}$ pairwise edges in increasing order of hyperedge degree, and we analyze the performance variation as we decompose hyperedges in Fig. 3a.² We observe poorer performance as we decompose hyperedges into pairwise edges since decomposing hyperedges induces noise in the network. The worst performance is achieved when all hyperedges are decomposed, which is when THINK degenerates to hyperbolic graph attention model. Through this experiment, we note that hypergraphs effectively capture higher-order relations between nodes as opposed to simple graphs.

Impact of Hyperbolic Learning We probe the impact of highly influential nodes (hubs) and the effectiveness of domain knowledge on THINK's performance. We identify hubs of the scale-free network by sorting the nodes in decreasing order of their degree and we successively remove the corresponding hubs' hyperedges. Then, we compare THINK to its Euclidean variant in Fig. 3b.² We observe poorer performance for both models as we remove edges, and they perform the worst after all hyperedges are removed, which essentially degenerates THINK to a temporal model. Interestingly, we note that as we isolate the most influential hubs, THINK's performance drops (larger drops in hypergraphs with low δ_{hg}), likely because it is unable to incorporate their strong impact on other nodes and the network becomes less hyperbolic.

²Note that we observe similar trends in other datasets.

VI. CONCLUSION

Building on our δ_{hg} hyperbolicity formulation for hypergraphs, we introduced an elegant distance-guided attentive hyperbolic hypergraph neighborhood aggregation mechanism (DHHAN), which better captures higher-order relations in the hyperbolic space. Then, using DHHAN as a building block, we developed an end-to-end framework, THINK, which blends hyperbolic temporal convolutions with spatial DHHAN. Our experiments on seven downstream tasks demonstrated THINK's effectiveness over dynamic time-evolving and spatio-temporal hypergraph networks in comparison to various benchmarks.

ACKNOWLEDGMENT

This research was supported in part by the US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329.

REFERENCES

- [1] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Transactions on Information Systems*, 2019.
- [2] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah, "Spatiotemporal hypergraph convolution network for stock movement forecasting," in *2020 IEEE International Conference on Data Mining*, 2020.
- [3] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *AAAI Conference on Artificial Intelligence*, Jul. 2019.
- [4] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Spectral temporal graph neural network for multivariate time-series forecasting," in *Advances in Neural Information Processing Systems NeurIPS 2020*, 2020.
- [5] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *International Conference on Knowledge Discovery & Data Mining*, 2019.
- [6] D. Xu, C. Ruan, E. Korpoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *International Conference on Learning Representations*, 2020.
- [7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.
- [8] J. Yi and J. Park, "Hypergraph convolutional recurrent neural network," in *International Conference on Knowledge Discovery*, 2020.
- [9] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, 2021.
- [10] R. Zhang, Y. Zou, and J. Ma, "Hyper-sagmn: a self-attention based graph neural network for hypergraphs," in *International Conference on Learning Representations*, 2019.
- [11] S. Battiston, J. B. Glattfelder, D. Garlaschelli, F. Lillo, and G. Caldarelli, "The structure of financial networks," in *Network Science*, 2010.
- [12] S. Fang and P. Egan, "Measuring contagion effects between crude oil and chinese stock market sectors," *The Quarterly Review of Economics and Finance*, 2018.
- [13] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, "On the hyperbolicity of small-world and tree-like random graphs," *arXiv:1201.1717*, 2013.
- [14] F. Sala, C. De Sa, A. Gu, and C. Re, "Representation tradeoffs for hyperbolic embeddings," in *Machine Learning Research*, 2018.
- [15] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Advances in Neural Information Processing Systems* 32, 2019.
- [16] M. Gromov, "Hyperbolic groups," in *Essays in Group Theory*. Springer New York, 1987.

- [17] L. Lu and X. Peng, “High-ordered random walks and generalized laplacians on hypergraphs,” in *International Workshop on Algorithms and Models for the Web-Graph*, 2011.
- [18] F. Béres, D. M. Kelen, R. Pálócs, and A. A. Benczúr, “Node embeddings in dynamic graphs,” *Applied Network Science*, Aug. 2019.
- [19] B. Rozemberczki, P. Scherer, O. Kiss, R. Sarkar, and T. Ferenci, “Chickenpox cases in Hungary: a benchmark dataset for spatiotemporal signal processing with graph neural networks,” *arXiv:2102.08100*, 2021.
- [20] B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, M. Astefanoaei, O. Kiss, F. Beres, N. Collignon, and R. Sarkar, “PyTorch geometric temporal: Spatiotemporal signal processing with neural machine learning models,” *arXiv:2104.07788*, 2021.
- [21] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, “Modeling the stock relation with graph network for overnight stock movement prediction,” in *International Joint Conference on Artificial Intelligence*, 2020.
- [22] J. Huang, Y. Zhang, J. Zhang, and X. Zhang, “A tensor-based sub-mode coordinate algorithm for stock prediction,” in *IEEE Conference on Data Science in Cyberspace*, 2018.
- [23] M. Borassi, A. Chessa, and G. Caldarelli, “Hyperbolicity measures democracy in real-world networks,” *Physical Review E*, 2015.
- [24] A. Tifrea, G. Becigneul, and O.-E. Ganea, “Poincare GloCe: Hyperbolic word embeddings,” in *International Conference on Learning Representations*, 2019.
- [25] R. Shimizu, Y. Mukuta, and T. Harada, “Hyperbolic neural networks++,” in *International Conference on Learning Representations*, 2021.
- [26] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, “A theory of power-law distributions in financial market fluctuations,” *Nature*, 2003.
- [27] A. A. Ungar, *Analytic hyperbolic geometry: Mathematical foundations and applications*, 2005.
- [28] S. Zhu, S. Pan, C. Zhou, J. Wu, Y. Cao, and B. Wang, “Graph geometry interaction learning,” in *Advances in Neural Information Processing Systems*, 2020.
- [29] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” in *International Conference on Neural Information Processing*, 2018.
- [30] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, “EvolveGCN: Evolving graph convolutional networks for dynamic graphs,” in *AAAI Conference on Artificial Intelligence*, 2020.
- [31] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *International Conference on Learning Representations*, 2018.
- [32] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, “T-GCN: A temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [33] X. Xu, T. Zhang, C. Xu, Z. Cui, and J. Yang, “Spatial-temporal tensor graph convolutional network for traffic prediction,” *CoRR*, 2021.
- [34] A. Taheri and T. Berger-Wolf, “Predictive temporal embedding of dynamic graphs,” in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.
- [35] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith, “Predicting risk from financial reports with regression,” in *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.
- [36] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, “Hyperbolic image embeddings,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [37] X. Sun, H. Yin, B. Liu, H. Chen, J. Cao, Y. Shao, and N. Q. Viet Hung, “Heterogeneous hypergraph embedding for graph classification,” in *ACM International Conference on Web Search and Data Mining*, 2021.
- [38] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, “Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach,” *AAAI Conference on Artificial Intelligence*, 2021.

APPENDIX

A. Dataset-Level δ_{rel} Hyperbolicity Estimation

Following [24], we use the Gromov product [16] to estimate the dataset-level hyperbolicity. Let us define $\mathbf{x}, \mathbf{y}, \mathbf{z} \in W$ as the temporal features, where W is the metric space.

Algorithm 1: s -walk-distance

Input : Incidence matrix $\mathbf{H}^{|V| \times |E|}$, int s , source S , target T

Result: s -distance between S and T

initialize $s_adj^{|V| \times |V|} = 0$;

```

for each vertex  $v \in V$  do
     $e = \mathbf{H}[v]$ ;
    for each vertex  $v' \in V$  do
         $ne = \mathbf{H}[v']$ ;
        if  $\text{length}(e \cap ne) \geq s$  then
             $s\_adj[v][v'] = 1$ ;
        end
    end
end

```

end

$\mathcal{G} = \text{create_graph_from_adjacency}(s_adj)$;

return $\text{Dijkstra}(\mathcal{G}, S)[T]$

The Gromov product for features x and y is given by

$$(\mathbf{y}, \mathbf{z})_{\mathbf{x}} = \frac{1}{2}(l(\mathbf{x}, \mathbf{y}) + l(\mathbf{x}, \mathbf{z}) - l(\mathbf{y}, \mathbf{z})) \quad (18)$$

where $l(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.

We define the δ hyperbolicity as the smallest non-negative value such that the following holds:

$$(\mathbf{x}, \mathbf{z})_{\mathbf{w}} \geq \min((\mathbf{x}, \mathbf{y})_{\mathbf{w}}, (\mathbf{y}, \mathbf{z})_{\mathbf{w}}) - \delta. \quad (19)$$

Following [36], we use the scale-invariant δ_{rel} hyperbolicity, given by $\delta_{rel} = \frac{2\delta}{\text{diam}(W)}$, where $\text{diam}(W)$ is the largest pairwise distance (diameter) for the metric space W . By definition, $\delta_{rel} \in [0, 1]$ and specifies how close a dataset is to a hyperbolic space. A low δ_{rel} hyperbolicity (i.e., close to 0) for a network indicates that it has an underlying hyperbolic geometry.

B. Hypergraph Construction

We construct a hypergraph $G = (V, E)$, where each vertex $v \in V$, and each hyperedge $e \in E$ is a subset of related nodes. **DTT, CPox, WMill** Following [37], we created hyperedges based on the neighborhood of each node. For each node v , we first found the neighbors $N(v)$ and we created a set $N = \{(v, N(v)) | v \in V\}$. Then, we merged pairs of elements $(v_i, N(v_i)), (v_j, N(v_j))$ based on their similarity which is calculated using Sørensen-Dice coefficient (SCD). We calculated SCD for every pair of elements and we merged them until no two pairs had an SCD score lower than a threshold.

Stock Datasets: NYSE, NASDAQ, TSE and CSE Following [38] we constructed hyperedges between stocks based on: (i) industry hyperedges and (i) Wiki corporate hyperedges. The former connect stocks belonging to the same industry, while the latter consist of first- and second-order corporate relationships between stocks. The *first-order relation* is defined as $X \xrightarrow{R1} Y$, where $R1$ represents the entity-relation between stocks X, Y . We constructed a hyperedge of a source stock and a set of target stocks related to it via the same Wikidata relation. The *second-order relation* is pairwise in nature and defined as $X \xrightarrow{R2} Z \xleftarrow{R3} Y$, where Z denotes an entity connecting X and Y via entity-relations $R2$, and $R3$.