

# Pioneer: Physics-informed Riemannian Graph ODE for Entropy-increasing Dynamics

Li Sun<sup>1\*</sup>, Ziheng Zhang<sup>1</sup>, Zixi Wang<sup>1</sup>, Yujie Wang<sup>1</sup>, Qiqi Wan<sup>1</sup>, Hao Li<sup>1</sup>, Hao Peng<sup>2</sup>, Philip S. Yu<sup>3</sup>

<sup>1</sup>North China Electric Power University, Beijing 102206, China

<sup>2</sup>Beihang University, Beijing 100191, China

<sup>3</sup>Department of Computer Science, University of Illinois at Chicago, IL 60607, USA  
ccesunli@ncepu.edu.cn; zhangzh@ncepu.edu.cn; psyu@uic.edu

## Abstract

Dynamic interacting system modeling is important for understanding and simulating real world systems. The system is typically described as a graph, where multiple objects dynamically interact with each other and evolve over time. In recent years, graph Ordinary Differential Equations (ODE) receive increasing research attentions. While achieving encouraging results, existing solutions prioritize the traditional Euclidean space, and neglect the intrinsic geometry of the system and physics laws, e.g., the principle of entropy increasing. The limitations above motivate us to rethink the system dynamics from a fresh perspective of Riemannian geometry, and pose a more realistic problem of physics-informed dynamic system modeling, considering the underlying geometry and physics law for the first time. In this paper, we present a novel physics-informed Riemannian graph ODE for a wide range of entropy-increasing dynamic systems (termed as Pioneer). In particular, we formulate a differential system on the Riemannian manifold, where a manifold-valued graph ODE is governed by the proposed constrained Ricci flow, and a manifold preserving Gyro-transform aware of system geometry. Theoretically, we report the provable entropy non-decreasing of our formulation, obeying the physics laws. Empirical results show the superiority of Pioneer on real datasets.

Code — <https://github.com/nakks2/Pioneer>

## Introduction

The dynamic interacting systems are ubiquitous in the real world, ranging from meteorology and the spread of COVID (Huang et al. 2024) to social networks (Luo et al. 2023). Dynamic system modeling is important for understanding and simulating the real-world systems. In the literature, Ordinary Differential Equations (ODEs) (Chen et al. 2018) show success in modeling continuous evolvement. Given that these interacting systems are naturally represented as graphs, Graph Neural Networks (GNNs) (Velickovic et al. 2018) are frequently employed alongside other methods to incorporate the intercorrelations. Recently, graph ODEs are introduced and achieve encouraging results (Huang, Sun, and Wang 2021). However, there are several important issues largely remaining open.

\*Corresponding Author: Li Sun

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

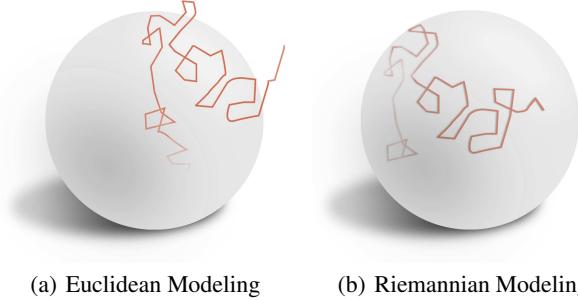


Figure 1: Illustrated example on Earth’s system. Traditional Euclidean models expose to generate unrealistic trajectory as in (a), while we preserve the manifold geometry as in (b).

On the one hand, previous studies prioritize the traditional Euclidean space, and neglect the intrinsic **geometry** of real systems. We illustrate the importance of geometry awareness in Figure 1. Unaware of the geometry, traditional Euclidean models generate trajectories that extend beyond the Earth’s surface, which is unreasonable and unrealistic. Riemannian geometry provides an elegant framework for complex systems (Sun et al. 2022b). The Earth is a 2D sphere and does not have any isometric alternative in Euclidean space of any dimension (O’shea 2009). Unfortunately, Riemannian manifold has not yet been introduced to model dynamic interacting system, to the best of our knowledge.

On the other hand, real systems obey the **physics laws**, e.g., the entropy-increasing principle of thermodynamics (Mackey 1989). In contrast, existing solutions largely overlook such physics laws in dynamic modeling, and thus the generated states are not aligned with the physics law on real-world datasets, as investigated in our Experiment. In recent years, physics-informed neural nets (Liu and Rahnemoonfar 2024; Horie and Mitsume 2022) are to embed physics laws to the neural model, but they primarily focus on solving Partial Differential Equations (PDEs), and cannot use for regression or dynamical prediction yet. In Riemannian geometry, Ricci flow is off the shelf to describe system evolution (Bai et al. 2021; Ollivier 2007), but it is not aligned with the entropy-increasing principle (Baptista, MacArthur, and Banerji 2024). So far, it still lacks a principled way to incorporate the physics law with the dynamic models.

That is, **existing solutions expose to the danger of generating geometry-unaware or anti-physics trajectories**. Such limitations result in unreasonable prediction contrary to the fact, preventing the wide use in real scenarios, and motivate us to rethink the dynamics modeling of real-world systems. We argue that the success of dynamics modeling is characterized not only by prediction accuracy comparing to the truth, but also by the reasonable prediction in line with physics laws. Thus, we pose a more realistic yet challenging problem of *physics-informed dynamic system modeling*.

We approach the dynamic system modeling from a fresh perspective using Riemannian geometry, and propose a novel Physics-informed Riemannian graph ODE for the ubiquitous entropy-increasing dynamic systems, named **Pioneer**. Given trajectory observations, we model the underlying dynamics of the objects' co-evolvement, where the correlation among objects tends to evolve over time as well. The key novelty is that, in light of aforementioned issues, **Pioneer** is also designed to generate reasonable outputs, ensuring that the predicted trajectories align with the system's geometry (such as the Earth's surface) and that the system states over time adhere to physical principles. Specifically, **Pioneer** is a differential system on the Riemannian manifold, where a **Manifold ODE** is introduced to model the continuous co-evolvement with a manifold-valued GNN. Informed of physics laws, a **constrained Ricci flow** is formulated to study the continuous changes of object correlation. In particular, we design a neural model over object features to constrain the canonical Ricci flow so that, *theoretically, the proposed flow is proved to be entropy non-decreasing* (Theorem 3.1). Being aware of the geometry, we propose **Gyro-transform** for encoding/decoding process, so that the generated trajectories lie on the manifold. Concretely, *Gyro-transform conducts manifold-preserving transformation by simple matrix-vector multiplication* (Theorem 3.2), and no longer uses the expensive exponential/logarithmic maps. Finally, both accuracy and rationale are evaluated on the real datasets.

Overall, key contributions are summarized as follows:

- We rethink system dynamics over the graph, and pose *a more realistic problem of physics-informed system dynamics*, considering underlying geometry and physics law for the first time, to the best of our knowledge.
- We propose a physics-informed ordinary differential system on the Riemannian manifold (**Pioneer**), where we formulate a constrained Ricci flow, in line with physics law, and manifold-preserving Gyro-transform.
- Extensive empirical results show **Pioneer** outperforms the state-of-the-art methods on several real datasets, and we analyze the geometry and entropy of real systems.

## Preliminaries and Notations

We formally review basic notions of Riemannian geometry and thermodynamics, specify the limitations of prior solutions, and thereby put forward a more realistic yet challenging problem of *physics-informed dynamic system modeling*.

**Riemannian Manifold** Riemannian geometry provides a systematic approach to study the complex structures. In Riemannian geometry, a graph is typically related to some Riemannian manifold, i.e., a smooth manifold  $\mathcal{M}$  endowed with Riemannian metric  $g$ . Each point  $x$  in the manifold is associated with a tangent space  $T_x\mathcal{M}$ , where Riemannian metric is defined. Mapping between the tangent space and manifold is given by the exponential map  $\text{Exp}_x : T_x\mathcal{M} \rightarrow \mathcal{M}$  and logarithmic map  $\text{Log}_x : \mathcal{M} \rightarrow T_x\mathcal{M}$ . There exist three types of isotropic manifolds characterized by the constant curvature  $\kappa$ : hyperbolic space of  $\kappa < 0$ , hyperspherical space of  $\kappa > 0$ , and Euclidean space of  $\kappa = 0$  as a special case.

**Thermodynamics and Ricci Flow** In physics, a thermodynamic system is described as a differential heat equation, which is equivalently given by Ricci flow from geometric perspective (Khan 2022). Bai et al. (2021) define the canonical *Ricci flow* on the graph, where the system dynamics is manifested in the evolvement of correlations among nodes, i.e., the weight function over time on the edges  $w_{ij}(t)$ . Given a weighted graph  $G(\mathcal{V}, \mathcal{E}, w(t))$ , the Ricci flow is written as

$$\frac{d}{dt}w_{ij}(t) = -R(i, j)w_{ij}(t), \quad (1)$$

where  $\mathcal{V}$  and  $\mathcal{E}$  denote the node set and edge set, respectively. *Ricci curvature*  $R(i, j)$  in Riemannian geometry characterizes the local geometry on the edge, and is defined by the *Wasserstein metric*  $W^1$  of measure space regarding the manifold. There exist two classical discretizations: Ollivier (2007) and Forman (2003). Ollivier's method explicitly calculates the  $W^1$  metric where the nested linear programming prevents gradient backpropagation, and thus is not compatible with our deep learning task. Instead, we employ Forman's version which studies  $W^1$  metric implicitly. Concretely, a family of Forman-Ricci curvature  $R_{ij}^F$  is given with some weighting function on the node. In this paper, we utilize the unity weight for simplicity, and the curvature of the edge  $(i, j) \in \mathcal{E}$  takes the form as follows,

$$R_{ij}^F = 2 - \left( \sum_{u \sim i} \sqrt{\frac{w_{ij}}{w_{iu}}} + \sum_{v \sim j} \sqrt{\frac{w_{ij}}{w_{jv}}} \right), \quad (2)$$

where  $u$  enumerates the set of vertices that are connected to vertex  $i$  (excluding vertex  $j$ ), and same as  $v$ .

**Dynamic System and Graph ODEs** We consider a dynamic system of  $N$  interacting objects. The observations consist of objects' trajectories  $\mathcal{X} = \{\mathbf{X}^1, \dots, \mathbf{X}^{T_{obs}}\}$  and the corresponding interaction graph  $\mathcal{G} = \{\mathbf{G}^1, \dots, \mathbf{G}^{T_{obs}}\}$  which evolves continuously. The system snapshot at timestamp  $t$  is given as  $(\mathbf{A}^t, \mathbf{X}^t)$ , where  $\mathbf{A}^t(i, j) = w_{ij}^t$  is the weight of the edge between objects  $i$  and  $j$ , and  $\mathbf{X}^t \in \mathbb{R}^{N \times F}$  is the feature matrix of  $N$  objects. In practice, the time intervals are typically nonuniform/irregular. ODEs show superiority to model the continuous dynamics, compared to the recurrent models with regular inputs (Chen et al. 2018). Concretely, the continuous co-evolvement over time is described as the ODE with dependent variables of  $N$  objects. The velocity of object is  $\frac{d\mathbf{z}_i^t}{dt} = f(\mathbf{z}_1^t, \dots, \mathbf{z}_N^t)$ , where

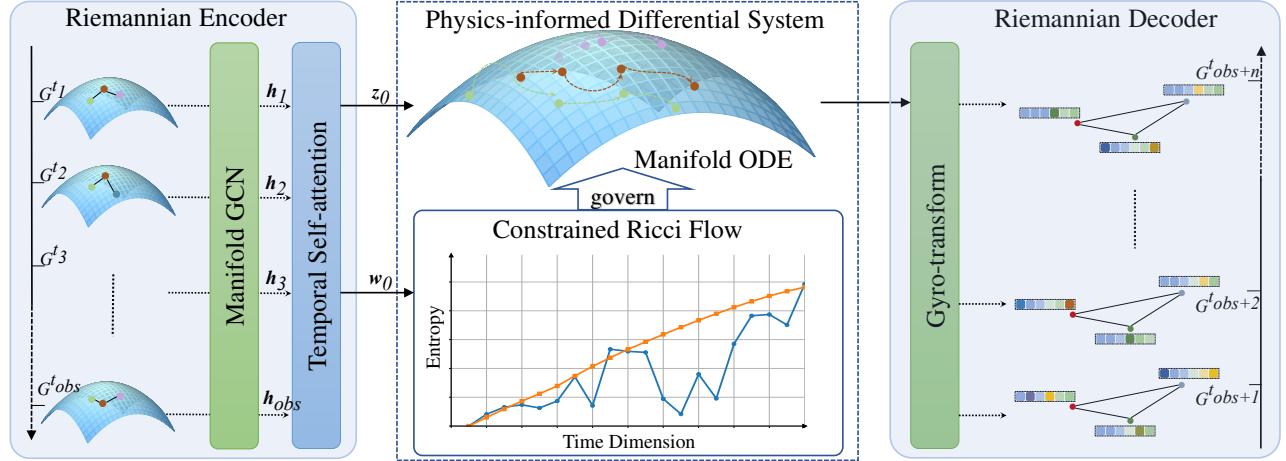


Figure 2: Overall architecture of **Pioneer**.

$z$  denotes the coordinate. Given the initial state  $z^0$ , the entire trajectory is given by  $z^T = z^0 + \int_{t=0}^T f(z_1^t, \dots, z_N^t)$ . Various frameworks with graph ODE have been introduced (Huang, Sun, and Wang 2021; Huang et al. 2024; Luo et al. 2023). Surprisingly, we find that *existing solutions expose to the danger of generating geometry-unaware or anti-physics trajectories*, e.g., generating trajectories extend beyond the earth surface. Such limitations motivate us to rethink dynamics modeling of real-world systems and pose the problem of physics-informed dynamic system modeling. In this paper, we focus on the entropy-increasing dynamic system, which is ubiquitous in the real world.

**Problem (Physics-informed Dynamic System Modeling).** *Given system observations  $\{\mathcal{X}, \mathcal{G}\}$ , our goal is to learn a neural simulator on the underlying geometry  $\mathcal{M}$  of the system, which not only captures the system dynamics by predicting future trajectories  $X^{T_{obs}+1:T}$ , but also adheres to the physics law of entropy-increasing.*

## Methodology: Pioneer

We propose a novel physics-informed Riemannian graph ODE for the ubiquitous entropy-increasing dynamic system (**Pioneer**), and our novelty lies in that *the generated prediction of Pioneer is aligned with the geometry and physics law of real system*. In Pioneer, we present an ODE system on the Riemannian manifold, where a *Constrained Ricci Flow* is formulated to control the continuous changes of system states in line with entropy-increasing principle, and we introduce a *Gyro-transform* for manifold-preserving encoding/decoding, so that the generated trajectories lie on the manifold. The overall architecture is illustrated in Fig. 2. Before elaborating on each component, we introduce the Riemannian manifold in which we build our model.

**$\kappa$ -sterographical model** The proposed Pioneer is applicable to any geodesically complete manifold (e.g., hyperbolic space and hyperspherical space) where the closed-form logarithmic map is defined. We opt for the  $\kappa$ -sterographical model as it provides the gyrovector formalism to unify con-

stant curvatures of positive, negative and zero values. Specifically, for any constant curvature  $\kappa$  and a dimension  $d \geq 2$ ,  $\kappa$ -*stereographic model* is defined on the smooth manifold of  $\mathfrak{S}_\kappa^d = \{x \in \mathbb{R}^d \mid -\kappa\|x\|^2 < 1\}$  endowed with the Riemannian metric  $g_x^\kappa = (\lambda_x^\kappa)^2 \mathbf{I}$ , and  $\lambda_x^\kappa = 2(1 + \kappa\|x\|^2)^{-1}$  is the conformal factor. The distance function is thus given as

$$d_\kappa(x, y) = 2 \tan_\kappa^{-1}(\|(-x) \oplus_\kappa y\|). \quad (3)$$

The logarithmic map takes the form of

$$\text{Log}_x^\kappa(y) = \frac{2}{\lambda_x^\kappa} \tan_\kappa^{-1}(\|(-x) \oplus_\kappa y\|) \frac{(-x) \oplus_\kappa y}{\|(-x) \oplus_\kappa y\|}, \quad (4)$$

where  $\oplus_\kappa$  denotes Möbius addition of gyrovectors, and we have  $\tan_\kappa(x)$  defined as  $\tan(x)$  for  $\kappa > 0$ ,  $\tanh(x)$  for  $\kappa < 0$ , and  $\tan_\kappa(x) = x$  for  $\kappa = 0$ .

## Physics-informed Differential System

In the heart of Pioneer, we define a novel differential system on the  $\kappa$ -sterographical model  $\mathfrak{S}_\kappa^d$  to model the underlying dynamics. The system presents as a couple of ODEs in which a *Manifold ODE* is governed by the proposed ODE of *Constrained Ricci Flow*, as shown in Fig. 2. Given the initial state  $Z$  on the manifold, the system takes the form of

$$\frac{dZ^t}{dt} = u(Z^t; \Theta_1) \in \mathcal{T}\mathfrak{S}_\kappa^d, \quad Z^0 = Z \quad (5)$$

$$\frac{dw_{ij}^t}{dt} = \left( R_{ij}^F - e^{f(z_i^t, z_j^t; \Theta_2)} \right) w_{ij}^t, \quad z_i^t, z_j^t \in \mathfrak{S}_\kappa^d. \quad (6)$$

To be specific, equation 5 is the *Manifold ODE* on  $\mathfrak{S}_\kappa^d$  describing the system co-evolution over time, where  $Z^t$  collects the objects' latent representations at timestamp  $t$ . In the manifold ODE, *vector field*  $u$  is defined on a section of tangent bundle  $\mathcal{T}\mathfrak{S}_\kappa^d$ , and assigns latent representation  $z_i^t \in \mathfrak{S}_\kappa^d$  on the manifold to a tangent vector of the bundle. That is,  $u$  is a function that receives manifold-value vector and outputs tangent vector. In light of the co-evolution among the objects, we define the vector field with a graph neural network,

$$u(Z^t; \Theta) = \text{GAT}(\mathbf{A}^t, \text{Log}_\kappa^\kappa(Z^t); \Theta). \quad (7)$$

Concretely, we conduct the logarithmic map  $\text{Log}_o^\kappa$  to transform manifold-value vector to tangent vector, and model the dependent objects in the tangent space with graph attention network (GAT) (Velickovic et al. 2018), where the attentional weight is defined by Eq. 6. Note that, we cannot directly apply an existing GNN owing to the input/output requirement of the vector field.

Equation 6 is the proposed **Constrained Ricci Flow**, an ODE over the manifold-valued  $\mathbf{z}^t \in \mathbb{S}_\kappa^d$ . It injects the physics law into the neural system, while describing the continuous changes of intercorrelation among objects (weighted graph structure). In particular, we propose to constrain the canonical Ricci flow in Eq. 1 by a parameterized model,

$$f(\mathbf{z}_i^t, \mathbf{z}_j^t; \Theta_2) = \sigma(\text{MLP}_{\Theta_2}(\text{Log}_o^\kappa(\mathbf{z}_i^t) \parallel \text{Log}_o^\kappa(\mathbf{z}_j^t))), \quad (8)$$

where  $\sigma$  and  $\parallel$  are sigmoid and concatenation, respectively. MLP is a multilayer perception with parameter  $\Theta_2$ . The intuition is that object correlation is expressed by their latent representations, and the advantage lies in the theory below.

**Theory** We elaborate on an interesting theoretical result of the Constrained Ricci Flow defined in Eq. 6, and we begin with reviewing the definition of von Neumann entropy.

**von Neumann entropy (2019).** *Given a dynamic graph  $G$  with evolving weighed adjacency matrix  $\mathbf{A}^t$ , the von Neumann entropy at time  $t$  is defined as  $H^t = -\sum_{i=1}^n \lambda_i^t \ln \lambda_i^t$ , where  $\lambda_i^t$  is the  $i$ th eigenvalue of  $\frac{1}{N} \mathbf{L}^t$ , where  $\mathbf{L}^t = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A}^t \mathbf{D}^{-\frac{1}{2}}$ , and  $\mathbf{D}$  is the diagonal degree matrix.*

Next, we study the derivative of  $\frac{dH}{dt}$  to characterize the continuous change of entropy over time.

**Theorem 3.1 (Entropy Non-decreasing of Constrained Ricci Flow).** *If the interacting system is governed by the constrained Ricci flow defined in Eq. 6, we have the derivative of von Neumann entropy over time  $\frac{dH}{dt} \geq 0$  hold under a slight condition that  $m \geq n$ , where  $n$  and  $m$  denote the number of nodes and edges, respectively.*

*Proof.* We present the key ideas and equations here. Let  $Ric$  to be the summation of Ricci curvature over graph. First, we begin with analyzing  $dRic$  over time under the proposed flow in Eq. 6, and give the lower bound as follows.

$$\frac{dRic}{dt} = \frac{1}{2} \sum_{e_{ij}} g(e_{ij}) \left( g(e_{ij}) + e^{f(\mathbf{z}_i^t, \mathbf{z}_j^t; \Theta_2)} - 2 \right) \quad (9)$$

$$\geq \frac{1}{2} [\sum_{e_{ij}} g(e_{ij}) - m] \quad (10)$$

where  $g(e_{ij}) = \sum_{u \sim i} \sqrt{\frac{w_{ij}}{w_{iu}}} + \sum_{v \sim j} \sqrt{\frac{w_{ij}}{w_{vj}}}$ . Second, we scale  $\sum_{e_{ij}} g(e_{ij})$  so as to establish the connection between the inequality above and the invariant values of the graph.

$$\sum_{e_{ij}} g(e_{ij}) = \frac{1}{2} \sum_{e_{ij}} \sum_{e_{uv} \sim e_{ij}} \left( \sqrt{\frac{w_{ij}}{w_{uv}}} + \sqrt{\frac{w_{uv}}{w_{ij}}} \right) \quad (11)$$

$$\sum_{e_{ij}} g(e_{ij}) \geq \sum_v d^2(v) - 2m \quad (12)$$

where  $e_{uv}$  enumerate the edges that intersect with edge  $e_{ij}$ , and  $d(v)$  denotes the degree of  $v$ . With  $\sum_v d^2(v) \geq 4m - n$  (detailed in Appendix A.1), we have  $\frac{dRic}{dt} \geq \frac{m-n}{2}$ . Third, we consider the measure space over the Riemannian structure. Under the second-order derivative w.r.t. time, the Hessian regarding the entropy is shown to be  $k$ -concavity, yielding the result that changes in entropy and curvature are positively correlated (Lott and Villani 2009),  $dH \cdot dRic \geq 0$ . Thus,  $\frac{dH}{dt} \geq 0$  holds on the condition of  $m \geq n$ , completing the proof. (Note that, the condition is satisfied on most of the real world graphs.)  $\square$

In other words, dynamic interacting systems endowed with the proposed flow *are theoretically in line with the generic physics principle of entropy increasing* (more strictly, non-decreasing). Additionally, it is noteworthy to mention that the canonical Ricci flow do not possess monotonicity over entropy (Baptista, MacArthur, and Banerji 2024).

## Manifold-Preserving Transformation

*Aware of the geometry, we consider the manifold-preserving transformation to generate trajectory on the manifold as well as process system observations. In previous works, feature transformation typically relies on the couple of exponential/logarithmic maps (Xiong et al. 2022), introducing a tangent space as the intermediary. Note that, Euclidean or tangent space is not isometric to the manifold (Petersen 2016; Yu and Sa 2023). It leads to inevitable mapping error, which is also evidenced in Experiment, and is a key reason of the fully Riemannian design of Pioneer.*

To bridge this gap, we formulate a novel **Gyro-transform** in any constant curvature. The advantage lies in that Gyro-transform only requires simple matrix-vector multiplication and does not involves the tangent space (manifold-preserving). In particular, the gyro-transform from  $\mathbb{S}_\kappa^{d_1}$  to  $\mathbb{S}_\kappa^{d_2}$  is performed by left multiplying  $\mathbf{z} \in \mathbb{S}_\kappa^{d_1}$  with the matrix  $Gyr_{\mathbf{z}}(\mathbf{W}) \in \mathbb{R}^{d_1 \times d_2}$ , which is defined by rescaling a weight matrix  $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$  as follows,

$$Gyr_{\mathbf{z}}(\mathbf{W}) = f_{scal}^\kappa(\mathbf{W}, \mathbf{z}) \mathbf{W} \quad (13)$$

where the scaling function  $f_{scal}^\kappa$  is defined as

$$f_{scal}^\kappa(\mathbf{W}, \mathbf{z}) = \frac{\sqrt{\kappa^{-1}[1 - (\lambda_z^\kappa - 1)^2]}}{\lambda_z^\kappa \|\mathbf{W}\mathbf{z}\|}, \quad (14)$$

$\lambda_z^\kappa$  is the conformal factor, and  $sgn$  is the sign function. The derivation of Gyro-transform is provided in Appendix A.2.

**Theorem 3.2 (Manifold-preserving of Gyro-transform).** *Given  $\mathbf{z} \in \mathbb{S}_\kappa^{d_1}$  and  $\kappa$  of either positive or negative value,  $Gyr_{\mathbf{z}}(\mathbf{W})\mathbf{z} \in \mathbb{S}_\kappa^{d_2}$  holds for any  $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$ .*

*Proof.* We verify the correctness of the claim above, and further details are given in Appendix A.2. That is, we are to prove  $-\kappa \|Gyr_{\mathbf{z}}(\mathbf{W})\mathbf{z}\| < 1$  given the domain of manifold  $\mathbb{S}_\kappa^d$ . Instead of the inequality of  $\mathbb{S}_\kappa^d$ , we examine the equality in Lorentz/spherical model,  $\mathfrak{L}_\kappa^d = \left\{ \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_s \end{bmatrix} \in \mathbb{R}^d \mid sgn(\kappa)x_t^2 + \mathbf{x}_s^\top \mathbf{x}_s = \frac{1}{\kappa} \right\}$ , where  $sgn$  is the sign function. (Note that,  $\mathfrak{L}_\kappa^d$  model is equivalent to  $\mathbb{S}_\kappa^d$  model, and the connection is established by  $\kappa$ -stereographical projec-

tion  $\Phi : \mathfrak{L}_\kappa^d \rightarrow \mathfrak{st}_\kappa^d$ .) First, we conduct  $\Phi^{-1}$  on  $Gyr_z(\mathbf{W})z$ ,

$$\Phi^{-1}(Gyr_z(\mathbf{W})z) = \begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{|\kappa|}}(\lambda_z^\kappa - 1) \\ \lambda_z^\kappa f_{scal}^\kappa(\lambda_z^\kappa) \mathbf{W}z \end{bmatrix}.$$

Second, we have the manifold equation of  $\mathfrak{L}_\kappa^{d_2}$  model hold for  $\Phi^{-1}(Gyr_z(\mathbf{W})z)$ , completing the proof.  $\square$

In addition, the proposed Gyro-transform is more computationally efficient than the couple of exponential/logarithmic maps, since  $Gyr_z(\mathbf{W})z$  has fewer norms and no tanh's.

**Remark** Fully manifold operators of Chen et al. (2022); Bdeir, Schwethelm, and Landwehr (2024) are designed for hyperbolic space only, while Lorentz projector (Sun et al. 2023c) is for manifold shift (i.e., from hyperbolic to hyperspherical space, or vice versa). None of them meets the need of feature transform in any constant curvature.

### Riemannian Encoder & Decoder

In **Pioneer**, to generate the initial states of ODEs, the Riemannian encoder is constructed with Gyro-transform and the weighted gyro-midpoint aggregator (Petersen 2016),

$$\text{mid}_\kappa(\mathbf{h}, \alpha) = \frac{1}{2} \otimes_\kappa \left( \sum_{i=1}^{T_{Obs}} \frac{\alpha_i \lambda_{\mathbf{h}_i}^\kappa}{\sum_{j=1}^{T_{Obs}} \alpha_j (\lambda_{\mathbf{h}_j}^\kappa - 1)} \mathbf{h}_i^i \right), \quad (15)$$

where  $\alpha_i$  denotes the weight,  $\mathbf{h}_i^i \in \mathfrak{S}_\kappa^d$ ,  $\text{mid}_\kappa(\mathbf{h}, \alpha) \in \mathfrak{S}_\kappa^d$ . First, for each observed graph  $\mathbf{G}^1, \dots, \mathbf{G}^{T_{Obs}}, \mathbf{G}^t(\mathbf{A}, \mathbf{X})$ , we obtain the object representations by the manifold GCN, where the feature transformation of Gyro-transform  $\mathbf{h} = Gyr_x(\mathbf{W})x$  is followed up with the aggregation of  $\mathbf{h} = \text{mid}_\kappa(\mathbf{h}; \alpha)$ . Second, we conduct the temporal self-attention on the manifold to derive the initial state of manifold ODE in Eq. 5. Concretely, the contribution of each timestamp is modeled with temporal self-attention  $\alpha$  parameterized by  $\mathbf{w}$ ,

$$\alpha_i = \frac{\exp(\mathbf{w}^\top [\mathbf{t}_i || \mathbf{t}_{initial}])}{\sum_j^{T_{Obs}} \exp(\mathbf{w}^\top [\mathbf{t}_i || \mathbf{t}_{initial}])}, \quad (16)$$

$$[\mathbf{t}]_{2i} = \sin\left(\frac{t}{10000^{2i/d}}\right), \quad [\mathbf{t}]_{2i+1} = \cos\left(\frac{t}{10000^{2i/d}}\right). \quad (17)$$

For timestamp  $t$ , we leverage the time encoding  $\mathbf{t}$ , and elements of even and odd index are given in Eq. 17, and  $t_{initial}$  is the time of initial state. Accordingly, the initial state of constrained Ricci flow in Eq. 6 is given by the distance metric on the manifold,  $w_{ij} = \frac{\exp(-d_\kappa(z_i, z_j))}{\sum_{k=1}^n \exp(-d_\kappa(z_i, z_k))}$ .

The decoder of **Pioneer** is given as a Gyro-transform to generate prediction on the manifold.

**Objective Function & Complexity Analysis.** Finally, we formulate the reconstruction objective as follows

$$\mathcal{J} = \sum_{t=1}^T \left( \sum_{i=1}^N d_\kappa^2(\mathbf{y}_i(t), \hat{\mathbf{y}}_i(t)) + \sum_{(i,j) \in \mathcal{E}} |w_{ij}(t) - \hat{w}_{ij}(t)|^2 \right), \quad (18)$$

where  $\hat{\mathbf{y}}_i(t)$  and  $\hat{w}_{ij}(t)$  are given by the observations, and  $T$  is the prediction length.  $\mathbf{y}_i(t)$  is the output of the decoder, and  $w_{ij}(t)$  is given by the constrained Ricci flow. We summarize the overall learning procedure in Algorithm 1. Accordingly, the computational complexity of the encoder is

---

### Algorithm 1: Learning Algorithm of **Pioneer**

---

```

Input: The system observations  $(\mathcal{G}, \mathcal{X})$ 
Output: The parameters of Pioneer
1 Initialize modal parameters;
2 while model not converged do
3   for each training sequence do
4     Separate the sequence into observed half
5        $[T_0, T_1]$  and predicted half  $[T_1, T_2]$ ;
6        $\triangleright$  Initial State Encoder
7       Obtain object representations by manifold
8       GCN;
9       Derive initial states with the temporal
10      self-attention in Eqs. 15-16;
11       $\triangleright$  Riemannian Differential System
12      Solve the ODEs in Eqs. 5-6 over  $[T_1, T_2]$ ;
13       $\triangleright$  Decoder
14      Generate prediction via the Gyro-transform
15      in Eq. 13;
16      Compute the objective function in Eq. 18;
17      Update the parameters by gradient descent.

```

---

$O(TN^2)$ , and  $O(TN^2)$  for the proposed ODEs, and  $O(N)$  for the decoder, where  $T$  is the number of objects. It yields the *Square complexity* as a whole. Though the complexity is the same as existing graph ODEs (Huang, Sun, and Wang 2023; Chen et al. 2024), **Pioneer** is aware of the geometry, and the generated system state adhere to the physics law of the ubiquitous entropy-increasing real systems.

## Experiment

**Datasets & Baselines** The experiments are conducted on three real-world datasets of dynamic systems, including Weather (Nikitin et al. 2022), CReSIS (Gogineni et al. 2013) and Social (Gu, Sun, and Gao 2017). We follow Huang, Sun, and Wang (2021) to preprocess the datasets. To evaluate **Pioneer**, we compare with nine strong baselines of two categories: (1) RNN-based methods: LSTM, NRI (Kipf et al. 2018), VGRNN (Hajiramezanali et al. 2019) and DGCNN (Li et al. 2023). (2) ODE-based methods: NODE (Chen et al. 2018), LG-ODE (Huang, Sun, and Wang 2020), CG-ODE (Huang, Sun, and Wang 2021), HBNODE (Xia et al. 2021) and HOPE (Luo et al. 2023). So far, existing methods neither consider Riemannian geometry nor system entropy, and we are dedicated to bridging this gap.

**Evaluation Protocol** We focus on predicting the future trajectories, and evaluate all methods by the metrics of Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) following (Huang, Sun, and Wang 2021; Luo et al. 2023). We perform 10 independent runs for each case, and report the mean with standard derivations.

**Configuration & Reproducibility** In **Pioneer**, graph attention layer in manifold ODE is stacked twice, and MLP has 1 hidden layers. The dimension of latent representation on manifold  $\mathfrak{S}$  is set as 16. For Euclidean features, we apply

Methods	Social Dataset (Different Prediction Length)								Weather Dataset (Different Prediction Length)							
	10		20		40		5		10		15					
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
LSTM	12.18 $\pm$ 0.21	0.14 $\pm$ 0.04	26.16 $\pm$ 0.25	0.26 $\pm$ 0.05	61.74 $\pm$ 0.35	0.51 $\pm$ 0.02	18.99 $\pm$ 0.45	5.90 $\pm$ 0.03	23.42 $\pm$ 0.18	8.31 $\pm$ 0.04	75.22 $\pm$ 0.31	9.58 $\pm$ 0.03				
NRI	29.35 $\pm$ 0.42	0.56 $\pm$ 0.09	43.24 $\pm$ 1.01	0.69 $\pm$ 0.12	68.23 $\pm$ 1.34	0.70 $\pm$ 0.11	22.75 $\pm$ 4.94	7.13 $\pm$ 1.36	22.38 $\pm$ 1.66	8.06 $\pm$ 0.85	68.16 $\pm$ 4.00	9.82 $\pm$ 0.19				
VGRNN	11.78 $\pm$ 0.29	0.14 $\pm$ 0.02	26.87 $\pm$ 0.20	0.28 $\pm$ 0.07	54.90 $\pm$ 0.42	0.51 $\pm$ 0.06	14.59 $\pm$ 0.23	5.79 $\pm$ 0.13	19.56 $\pm$ 1.00	7.96 $\pm$ 0.22	74.37 $\pm$ 3.61	9.61 $\pm$ 0.45				
DGCRN	11.87 $\pm$ 0.20	0.14 $\pm$ 0.02	25.84 $\pm$ 0.23	0.31 $\pm$ 0.06	57.03 $\pm$ 0.35	0.58 $\pm$ 0.09	15.39 $\pm$ 1.36	5.75 $\pm$ 0.36	21.57 $\pm$ 2.82	7.84 $\pm$ 0.53	45.66 $\pm$ 0.93	9.20 $\pm$ 0.20				
NODE	11.64 $\pm$ 0.17	0.16 $\pm$ 0.02	24.92 $\pm$ 0.35	0.30 $\pm$ 0.07	57.50 $\pm$ 0.33	0.64 $\pm$ 0.03	15.96 $\pm$ 0.16	5.99 $\pm$ 0.15	24.37 $\pm$ 1.80	8.22 $\pm$ 0.76	60.92 $\pm$ 1.35	8.83 $\pm$ 0.13				
HBNODE	10.78 $\pm$ 0.25	0.17 $\pm$ 0.02	26.37 $\pm$ 0.48	0.32 $\pm$ 0.08	61.51 $\pm$ 0.65	0.62 $\pm$ 0.07	18.56 $\pm$ 0.45	6.06 $\pm$ 0.25	24.43 $\pm$ 0.87	8.12 $\pm$ 0.09	53.31 $\pm$ 0.26	9.09 $\pm$ 0.63				
LG-ODE	12.80 $\pm$ 0.16	0.13 $\pm$ 0.01	25.46 $\pm$ 0.25	0.24 $\pm$ 0.04	55.59 $\pm$ 0.43	0.51 $\pm$ 0.03	14.53 $\pm$ 0.26	5.78 $\pm$ 0.11	21.99 $\pm$ 0.66	7.86 $\pm$ 0.42	45.41 $\pm$ 1.35	8.68 $\pm$ 0.42				
CG-ODE	12.63 $\pm$ 0.20	0.13 $\pm$ 0.02	24.47 $\pm$ 0.38	0.23 $\pm$ 0.04	49.09 $\pm$ 0.35	0.49 $\pm$ 0.06	14.01 $\pm$ 0.32	5.92 $\pm$ 0.07	20.87 $\pm$ 0.51	7.99 $\pm$ 0.31	28.17 $\pm$ 1.25	8.85 $\pm$ 0.31				
HOPE	10.71 $\pm$ 0.12	0.13 $\pm$ 0.01	22.13 $\pm$ 0.19	0.23 $\pm$ 0.03	47.82 $\pm$ 0.22	0.49 $\pm$ 0.03	13.76 $\pm$ 0.16	5.83 $\pm$ 0.15	19.54 $\pm$ 0.35	7.75 $\pm$ 0.22	24.16 $\pm$ 0.55	8.80 $\pm$ 0.08				
<b>Pioneer</b>	<b>9.15<math>\pm</math>0.12</b>	<b>0.11<math>\pm</math>0.01</b>	<b>18.67<math>\pm</math>0.12</b>	<b>0.21<math>\pm</math>0.01</b>	<b>44.07<math>\pm</math>0.18</b>	<b>0.45<math>\pm</math>0.02</b>	<b>12.50<math>\pm</math>0.48</b>	<b>5.74<math>\pm</math>0.09</b>	<b>17.27<math>\pm</math>0.23</b>	<b>7.76<math>\pm</math>0.02</b>	<b>19.39<math>\pm</math>0.01</b>	<b>8.64<math>\pm</math>0.03</b>				

Table 1: Prediction results with varying prediction lengths on Social and Weather datasets in terms of MAPE(%) and RMSE. Standard derivations are given in the subscript. The best results are **bolded** and the runners-up are underlined.

Methods	CReSIS Dataset (Different Prediction Length)							
	1		2		3			
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
LSTM	3.70	52.49	4.65	66.02	4.64	68.98		
NRI	3.82	51.89	4.58	68.78	4.83	62.79		
VGRNN	<u>3.62</u>	50.41	4.42	65.22	4.42	62.19		
DGCRN	3.71	48.52	4.41	65.02	4.38	58.18		
NODE	3.89	52.90	4.46	65.51	4.70	63.74		
HBNODE	3.76	51.78	4.41	66.24	4.73	62.75		
LG-ODE	3.96	52.51	4.77	63.26	4.64	61.66		
CG-ODE	3.86	49.71	<u>4.40</u>	62.52	4.32	58.07		
HOPE	3.72	47.95	4.50	<u>62.11</u>	4.23	57.29		
<b>Pioneer</b>	<b>3.61</b>	<b>47.78</b>	<b>4.37</b>	<b>60.86</b>	<b>4.22</b>	<b>57.24</b>		

Table 2: Prediction results with varying prediction lengths on CReSIS datasets in terms of MAPE(%) and RMSE. The best results are **bolded** and the runners-up are underlined.

the exponential map to obtain corresponding manifold features in the initialization. The differential system is solved by Euler method on the manifold (Bielecki 2002), and parameters are optimized by Adam with learning rate of  $5e-4$ .

## Results and Discussion

**Main Results** Table 1 and Table 2 collect the prediction results on Social, Weather and CReSIS datasets with varying prediction length. Take CReSIS for instance, the prediction length of 3 means we examine the prediction error over the future 3 years. As shown in the tables, ODEs tend to obtain better performance especially for length term prediction. Graph ODEs (i.e., HOPE, CG-ODE and LG-ODE) outperform NODE and HBNODE for the most cases, suggesting the importance of correlation modeling. Note that, the proposed **Pioneer** consistently achieves the best results on all the real systems except only one case in Weather dataset. We argue that our success lies in its geometry and the adherence to physics law, and we further investigate proposed components in **Pioneer**.

**Ablation Study** We evaluate the effectiveness of the proposed 1) constrained Ricci flow, and 2) manifold-preserving Gyro-transform. For the former objective, we design three manifold-valued variants as follows: *w/oEvo* variant considers the fixed graph structure and disables Eq. 6, verifying the significance of modeling structural evolvement. *w/oRic* variant utilizes the parameterized  $f$  over node features to

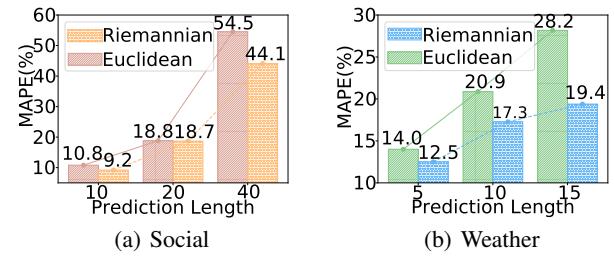


Figure 3: Results of geometric ablation.

describe the graph structure over time  $\frac{dw}{dt} = f$ , and does not consider the principle of Ricci flow. In *w/oCon* variant, Eq. 6 is replaced by the canonical Ricci flow in Eq. 1 with neither entropy constraint nor node features. To evaluate Gyro-transform, we design the *w/oGyr* variant which replaces Gyro-transform by the couple of exponential and logarithmic maps, involving the tangent space. The prediction results of the variants are collected in Table 3, and we find that: 1) **Pioneer** receives performance gain compared to *w/oGyr* variant, showing the superiority of manifold-preserving operations. 2) It suggests the importance of considering structural evolvement in dynamic system, given that *w/oEvo* variant has the worst results. 3) We achieve better results than *w/oRic* of direct parameterized model and *w/oCon* of the canonical flow, while the next part elaborates on another strength of **Pioneer**.

**Discussion on Entropy** We discuss the **generated system states** regarding the physics law in dynamic systems. To this end, we examine the continuous entropy changes of the real system and its simulators over time. The simulators include **Pioneer**, canonical Ricci flow, CG-ODE (Huang, Sun, and Wang 2021) as well as *w/oRic* variant. Concretely, we calculate the von Neumann entropy at different system snapshots, and report the results on Social dataset in Fig. 4. The real system presents the trend of entropy increasing in general, while the previous CG-ODE violates the fact in physics. Note that, Ricci flow itself cannot ensure entropy increasing, and *only the physics-informed Pioneer is in line with the physics fact of entropy increasing principle*.

**Discussion on Geometry & Case Study** First, we analyze the impact of different geometries with the mean error of

Variants	Social						Weather					
	10		20		40		5		10		15	
	MAPE	RMSE										
w/oEvo	11.61	0.13	24.37	0.25	59.23	0.51	15.94	6.13	18.51	7.98	20.70	8.87
w/oRic	<u>10.49</u>	0.13	22.17	0.23	<u>53.65</u>	<u>0.46</u>	<u>12.59</u>	<u>5.80</u>	17.74	7.78	19.71	8.77
w/oCon	10.76	<u>0.12</u>	<u>18.82</u>	<u>0.22</u>	54.47	0.49	12.62	5.82	<u>17.72</u>	7.78	<u>19.53</u>	<u>8.73</u>
w/oGyr	10.63	<u>0.12</u>	21.58	0.23	<u>45.09</u>	<u>0.46</u>	13.63	5.93	17.74	<u>7.77</u>	19.75	8.84
<b>Pioneer</b>	<b>9.15</b>	<b>0.11</b>	<b>18.67</b>	<b>0.21</b>	<b>44.07</b>	<b>0.45</b>	<b>12.50</b>	<b>5.74</b>	<b>17.27</b>	<b>7.76</b>	<b>19.39</b>	<b>8.64</b>

Table 3: Ablation study results on Social and Weather datasets with varying prediction lengths in terms of MAPE(%) and RMSE. The best results are **bolded** and the runners-up are underlined.

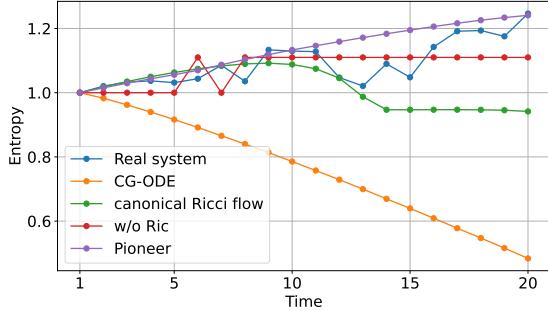


Figure 4: The changes of entropy over time.

prediction results, MAPE metric for instance. Fig. 3 shows the comparison between **Pioneer** and its corresponding Euclidean version. We find that the proposed model is superior to the Euclidean one. Second, we take a closer look by studying the spatial error distribution over the real system. In particular, we conduct a case study on CReSIS (Gogineni et al. 2013), a collection of glacier thickness in Greenland. We plot the error distribution of the Riemannian **Pioneer** on the earth’s surface around North Pole in Fig. 5(a), and that of Euclidean counterpart in Fig. 5(b), where darker color denotes larger error. Note that, *Euclidean model presents significant error near the North Pole, and the error distribution is generally related to geometric locations*, while **Pioneer** is more reliable throughout the earth’s system, verifying the importance of geometry and the motivation of our study.

## Related Work

**Dynamic Interacting System & Graph ODEs** Early practices leverage the recurrent models to study the sequential pattern of the objects. Given the correlation among objects, GNNs (Yang et al. 2023, 2024) are then incorporated with the recurrent models (Nicolicioiu, Duta, and Leordeanu 2019; Hajiramezanali et al. 2019; Fan et al. 2022). In recent years, graph ODEs show encouraging results for modeling the correlation and evolvement collaboratively (Huang, Sun, and Wang 2021; Luo, Haffari, and Pan 2023). Luo et al. (2023); Zhang et al. (2022); Xhonneux, Qu, and Tang (2020) introduce second-order ODE on graphs, while Huang, Sun, and Wang (2021); Luo, Haffari, and Pan (2023) consider the structural evolvement to model the dynamic correlation, and Chen et al. (2024) reconsider the graph structure of dynamic systems. Recently, Huang et al. (2023) study time-reversal symmetric systems. Lee, Trask, and Stinis (2021) present

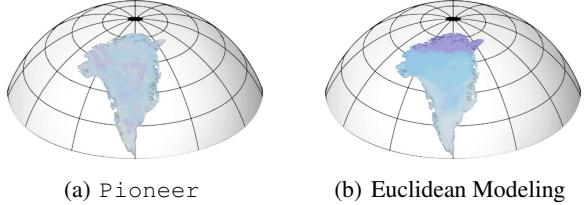


Figure 5: Prediction results of Glacier thickness.

a new parameterization of brackets for learning irreversible dynamics. Gruber, Lee, and Trask (2023) further develop a graph attention mechanism to enable the irreversibility in deep GNNs. Different from previous works, we build the differential system upon Riemannian manifold, considering the entropy increasing of dynamic systems.

**Riemannian Deep Learning on Graphs** Categorized by the geometry, hyperbolic space is suitable to hierarchical structures and show the superiority to the Euclidean counterpart (Sun et al. 2024c; Fu et al. 2024; Wei et al. 2024). Hyperspherical space achieves remarkable success to embed cyclical structures (Bonev et al. 2023). Recent studies further investigate the constant curvature spaces (Bachmann, Bécigneul, and Ganea 2020; Sun et al. 2024d), product spaces (Gu et al. 2019; Sun et al. 2022b, 2024e), quotient spaces (Xiong et al. 2022), SPD manifolds (Nguyen and Yang 2023), etc. For dynamic graphs, there exist Riemannian models focusing on link prediction and/or node classification (Yang et al. 2021; Sun et al. 2021, 2022a, 2023c). Recently, Ricci curvature is introduced to structure learning (Nguyen et al. 2023; Sun et al. 2023a) or clustering (Sun et al. 2023b), while its physics aspects are not explored. Also, we notice that Lou et al. (2020); Sun et al. (2024b,a); Wang et al. (2024); Chen and Lipman (2024) studies Riemannian ODEs or SDEs as generative models.

## Conclusion

We present the first graph differential system on Riemannian manifold (**Pioneer**) to study the dynamics of real-world systems. In accordance to the physics law, **Pioneer** is endowed with the constrained Ricci flow, which is theoretically proved to obey entropy non-decreasing with slight constraints. Meanwhile, we formulate the Gyro-transform to guarantee the manifold-preserving in both encoding and decoding processes. Finally, we empirically discuss the geometry and entropy changes in real system.

## Acknowledgements

This work is supported in part by NSFC under grants 62202164 and 62322202. Prof. Philip S. Yu is supported in part by NSF under grants III-2106758, and POSE-2346158.

## References

Bachmann, G.; Béćigneul, G.; and Ganea, O. 2020. Constant Curvature Graph Convolutional Networks. In *Proceedings of the 37th ICML*, 486–496. PMLR.

Bai, S.; Lin, Y.; Lu, L.; Wang, Z.; and Yau, S.-T. 2021. Ollivier Ricci-flow on weighted graphs. *American Journal of Mathematics*, To be determined: 1–20.

Baptista, A.; MacArthur, B. D.; and Banerji, C. R. 2024. Charting cellular differentiation trajectories with Ricci flow. *Nature Communications*, 15(1): 2258.

Bdeir, A.; Schwethelm, K.; and Landwehr, N. 2024. Fully Hyperbolic Convolutional Neural Networks for Computer Vision. In *Proceedings of the 12th ICLR*. OpenReview.net.

Bielecki, A. 2002. Estimation of the euler method error on a riemannian manifold. *Communications in Numerical Methods in Engineering*, 18(11): 757–763.

Bonev, B.; Kurth, T.; Hundt, C.; Pathak, J.; Baust, M.; Kashinath, K.; and Anandkumar, A. 2023. Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere. In *Proceedings of the 40th ICML*, volume 202, 2806–2823. PMLR.

Chen, L.; Wu, K.; Lou, J.; and Liu, J. 2024. Signed Graph Neural Ordinary Differential Equation for Modeling Continuous-Time Dynamics. In *Proceedings of the 38th AAAI*, 8292–8301. AAAI Press.

Chen, R. T. Q.; and Lipman, Y. 2024. Flow Matching on General Geometries. In *Proceedings of the 12th ICLR*. OpenReview.net.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. In *Advances in 31st NeurIPS*, 6572–6583.

Chen, W.; Han, X.; Lin, Y.; Zhao, H.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2022. Fully Hyperbolic Neural Networks. In *Proceedings of the 60th ACL*, 5672–5686. ACL.

Fan, J.; Bai, J.; Li, Z.; Ortiz-Bobea, A.; and Gomes, C. P. 2022. A GNN-RNN approach for harnessing geospatial and temporal information: application to crop yield prediction. In *Proceedings of the AAAI*, volume 36, 11873–11881.

Forman, R. 2003. Bochner’s Method for Cell Complexes and Combinatorial Ricci Curvature. *Discret. Comput. Geom.*, 29(3): 323–374.

Fu, X.; Gao, Y.; Wei, Y.; Sun, Q.; Peng, H.; Li, J.; and Li, X. 2024. Hyperbolic Geometric Latent Diffusion Model for Graph Generation. In *Proceedings of the 41st ICML*.

Gogineni, S.; Yan, J. B.; Gomez, D.; Rodriguez-Morales, F.; Paden, J.; and Leuschen, C. 2013. Ultra-wideband radars for remote sensing of snow and ice. In *IEEE MTT-S International Microwave and RF Conference*.

Gruber, A.; Lee, K.; and Trask, N. 2023. Reversible and irreversible bracket-based dynamics for deep graph neural networks. In *Advances in NeurIPS*.

Gu, A.; Sala, F.; Gunel, B.; and Ré, C. 2019. Learning mixed-curvature representations in products of model spaces. In *Proceedings of the ICLR*.

Gu, Y.; Sun, Y.; and Gao, J. 2017. The Co-Evolution Model for Social Network Evolving and Opinion Migration. In *Proceedings of the 23rd SIGKDD*, 175–184. ACM.

Hajiramezanali, E.; Hasanzadeh, A.; Narayanan, K. R.; Duffield, N.; Zhou, M.; and Qian, X. 2019. Variational Graph Recurrent Neural Networks. In *Advances in 32nd NeurIPS*, 10700–10710.

Horie, M.; and Mitsume, N. 2022. Physics-Embedded Neural Networks: Graph Neural PDE Solvers with Mixed Boundary Conditions. In *Advances in the 35th NeurIPS*.

Huang, Z.; Hwang, J.; Zhang, J.; Baik, J.; Zhang, W.; Wodarz, D.; Sun, Y.; Gu, Q.; and Wang, W. 2024. Causal Graph ODE: Continuous Treatment Effect Modeling in Multi-agent Dynamical Systems. In *Proceedings of the ACM The Web Conference*, 4607–4617. ACM.

Huang, Z.; Sun, Y.; and Wang, W. 2020. Learning Continuous System Dynamics from Irregularly-Sampled Partial Observations. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in the 33rd NeurIPS*.

Huang, Z.; Sun, Y.; and Wang, W. 2021. Coupled Graph ODE for Learning Interacting System Dynamics. In *Proceedings of the 27th SIGKDD*, 705–715. ACM.

Huang, Z.; Sun, Y.; and Wang, W. 2023. Generalizing Graph ODE for Learning Complex System Dynamics across Environments. In *Proceedings of the 29th SIGKDD*, 798–809. ACM.

Huang, Z.; Zhao, W.; Gao, J.; Hu, Z.; Luo, X.; Cao, Y.; Chen, Y.; Sun, Y.; and Wang, W. 2023. TANGO: Time-Reversal Latent GraphODE for Multi-Agent Dynamical Systems. In *NeurIPS 2023 Workshop DLDE*.

Khan, G. 2022. An Illustrated Introduction to the Ricci Flow. 1–42.

Kipf, T. N.; Fetaya, E.; Wang, K.; Welling, M.; and Zemel, R. S. 2018. Neural Relational Inference for Interacting Systems. In *Proceedings of the 35th ICML*, 2693–2702. PMLR.

Lee, K.; Trask, N.; and Stinis, P. 2021. Machine learning structure preserving brackets for forecasting irreversible processes. In *Advances in NeurIPS*, 5696–5707.

Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; and Li, Y. 2023. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution. *ACM Trans. Knowl. Discov. Data*, 9:1–9:21.

Liu, Z.; and Rahnemoonfar, M. 2024. Learning Spatio-Temporal Patterns of Polar Ice Layers With Physics-Informed Graph Neural Network. *CoRR*, abs/2406.15299.

Lott, J.; and Villani, C. 2009. Ricci Curvature for Metric-Measure Spaces via Optimal Transport. *Annals of Mathematics*, 169(3): 903–991.

Lou, A.; Lim, D.; Katsman, I.; Huang, L.; Jiang, Q.; Lim, S.; and Sa, C. D. 2020. Neural Manifold Ordinary Differential Equations. In *Advances in the 33rd NeurIPS*.

Luo, L.; Haffari, G.; and Pan, S. 2023. Graph Sequential Neural ODE Process for Link Prediction on Dynamic and Sparse Graphs. In *Proceedings of the 16th WSDM*, 778–786. ACM.

Luo, X.; Yuan, J.; Huang, Z.; Jiang, H.; Qin, Y.; Ju, W.; Zhang, M.; and Sun, Y. 2023. HOPE: High-order Graph ODE For Modeling Interacting Dynamics. In *Proceedings of the ICML*, 23124–23139. PMLR.

Mackey, M. C. 1989. The dynamic origin of increasing entropy. *Reviews of Modern Physics*, 61: 981.

Minello, G.; Rossi, L.; and Torsello, A. 2019. On the von Neumann entropy of graphs. *J. Complex Networks*, 7(4): 491–514.

Nguyen, K.; Hieu, N. M.; Nguyen, V. D.; Ho, N.; Osher, S. J.; and Nguyen, T. M. 2023. Revisiting Over-smoothing and Over-squashing Using Ollivier-Ricci Curvature. In *Proceedings of the ICML*, volume 202, 25956–25979. PMLR.

Nguyen, X. S.; and Yang, S. 2023. Building Neural Networks on Matrix Manifolds: A Gyrovector Space Approach. In *Proceedings of the 40th ICML*, volume 202, 26031–26062. PMLR.

Nicolicioiu, A. L.; Duta, I.; and Leordeanu, M. 2019. Recurrent Space-time Graph Neural Networks. In *Advances in 32nd NeurIPS*, 12818–12830.

Nikitin, A. V.; John, S. T.; Solin, A.; and Kaski, S. 2022. Non-separable Spatio-temporal Graph Kernels via SPDEs. In *Proceedings of the AISTATS*, 10640–10660. PMLR.

Ollivier, Y. 2007. Ricci curvature of Markov chains on metric spaces. *Journal of Functional Analysis*, 256: 810–864.

O’shea, D. 2009. *The Poincaré conjecture: In search of the shape of the universe*. Bloomsbury Publishing USA.

Petersen, P. 2016. *Riemannian Geometry*, 3rd edition. Springer-Verlag.

Sun, L.; Hu, J.; Li, M.; and Peng, H. 2024a. R-ODE: Ricci Curvature Tells When You Will be Informed. In *Proceedings of the ACM SIGIR*.

Sun, L.; Hu, J.; Zhou, S.; Huang, Z.; Ye, J.; Peng, H.; Yu, Z.; and Yu, P. S. 2024b. RicciNet: Deep Clustering via A Riemannian Generative Model. In *Proceedings of the ACM Web Conference (WWW)*, 4071–4082.

Sun, L.; Huang, Z.; Peng, H.; Wang, Y.; Liu, C.; and Yu, P. S. 2024c. LSEnet: Lorentz Structural Entropy Neural Network for Deep Graph Clustering. In *Proceedings of the 41st ICML*.

Sun, L.; Huang, Z.; Wan, Q.; Peng, H.; and Yu, P. S. 2024d. Spiking Graph Neural Network on Riemannian Manifolds. In *Advances in NeurIPS*.

Sun, L.; Huang, Z.; Wang, Z.; Peng, H.; and Yu, P. S. 2024e. Motif-aware Riemannian Graph Neural Network with Generative-Contrastive Learning. In *Proceedings of the 38th AAAI*, 9044–9052.

Sun, L.; Huang, Z.; Wu, H.; Ye, J.; Peng, H.; Yu, Z.; and Yu, P. S. 2023a. DeepRicci: Self-supervised Graph Structure-Feature Co-Refinement for Alleviating Over-squashing. In *Proceedings of the 23rd ICDM*, 558–567.

Sun, L.; Wang, F.; Ye, J.; Peng, H.; and Yu, P. S. 2023b. Con-gregate: Contrastive Graph Clustering in Curvature Spaces. In *Proceedings of the 32nd IJCAI*, 2296–2305.

Sun, L.; Ye, J.; Peng, H.; Wang, F.; and Yu, P. S. 2023c. Self-Supervised Continual Graph Learning in Adaptive Riemannian Spaces. In *Proceedings of the 37th AAAI*, 4633–4642.

Sun, L.; Ye, J.; Peng, H.; and Yu, P. S. 2022a. A Self-supervised Riemannian GNN with Time Varying Curvature for Temporal Graph Learning. In *Proceedings of the 31st CIKM*, 1827–1836.

Sun, L.; Zhang, Z.; Ye, J.; Peng, H.; Zhang, J.; Su, S.; and Yu, P. S. 2022b. A Self-Supervised Mixed-Curvature Graph Neural Network. In *Proceedings of AAAI’22*, 4146–4155.

Sun, L.; Zhang, Z.; Zhang, J.; Wang, F.; Peng, H.; Su, S.; and Yu, P. S. 2021. Hyperbolic Variational Graph Neural Network for Modeling Dynamic Graphs. In *Proceedings of the 35th AAAI*, 4375–4383.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the 6th ICLR*. OpenReview.net.

Wang, Y.; Zhang, S.; Ye, J.; Peng, H.; and Sun, L. 2024. A Mixed-Curvature Graph Diffusion Model. In *Proceedings of the 33rd ACM CIKM*, 2482–2492. ACM.

Wei, Y.; Yuan, H.; Fu, X.; Sun, Q.; Peng, H.; Li, X.; and Hu, C. 2024. Poincaré Differential Privacy for Hierarchy-Aware Graph Embedding. In *Proceedings of AAAI*, 9160–9168.

Xhonneux, L.-P.; Qu, M.; and Tang, J. 2020. Continuous graph neural networks. In *Proceedings of the 37th ICML*, 10432–10441. PMLR.

Xia, H.; Suliafu, V.; Ji, H.; Nguyen, T. M.; Bertozzi, A. L.; Osher, S. J.; and Wang, B. 2021. Heavy Ball Neural Ordinary Differential Equations. In *Advances in 34th NeurIPS*, 18646–18659.

Xiong, B.; Zhu, S.; Potyka, N.; Pan, S.; Zhou, C.; and Staab, S. 2022. Pseudo-Riemannian Graph Convolutional Networks. In *Advances in 35th NeurIPS*.

Yang, L.; Hu, W.; Xu, J.; Shi, R.; He, D.; Wang, C.; Cao, X.; Wang, Z.; Niu, B.; and Guo, Y. 2024. GAUSS: GrAph-customized Universal Self-Supervised Learning. In *Proceedings of WWW*, 582–593.

Yang, L.; Shi, R.; Zhang, Q.; Niu, B.; Wang, Z.; Cao, X.; and Wang, C. 2023. Self-supervised Graph Neural Networks via Low-Rank Decomposition. In *Advances in NeurIPS*.

Yang, M.; Zhou, M.; Kalander, M.; Huang, Z.; and King, I. 2021. Discrete-time Temporal Network Embedding via Implicit Hierarchical Learning in Hyperbolic Space. In *Proceedings of the 27th SIGKDD*, 1975–1985.

Yu, T.; and Sa, C. D. 2023. Random Laplacian Features for Learning with Hyperbolic Space. In *Proceedings of the 11th ICLR*. OpenReview.net.

Zhang, Y.; Gao, S.; Pei, J.; and Huang, H. 2022. Improving Social Network Embedding via New Second-Order Continuous Graph Neural Networks. In *Proceedings of the 28th SIGKDD*, 2515–2523. ACM.