NEURAL HAMILTONIAN FLOWS IN GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) suffer from oversmoothing and oversquashing problems when node features are updated over too many layers. Embedding spaces can also vary significantly for different data types, leading to the need for different GNN model types. In this paper, we model the embedding of a node feature as a Hamiltonian flow over time. As in physics where Hamiltonian flow conserves the energy over time, its induced GNNs enable a more stable feature updating mechanism. Moreover, since the Hamiltonian flows are defined on a general symplectic manifold, this approach allows us to learn the underlying manifold of the graph in training, in contrast to most of the existing literature that assumes a fixed graph embedding manifold. We test Hamiltonian flows of different forms and demonstrate empirically that our approach achieves better node classification accuracy than popular state-of-the-art GNNs.

1 INTRODUCTION

Graph neural networks (GNNs) (Yue et al., 2019; Ashoor et al., 2020; Kipf & Welling, 2017b; Zhang et al., 2022; Wu et al., 2021) have achieved great success in inferring the graph-structured data such as social media networks, the molecular graph in chemistry and so on. Most existing GNNs embed graph nodes to Euclidean spaces without further consideration of the dataset graph structures. However, for some graph structures like the tree-like graphs (Liu et al., 2019), the Euclidean space may not be a proper choice for the node embedding. Recently, hyperbolic GNNs (Chami et al., 2019; Liu et al., 2019) propose to embed nodes into the specific hyperbolic space, whereas conventional GNNs embed nodes in Euclidean space. It has been shown that tree-like graphs can be inferred more accurately by Hyperbolic GNNs. Furthermore, some works like Zhu et al. (2020) have tried to embed graph nodes in a mixed space of the Euclidean and hyperbolic spaces where the intrinsic graph local geometry is attained from the mixing weight.

Embedding nodes in hyperbolic space is achieved through the exponential map (Chami et al., 2019) which is essentially a geodesic curve on the hyperbolic manifold which is the projected integral curve of the cogeodesic flow on the cotangent bundle of the hyperbolic manifold (Lee, 2013; Klingenberg, 2011). In our work, we propose to embed the nodes, via more general Hamiltonian flows, into a general manifold, which generalizes the hyperbolic embedding space, i.e., a strongly constrained Riemannian manifold of constant sectional curvature equal to -1. Unlike the aforementioned works Zhu et al. (2020); Liu et al. (2019); Chami et al. (2019), which all consider the Riemannian manifold on which a metric tensor is assigned to determine the manifold structure, in our work, we do not assume the embedding manifold is a Riemannian manifold in general. Instead, we only assign a general symplectic structure to the cotangent bundle of the manifold and therefore generalize the Riemannian manifold graph node embedding literature.

From the physics perspective, the cotangent bundles are the natural phase spaces in classical mechanics (De León & Rodrigues, 2011) where the physical system evolves according to the basic laws of physics which are modeled as differential equations on the phase spaces. In this paper, we propose a new GNN paradigm based on Hamiltonian mechanics (Goldstein et al., 2001) for its energy-conservative nature. Our objective is to develop a new graph node manifold embedding strategy where the features evolve following the Hamiltonian flows on the manifold under the influence of neighbors. The learnable Hamiltonian function and the symplectic structure on the manifold guide that the node feature updating follows a learnable law analogous to the basic physical laws.

Main contributions. Our main contributions are summarized as follows:

1. We review the notion of symplectic manifolds and Hamiltonian vector fields on it from the differential geometry perspective.
2. We take the graph as a discretization of an underlying manifold. We learn a Hamiltonian flow associated to the Hamiltonian scalar function on its cotangent bundle, where different symplectic forms are investigated.
3. By the conservative nature of Hamilton equations, our model enables a stable training and inference process while updating the node features over time and layers.
4. We empirically demonstrate that our approach achieves better node classification accuracy than popular state-of-the-art GNNs. Furthermore, the oversmoothing of GNNs can be mitigated if the node features evolve through the Hamiltonian flows.

The rest of this paper is organized as follows. The related work is discussed first. We then introduce some basic concepts from differential geometry. Our model and framework are then presented. We present extensive experimental results, and we conclude our paper in the last section.

2 Related Work

While our paper is related to Hamiltonian neural networks in the literature, we are the first, to our best knowledge, to model graph-structured data with Hamiltonian equations. In what follows, we will briefly review Hamiltonian neural networks. Moreover, from the embedding manifold point of view, modeling graph-structured data with Hamiltonian equations actually embeds the graph nodes into a general flexibly learnable manifold via its symplectic cotangent bundle. Hence, we can also relate our work to Hyperbolic GNNs, which embed nodes into a hyperbolic Riemannian manifold.

Hamiltonian Neural Networks: Among these physically-inspired deep learning approaches, Hamiltonian equations have been applied to conserve an energy-like quantity when training neural networks. Greydanus et al. (2019); Zhong et al. (2020); Chen et al. (2021) train a neural network to infer the Hamiltonian dynamics of a physical system, where Hamiltonian equations are solved using neural ODE solvers. Haber & Ruthotto (2017) build Hamiltonian-inspired neural ODE to stabilize the gradients so as to avoid vanishing and exploding gradients. Huang et al. (2022) further studies the adversarial robustness of the Hamiltonian ODE. *In this paper, we mainly focus on applying Hamiltonian equations to graph neural networks.*

Riemannian Manifold GNNs: Most GNNs in the literature Yue et al. (2019); Ashoor et al. (2020); Kipf & Welling (2017b); Zhang et al. (2022); Wu et al. (2021) embed graph nodes to Euclidean spaces. In what follows, we simply call them (vanilla) GNNs. They perform well on some datasets like the Cora dataset McCallum et al. (2004) whose δ-hyperbolicity Chami et al. (2019) is high. When dealing with the datasets whose δ-hyperbolicity are low such as the Disease Chami et al. (2019) and Airport Chami et al. (2019) datasets, those GNNs suffer from improper node embedding because of their Euclidean nature. To better handle hierarchical graph data, (Liu et al., 2019; Chami et al., 2019; Zhang et al., 2021; Zhu et al., 2020) propose to embed nodes into a hyperbolic space, thus yielding hyperbolic GNNs. Moreover, Zhu et al. (2020) targets embedding graph nodes to a Riemannian manifold which is a mixed space of the Euclidean and hyperbolic spaces. This mixing operation has relaxed the strong space constraints if we only choose one of the two spaces for embedding. *In this paper, instead of embedding nodes in a Riemannian manifold, we embed nodes into a general, flexibly learnable manifold via the Hamiltonian flow on its symplectic cotangent bundle.*

Notations: We use the *Einstein summation convention* (Lee, 2013) for expressions with tensor indices.

3 Preliminaries

In this section, we briefly review the topic of symplectic manifolds and the Hamiltonian vector fields on it. We highly recommend the readers to Appendix A for a self-contained review of differential geometry that is related to our work. We will consider a graph as a discretization of a manifold later.

3.1 Manifold and Cotangent Bundles

A manifold is a topological space that *locally* looks like a Euclidean space. Strictly speaking, a topological space $(M, \mathcal{O})$, where $\mathcal{O}$ is the collection of open sets on space M, is called a d-

dimensional manifold if for every point $x \in M$, we can find an open neighborhood $U \in \mathcal{O}$ for x and a coordinate map

$$q : U \to q(U) \subseteq \mathbb{R}^d$$

that is a *homeomorphism*, where $\mathbb{R}^d$ is the d-dimensional Euclidean space with the standard topology. (U, q) is called a *chart* of the manifold, which gives us a numerical representation for a local area in M. In this work, we only consider smooth manifolds (Lee, 2013). We mainly consider the *symplectic form* (Lee, 2013) on the manifold with some other light shed on the metric tensor which is another type of $(0, 2)$ smooth tensor field.

Definition 1 (tangent vector and tangent space)**.** *Let $\gamma : I \to M$ be a smooth curve through $x \in M$ s.t. $\gamma(0) = x$ and $I \subseteq \mathbb{R}$ is an interval neighborhood of 0. The* tangent vector *is a directional derivative operator at x along γ that is the* linear map

$$v_{\gamma,x} : \mathcal{C}^\infty(M) \xrightarrow{\sim} \mathbb{R}, \quad f \mapsto (f \circ \gamma)'(0).$$

We also call the directional derivative operator as "velocity" of γ at 0 and denote it as $\dot{\gamma}(0)$. (More generally, the velocity of γ at t is denoted at $\dot{\gamma}(t)$ which is a tangent vector at point $\gamma(t) \in M$). Correspondingly, the tangent space *to M at x is the* vector space *over $\mathbb{R}$ with the underlying set*

$$T_x M \coloneqq \{v_{\gamma,x} \mid \gamma \text{ is a smooth curve and } \gamma(0) = x\}. \tag{1}$$

For a local chart (U, q) around x, we have a basis *of $T_x M$ as $\{(\frac{\partial}{\partial q^1})_x, \ldots, (\frac{\partial}{\partial q^d})_x\}$, where $(\frac{\partial}{\partial q^1})_x \coloneqq \partial_i (f \circ q^{-1})|_{q(x)}$. We call it the* chart induced basis.

Definition 2 (tangent bundle[1])**.** *The tangent bundle TM of M is $TM \coloneqq \coprod_{x \in M} T_x M$, equipped with the* projection *$\tilde{\pi} : TM \to M$ that maps a tangent vector to the point it is associated on M.*

Definition 3 (vector field)**.** *A* vector field *on M is a smooth section, i.e., a smooth map $\sigma : M \to TM$ s.t. $\tilde{\pi} \circ \sigma = id_M$, where id_M is the identity map on M. We denote the set of all vector fields on M as:*

$$\Gamma(TM) \coloneqq \{\sigma : M \to TM \mid \sigma \text{ is smooth and } \tilde{\pi} \circ \sigma = id_M\}. \tag{2}$$

Definition 4 (integral curve)**.** *Given a vector field X on M, an integral curve of X is a differentiable curve $\gamma : I \to M$ whose velocity at each point is equal to the value of X at that point:*

$$\dot{\gamma}(t) = X_{\gamma(t)} \quad \text{for all } t \in I. \tag{3}$$

Definition 5 (cotangent vector, cotangent bundle, dual basis, and covector field)**.** *For the vector space $T_x M$, a continuous linear functional from $T_x M$ to $\mathbb{R}$ is called a cotangent vector at x. The set of all such linear maps is denoted as $T_x^* M$ which is the dual vector space of $T_x M$. For $f \in \mathcal{C}^\infty(M)$, at each point x, we define the following linear operator in $T_x^* M$*

$$(\mathrm{d}f)_x : T_x M \to \mathbb{R}, \quad X_x \mapsto (\mathrm{d}f)_x(X_x) \coloneqq X_x(f).$$

Given a chart (U, q) with $x \in U$ and its chart induced basis*, the* dual basis *for the dual space $T_x^* M$ is the set $\left\{\left(\mathrm{d}q^1\right)_x, \ldots, \left(\mathrm{d}q^d\right)_x\right\}$, where we have $(\mathrm{d}q^a)_x \left(\left(\frac{\partial}{\partial q^b}\right)_x\right) = \left(\frac{\partial}{\partial q^b}\right)_x (q^a) = \delta_b^a$ with $\delta_b^a = 1$ iff $a = b$ and $\delta_b^a = 0$ otherwise. We call it the* chart induced dual basis *. Analogous to the above definition of the vector field, we can define the cotangent bundle of M as $T^* M \coloneqq \coprod_{x \in M} T_x^* M$ which is a $2d$-dimensional manifold equipped with the* canonical projection map *$\pi : T^* M \to M, \quad \omega \mapsto \pi(\omega)$, where $\pi(\omega)$ sends each vector in $T_x^* M$ to the point x at which it is cotangent. If $\omega \in \pi^{-1}(U) \subseteq T^* M$ with a local chart (U, q) s.t. $x \in U$, then $\omega \in T_{\pi(\omega)}^* M$ from the definition. Since $\pi(\omega) \in U$, ω can be written in terms of the* chart induced dual basis*:*

$$\omega = p_i(\omega)(\mathrm{d}q^i)_{\pi(\omega)}, \tag{4}$$

where $p_1, \ldots, p_d$ are smooth scalar functions. We can then define the following map as a local chart for manifold $T^* M$ induced from the chart (U, q) on M:

$$\xi : \pi^{-1}(U) \to q(U) \times \mathbb{R}^d \subseteq \mathbb{R}^{2d}, \quad \omega \mapsto (q^1(\pi(\omega)), \ldots, q^d(\pi(\omega)), p_1(\omega), \ldots, p_d(\omega)),$$

and the topological structure on $T^* M$ is derived from the initial topology to ensure continuity. The covector fields are smooth sections of $T^* M$. The set of all covector fields is denoted as $\Gamma(T^* M)$.

[1] In this paper, we may just use the word "bundle" to indicate the total space in the bundle.

Definition 6 (tensor field and 2-form)**.** *The differential k-form ω is the $(0, k)$ tensor field (using the smooth sections on tensor bundles, analogous to the above tangent or cotangent bundles) that admits alternating (Lee, 2013). Specifically, for the 2-form ω^2, at each point x, ω_x^2 is an* antisymmetric $(0, 2)$ *bilinear tensor as $\omega_x^2 : T_xM \times T_xM \to \mathbb{R}$ s. t. $\omega_x^2(X_1, X_2) = -\omega_x^2(X_2, X_1)\ \forall\, X_1, X_2 \in T_xM$, which in other words, ω satisfies*

$$\omega^2 : \Gamma(TM) \times \Gamma(TM) \to \mathcal{C}^\infty(M) \text{ s. t. } \omega^2(X, Y) = -\omega^2(Y, X)\ \forall\, X, Y \in \Gamma(TM) \tag{5}$$

In local chart representation, we have that every k-form ω^2 can be expressed locally *on U as*

$$\omega^2 = \omega^2_{a_1 \cdots a_k}\, \mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}, \tag{6}$$

where $\omega^2_{a_1 \cdots a_k}$ are smooth function, $1 \le a_1 < \cdots < a_k \le \dim M$ are increasing sequences and $\mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}$ is the wedge product (Lee, 2013).

Definition 7 (exterior derivative and closed form)**.** *The exterior derivative is a linear operator* d *that maps k-forms to $k + 1$-forms. A form ω is called* closed *if* $\mathrm{d}\omega = 0$.

Theorem 1. *From (Rudin et al., 1976; Lee, 2013), we know that $\mathrm{d} \circ \mathrm{d} \equiv 0$, which is dual to the fact that the boundary of the boundary of a manifold is empty from Stokes' theorem.*

3.2 Hamiltonian Vector Fields on Symplectic Cotangent Bundle

Definition 8 (symplectic and Riemannian)**.** *Let M be a smooth manifold.*

1. *symplectic form: A* 2-form *(so it is antisymmetric) ω^2 is said to be a* symplectic form *on M if it is closed, i.e. $\mathrm{d}\omega^2 = 0$, and it is* non-degenerate.
2. *metric tensor: A $(0, 2)$* tensor field *g is said to be a* Riemannian metric *on M if it is* non-degenerate *and symmetric.*

Remark 1. *A manifold equipped with a symplectic form ω^2 is called a* symplectic manifold*. Because of* non-degenerate*, on a symplectic manifold M, we can define an isomorphism between $\Gamma(TM)$ and $\Gamma(T^*M)$ by mapping a vector field $X \in \Gamma(TM)$ to a 1-form $\eta_V \in \Gamma(T^*M)$, where*

$$\eta_X(\cdot) \coloneqq \omega^2(\cdot, X)$$

Definition 9 (Hamiltonian flow and orbit)**.** *For a general symplectic manifold M with a symplectic form ω^2, if we have $H \in \mathcal{C}^\infty(M)$, then $\mathrm{d}H$ is a differential 1-form on M. We define the vector field called the* ***Hamiltonian flow*** *X_H* ***associated to the Hamiltonian*** *H, which satisfies that*

$$\eta_{X_H}(\cdot) = \mathrm{d}H(\cdot).$$

The integral curves of are called ***Hamiltonian orbits of*** *H:*

$$\dot{\gamma}(t) = (X_H)_{\gamma(t)} \quad \textit{for all } t \in I. \tag{7}$$

where $(X_H)_{\gamma(t)}$ is the tangent vector of X_H at $\gamma(t) \in M$.

The symplectic form are defined on M in Definition 8. However, from now on, we only consider the symplectic form defined on the cotangent bundle T^*M of a general manifold M.

Definition 10 (Poincaré 1-form and 2-form)**.** *On the cotangent bundle T^*M of a manifold M, we have a natural symplectic form, called the* Poincaré 1-form

$$\theta^1_{\text{Poincaré}} = p_i \mathrm{d}q^i. \tag{8}$$

Therefore, by the exterior derivative, we have the Poincaré 2-form

$$\omega^2_{\text{Poincaré}} = \mathrm{d}\theta^1 = \mathrm{d}(p_i \mathrm{d}q^i) = \sum_i \mathrm{d}p_i \wedge \mathrm{d}q^i$$

which is closed from Theorem 1. Poincaré 2-form *is a symplectic form on the cotangent bundle and cotangent bundles are the natural* ***phase spaces*** *in classical mechanics (De León & Rodrigues, 2011).*

For more general symplectic forms on the cotangent bundle, we can again use Theorem 1 to construct the closed 2-form from a 1-form which is potentially symplectic:

Corollary 1 ((Chen et al., 2021))**.** *According to Theorem 1, on the cotangent bundle T^*M of a manifold M, from a* 1*-form,*

$$\theta^1 = f_i \mathrm{d}q^i,$$

we derive a closed 2*-form using the exterior derivative* (35) *and its local representation is given by*

$$\omega^2 = \mathrm{d}\theta^1 = \mathrm{d}(f_i \mathrm{d}q^i) = \sum_{i<j} (\partial_i f_j - \partial_j f_i)\, \mathrm{d}p_i \wedge \mathrm{d}q^j$$

Corollary 2 ((Da Silva & Da Salva, 2008))**.** *According to* (5)*, from antisymmetry we have*

$$\omega^2(X_H, X_H) = 0 \tag{9}$$

which implies that hamiltonian vector fields preserve their hamiltonian functions H.

Remark 2. *In physics, hamiltonian functions are typically energy functions for a physical system. Corollary 2 indicates if the system updates its status according to the hamiltonian vector fields, the time-evolution of the system follows the law of conservation of energy.*

Definition 11 (Hamiltonian orbit generated from Poincaré 2-form (Lee, 2013))**.** *The Hamiltonian orbit generated by the Hamiltonian flow X_H on the cotangent bundle equipped with the Poincaré* 2*-form given in local coordinates (q, p) is given by*

$$\dot{q}^i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q^i}. \tag{10}$$

Definition 12 (cogeodesic orbits)**.** *If additionally M is equipped with a metric tensor g, i.e, if M is a pseudo-Riemannian manifold with metric g, and if we set the hamiltonian H on T^*M as*

$$H(q, p) = \frac{1}{2} g^{ij}(q) p_i p_j,$$

the Hamiltonian orbit, called the ***cogeodesic orbits of*** *(M, g), generated from Poincaré 2-form is*

$$\dot{q}^i = \frac{\partial H}{\partial p_i} = g^{ij} p_j, \quad \dot{p}_i = -\frac{\partial H}{\partial q^i} = -\frac{1}{2} \partial_i g^{jk} p_j p_k. \tag{11}$$

4 Proposed Framework

We now consider feature updating on graphs by making use of concepts from Section 3. We consider a graph as a discretization of a manifold M. For this manifold, we consider the cotangent bundle T^*M associated to M. The concept "momentum" from classical mechanics (De León & Rodrigues, 2011) can be viewed as a cotangent vector. For the node feature as a point on the manifold M, we associated to each point a "momentum" (i.e., a cotangent vector) as an external force to push the node feature updating on the manifold following Hamiltonian orbits.

In this paper, we consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a finite set $\mathcal{V}$ of vertices, together with a subset $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of edges. Let ${}^n q$ [2] be the d-dimensional input node feature for node n. In this work. we only consider transductive learning. We associate to each node an additional *learnable* vector ${}^n p \in \mathbb{R}^d$ which may be regarded as "momentum" of the node and is a cotangent vector of the manifold. In other words, we have $({}^n q, {}^n p) \in T^*M$ for each node. We next learn a Hamiltonian flow $X_{H_{\mathrm{net}}}$ associated to the Hamiltonian H_{net}[3] on the cotangent bundle T^*M, where T^*M is taken as a symplectic manifold equipped with different symplectic forms ω^2 and

$$H_{\mathrm{net}} : T^*M \to \mathbb{R}. \tag{12}$$

Our model is called *HamGNN* to indicate the Hamiltonian flow for node feature updating on the manifold. Specifically, we learn the Hamiltonian orbits of H_{net} which are integral curves starting from node features. The curves' evolutions are determined by both the node features and the "momentums".

[2]We put the node index to the left of the variable to distinguish it from the manifold dimension index.

[3]The lower index of a function F_{net} indicates the function F is parameterized by a neural network.

4.1 DIFFERENT HAMILTONIAN FLOW

We next give different symplectic forms ω^2 and H_{net} from which the corresponding Hamiltonian flow and its variations are obtained in our graph neural network model. The node features are then updated along the Hamiltonian flow, where each Hamiltonian orbit (7) is the solution of

$$({}^n q(t), {}^n p(t)) = {}^n\gamma(t) = (X_{H_{\text{net}}})_{{}^n\gamma(t)} \quad \text{for all } t \in I, \tag{13}$$

with the initial node feature ${}^n\gamma(0) = ({}^n q, {}^n p)$ at $t = 0$ with learnable ${}^n p$ for each node n. In the following, we will however omit the node index in $({}^n q, {}^n p)$ whenever possible to simplify the notations. The solution of (13) is given by the differential equation solver (Chen et al., 2018a). We take the solution $(q(T), p(T))$ at time T, and project it back to M, as $\pi({}^n q(T), {}^n p(T)) = {}^n q(T)$ (See Definition 5). The operation is conducted within one layer, and we call it the *Hamiltonian layer*.

From Corollary 2, we know if the Hamiltonian H_{net} indicates some energy function, the feature updating through the orbit indicates that the energy keeps fixed during the feature updating along the curve. Furthermore, we also relax the strict Hamiltonian orbit and add some system biases which indicate additional energy has been enforced during the node feature updating.

4.1.1 LEARNABLE ω^2_{net} AND H_{net}

From Corollary 1, if we take the exterior derivative of a 1-form θ^1_{net}, the induced ω^2_{net} is guaranteed to be closed. In (Chen et al., 2021), the authors follow a similar principle to learn from physical systems output like the mass-spring system and double pendulum. More specifically, we have

$$\theta^1_{\text{net}} = f_{i,\text{net}} \mathrm{d}q^i,$$

where $f_{i,\text{net}}$ is the output's i-th component of the neural network parameterized function

$$f_{\text{net}} : M \to \mathbb{R}^d.$$

From Corollary 1, we have the following closed 2-form

$$\omega^2 = \mathrm{d}\theta^1_{\text{net}} = \mathrm{d}(f_{i,\text{net}} \mathrm{d}q^i) = \sum_{i<j} (\partial_i f_{j,\text{net}} - \partial_j f_{i,\text{net}}) \, \mathrm{d}p_i \wedge \mathrm{d}q^j,$$

from which the Hamiltonian orbit is given by (Chen et al., 2021) as follows,

$$\left(\dot{q}^i, \dot{p}^i\right) = W^{-1}(q,p) \nabla H_{\text{net}}(q,p), \tag{14}$$

where the skew-symmetric $2d \times 2d$ matrix W, where each elements is written in terms of $(\partial_i f_{j,\text{net}} - \partial_j f_{i,\text{net}})$, is given in (48) due to space constraints.

4.1.2 POINCARÉ 2-FORM $\omega^2_{\text{Poincaré}}$ AND LEARNABLE H_{net}

Different from Section 4.1.1 where ω^2_{net} is learnable, we use the Hamiltonian orbit (10) derived from the fixed Poincaré 2-form $\omega^2_{\text{Poincaré}}$ on T^*M as follows

$$\dot{q}^i = \frac{\partial H_{\text{net}}}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H_{\text{net}}}{\partial q^i}. \tag{15}$$

4.1.3 POINCARÉ 2-FORM $\omega^2_{\text{Poincaré}}$ AND LEARNABLE CONVEX H_{net}

If H_{net} is restricted to strictly convex functions, the Hamiltonian formalism can degenerate to a *Lagrangian formalism* (De León & Rodrigues, 2011) through the Legendre transformation

$$H(q,p) = p_i \dot{q}^i - L(q, \dot{q}) \text{ with } \dot{q} = \dot{q}(p) \text{ such that } p = \frac{\partial L}{\partial \dot{q}}$$

where L is a convex Lagrangian in the following functional

$$S(q) = \int_a^b L(q(t), \dot{q}(t)) \mathrm{d}t. \tag{16}$$

According to (Calkin, 1996), a path $q(t)$ on M, with two fixed endpoints at $t = a$ and $t = b$, is a stationary point (i.e., a curve) of the functional S iff the following equation

$$\frac{\partial L}{\partial q^i}(q(t), \dot{q}(t)) - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}^i}(q(t), \dot{q}(t)) = 0$$

is correct, which is called Euler–Lagrange equation, and iff (10) is correct (Calkin, 1996). So to some extent, the solution to the following restricted Hamiltonian differential equation

$$\dot{q}^i = \frac{\partial H_{\mathrm{net}}}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H_{\mathrm{net}}}{\partial q^i} \quad \text{s. t. } H_{\mathrm{net}} \text{ is convex} \tag{17}$$

will *potentially* (strictly speaking, we only have a stationary curve) minimize the dual optimization energy function $S(q)$ for a curve $q(t)$ on the base manifold M. To guarantee that H_{net} is convex, we follow the convention in (Amos et al., 2017) with non-negative layer weight from the second layer in H_{net}, and all activation functions in H_{net} are convex and non-decreasing. This network design is proved to be able to approximate any convex functions as shown in (Chen et al., 2018b).

4.1.4 Poincaré 2-form $\omega^2_{\mathrm{Poincaré}}$ and Learnable H_{net} with Relaxation

Furthermore, different from Section 4.1.2, we enforce additional system biases along the curve as follows,

$$\dot{q}^i = \frac{\partial H_{\mathrm{net}}}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H_{\mathrm{net}}}{\partial q^i} + f_{\mathrm{net}}(q). \tag{18}$$

Intuitively speaking, instead of keeping the energy during the feature updating along the Hamiltonian orbit, we now also include additional energy during the node feature updating.

4.1.5 Poincaré 2-form $\omega^2_{\mathrm{Poincssaré}}$ and Learnable Metric g_{net}

Different from the above symplectic forms on the cotangent bundle T^*M, we now additionally assign the metric g to the base manifold M. The metric g measures the norms of tangent vectors and the angles between them, and in some sense, the "shape structure" of the manifold M is only available if we equipped M with a metric. However, according to the (11), the space complexity becomes order d^3 due to the partial derivative of g's output being a $d \times d$ matrix. We therefore only consider *diagonal metrics* to mitigate the space complexity. More specifically, we now define diagonal g as:

$$g_{\mathrm{net}}(q) = \mathrm{diag}([\underbrace{-1, \ldots, -1}_{r}, \underbrace{1, \ldots, 1}_{s}] \odot h_{\mathrm{net}}(q)) \tag{19}$$

where $h_{\mathrm{net}} : \mathbb{R}^d \mapsto \mathbb{R}^d$ consists of non-linear trainable layers and $\odot$ denotes element-wise multiplication. To ensure non-degeneracy the metric, the output of h_{net} is set to be away from 0 with the final activation function of it being *trictly positive*. The vector $[-1, \ldots, -1, 1, \ldots, 1]$ controls the signature (r, s) of the metric g with $r + s = d$, where r and s are the number of -1s and 1s, respectively. The signature (r, s) of the metric is set to be a hyperparameter to the Hamiltonian orbits:

$$\dot{q}^i = g^{ij}_{\mathrm{net}} p_j, \quad \dot{p}_i = -\frac{1}{2}\partial_i g^{jk}_{\mathrm{net}} p_j p_k. \tag{20}$$

The projection of cogeodesic orbits under π to the base manifold M is called *geodesic* which generalizes the notion of a "straight or shortest line" to a manifold with the curve length Lagrangian

$$L(q, \dot{q}) = g_{ij}(q)\dot{q}^i\dot{q}^j$$

in the optimization functional (16). Intuitively speaking, the node features evolve through the "shortest" curves on the manifold. The exponential map used in Hyperbolic GNNs (Chami et al., 2019) is essentially the *geodesic* curve on a hyperbolic manifold with an explicit formulation due to the hyperbolic assumption. We do not enforce any assumption here and let the model be feasible enough to learn the embedding geometry.

4.2 Neighborhood Aggregation

After the node features update along the Hamiltonian flows that have been introduced in Section 4.1, we then perform the neighborhood aggregation to the features $\{{}^n q^{(\ell)}(T)\}_{n \in \mathcal{V}}$ where ℓ indicates the

ℓ-th layer. Let $\mathcal{N}(n) = \{m : (n, m) \in \mathcal{E}\}$ denote the set of neighbors of node $n \in \mathcal{V}$. We only perform a simple yet efficient aggregation for node n as follows:

$$ {}^{n}q^{(\ell+1)} = {}^{n}q^{(\ell)}(T) + \frac{1}{|\mathcal{N}(n)|} \sum_{m \in \mathcal{N}(n)} {}^{m}q^{(\ell)}(T). \tag{21} $$

4.3 Model Architecture

We summarize HamGNN model architecture in Fig. 1, where we stack up multiple Hamiltonian layers together with graph convolutions to perform feature transformation and aggregation. Since the input node features in the datasets are sparse, an FC layer is first applied to compress the raw feature. Note that although the Hamiltonian orbits only depend on their starting node features, these features ready contain information aggregated from its neighbors because of the feature aggregation operations. Therefore each node is locally embedded into a manifold M that depends on its neighbors.

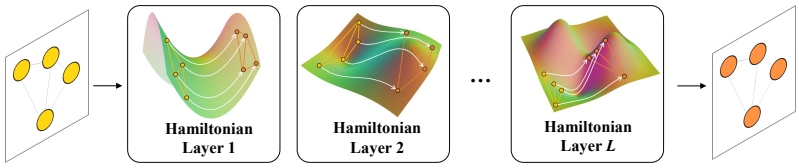

Figure 1: HamGNN model architecture.

5 Experiments

In this section, we implement the proposed HamGNNs with different settings as shown in Section 4.1 and Appendix B on the transductive node classification task. We select the citation networks Cora (McCallum et al., 2004), Citeseer (Sen et al., 2008), Pubmed (Namata et al., 2012), and two low hyperbolicity datasets (Chami et al., 2019), named Disease and Airport, as the benchmark datasets. To fairly compare the performance of the proposed HamGNN, we select several popular GNN models as the baseline, including the Euclidean GNN: GCN (Kipf & Welling, 2017a), GAT (Veličković et al., 2018), SAGE (Hamilton et al., 2017), and SGC (Wu et al., 2019), which build on Euclidean space; the hyperbolic GNNs (Chami et al., 2019; Liu et al., 2019): HGNN, HGCN, and HGAT, which learn the node embeddings on hyperbolic space and also the GIL Zhu et al. (2020) which learns a weighting between both spaces. To further demonstrate the advantage of HamGNN, we also include one vanilla ODE system, whose formulation is given in Appendix D.1 without the crucial Hamiltonian layer. Due to space constraints, we refer the readers to Appendix C for the datasets and implementation details.

5.1 Performance Results and Ablation Studies

The node classification performance on the benchmark datasets using the baseline models and the proposed HamGNNs with different settings is shown in Table 1. We clearly observe that the best and the second-best results of HamGNNs outperform *all the baselines on all datasets* and especially outperform all the baselines by a significant margin of 3% accuracy on the Airport dataset. According to (Chami et al., 2019; Liu et al., 2019), the Airport dataset has a tree-like structure that cannot be properly embedded into Euclidean spaces. Our HamGNNs under all the different settings however can have a better performance on this dataset due to the learnable embedding manifold without many constraints. The comprehensive surpass of our HamGNNs on all the datasets over all the baselines has demonstrated the advantage of including the Hamiltonian flows on the manifold to update the node features. Furthermore, the comparison between our HamGNNs with the GNN with just a vanilla ODE in (51) without any Hamiltonian mechanism has indicated the indispensability of the Hamiltonian layer.

We then compare between HamGNNs under different settings. First of all, there is no setting under which the HamGNN gets the best results over all the datasets, which suggests that for different datasets, Hamiltonian systems may behave differently on different graphs. We then compare the HamGNN using (14) and (15). The difference between those two settings is that the symplectic form ω^2 in (15) is set to be the special Poincaré 2-form while in (14), the symplectic form is a learnable one. We however observe that the two HamGNNs achieve similar performance and the more flexible symplectic form does not improve the model performance. This may be because of the fundamental Darboux theorem Theorem 3 in symplectic geometry which states that we can always find a Darboux

Method	Disease	Airport	Pubmed	Citeseer	Cora
MLP	50.00±0.00	76.96±1.77	71.95±1.38	58.10±1.87	57.15±1.15
HNN	56.40±6.32	80.49±1.54	71.60±0.47	55.13±2.04	58.03±0.55
GCN	81.10±1.33	82.25±0.56	77.83±0.77	71.78±0.34	80.29±2.29
GAT	87.01±2.77	92.99±0.83	77.58±0.81	68.07±1.31	80.33±0.61
SAGE	81.60±7.68	81.97±0.85	77.63±0.15	65.90±2.32	74.50±0.88
SGC	82.78±0.93	81.40±2.21	76.83±1.11	70.88±1.32	81.98±1.71
HGNN	80.51±5.70	84.54±0.72	76.65±1.38	69.43±0.99	79.53±0.98
HGCN	89.87±1.13	85.35±0.65	76.38±0.81	65.80±2.04	78.70±0.96
HGAT	88.68±3.36	87.50±0.99	78.00±0.50	69.20±0.96	80.88±0.75
GIL	90.78±0.45	91.52±1.74	77.76±0.57	71.10±1.24	82.10±1.12
Vanilla ODE in (51)	71.81±18.85	90.34±0.67	73.30±3.31	56.60±1.26	68.38±1.18
HamGNN (14)	88.35±1.51	93.66±0.16	*78.60±0.32*	71.52±1.41	81.24±0.59
HamGNN (15)	88.74±1.17	95.11±0.40	78.18±0.54	71.48±1.42	81.52±1.27
HamGNN (17)	84.57±5.78	93.28±1.40	**78.83±0.46**	*72.00±0.73*	81.84±0.88
HamGNN (18)	87.48±5.90	95.46±0.68	78.30±0.34	**72.38±0.85**	*81.56±0.97*
HamGNN (20)	*91.26±1.40*	95.50±0.48	78.08±0.48	70.12±0.86	**82.16±0.80**
HamGNN (49)	91.18±0.99	*95.80±0.19*	77.90±0.49	69.18±1.63	80.90±0.35
HamGNN (50)	**91.50±2.07**	**95.99±0.13**	78.26±0.64	69.10±1.95	80.10±1.56

Table 1: Node classification accuracy(%). The best and the second-best result for each criterion are highlighted in **bold** and *underlined and italic* respectively.

Dataset	Models	3 layers	5 layers	10 layers	20 layers
Cora	GCN	80.29±2.29	69.87±1.12	26.50±4.68	23.97±5.42
	HGCN	78.70±0.96	38.13±6.20	31.90±0.00	26.23±9.87
	HamGNN (15)	81.52±1.27	**81.58±0.73**	79.00±21.17	76.20±0.13
	HamGNN (17)	**81.84±0.88**	81.08±0.16	**81.40±0.44**	**80.58±0.30**

Table 2: Node classification accuracy(%) when increasing the number of layers on the Cora dataset.

coordinate system to give any symplectic form the *Poincaré* 2*-form*. The feature compressing FC to raw graph node feature has the network capacity to approximate the Darboux coordinate map. The chosen Hamiltonian function H therefore plays a more important role in the model performance. For example, the convex H_{net} in (17) achieves the best performance on the Citeseer dataset, while the cogeodesic orbit based HamGNN using (20) has the best performance on the Cora dataset.

5.2 Resilience to Over-Smoothing

As shown in Table 2, we clearly observe that HamGNNs still retain the node classification ability if more Hamiltonian layers are stacked, while the classification accuracy of other GNNs has dropped a lot. One reason for this stability against over-smoothing is that the node feature energy indicated by Hamiltonian H_{net} is fixed during the feature updating along the Hamiltonian orbit. Furthermore, we observe that if the H_{net} in (17) is convex, HamGNN can even retain its classification ability better than vanilla H_{net} in (15). The reason has been indicated in Section 4.1.3 since now the Hamiltonian formalism degenerates to a Lagrangian formalism with a possible minimization of the dual energy functional (16). In physics, lower energy in most cases indicates a more stable system equilibrium.

More Ablation Studies and Experiments: See Appendix D.

6 Conclusion

In this paper, we linked the notion of symplectic manifolds and the Hamiltonian vector fields to GNNs. Based on different Hamiltonian flows, the node features update stably. We empirically show that our approach achieves better node classification accuracy than popular state-of-the-art GNNs. Furthermore, the oversmoothing of GNNs can be mitigated if the node features evolve through the Hamiltonian flows.

REFERENCES

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.

Haitham Ashoor, Xiaowen Chen, Wojciech Rosikiewicz, Jiahui Wang, Albert Cheng, Ping Wang, Yijun Ruan, and Sheng Li. Graph embedding and unsupervised learning predict genomic sub-compartments from hic chromatin interaction data. *Nat. Commun.*, 11, 2020.

Melvin G Calkin. *Lagrangian and Hamiltonian mechanics*. World Scientific, 1996.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Proc. Advances Neural Inf. Process. Syst.*, 2019.

Ricky T. Q. Chen. torchdiffeq, 2018. URL `https://github.com/rtqichen/torchdiffeq`.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proc. Advances Neural Inf. Process. Syst.*, 2018a.

Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal control via neural networks: A convex approach. *arXiv preprint arXiv:1805.11835*, 2018b.

Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Neural symplectic form: Learning hamiltonian equations on general coordinate systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ana Cannas Da Silva and A Cannas Da Salva. *Lectures on symplectic geometry*, volume 3575. Springer, 2008.

Manuel De León and Paulo R Rodrigues. *Generalized Classical Mechanics and Field Theory: a geometrical approach of Lagrangian and Hamiltonian formalisms involving higher order derivatives*. Elsevier, 2011.

Marián Fecko. *Differential geometry and Lie groups for physicists*. Cambridge university press, 2006.

Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison Wesley, 3 edition, 2001.

Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Proc. Advances Neural Inf. Process. Syst.*, 2019.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34 (1):1–23, December 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proc. Advances Neural Inf. Process. Syst.*, 2017.

Philip Hartman. *Ordinary differential equations*. SIAM, 2002.

Yifei Huang, Yaodong Yu, Hongyang Zhang, Yi Ma, and Yuan Yao. Adversarial robustness of stabilized neural ode might be from obfuscated gradients. In Joan Bruna, Jan Hesthaven, and Lenka Zdeborova (eds.), *Proc. Mathematical and Scientific Machine Learning Conference*, pp. 497–515, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learn. Representations*, pp. 1–14, 2017a.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learn. Representations*, 2017b.

Wilhelm PA Klingenberg. *Riemannian geometry*, volume 1. Walter de Gruyter, 2011.

John M Lee. *Introduction to smooth manifolds*. Springer, London, 2013.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Proc. Advances Neural Inf. Process. Syst.*, 2019.

Andrew McCallum, Kamal Nigam, Jason D. M. Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2004.

Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *Workshop Mining and Learn. with Graphs*, 2012.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. Int. Conf. Learn. Representations*, pp. 1–12, 2018.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proc. Int. Conf. Mach. Learn.*, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. 32(1):4–24, 2021.

Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, 36(4):1241–1251, 2019.

Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proc. Web Conf.*, 2021.

Ziwei Zhang, Peng Cui, and Wenwu Zhu. Deep learning on graphs: A survey. 34(1):249–270, Jan 2022.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *Proc. Int. Conf. Learn. Representations*, 2020.

Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. Graph geometry interaction learning. In *Proc. Advances Neural Inf. Process. Syst.*, 2020.

A Differential Geometry and Hamiltonian System

In this supplementary material, we first review some important concepts from differential geometry that are highly related to our work. We hope that this overview makes the paper more accessible for readers from the graph learning community. Some concepts also appear in Section 3 and are repeated here for completeness.

A.1 Manifold, Bundles, and Fields

Roughly speaking, a manifold is a topological space that *locally* looks like Euclidean space. More strictly speaking, a topological space $(M, \mathcal{O})$, where $\mathcal{O}$ is the collection of open sets on space M, is called a d-dimensional manifold if for every point $x \in M$, we can find an open neighborhood $U \in \mathcal{O}$ for x and a coordinate map

$$q : U \to q(U) \subseteq \mathbb{R}^d$$

that is a *homeomorphism*, where $\mathbb{R}^d$ is the d-dimensional Euclidean space with the standard topology. (U, q) is called a *chart* of the manifold, which gives us a numerical representation for a local area in M. In this work, we only consider smooth manifolds (Lee, 2013) that any two overlapped charts are

smoothly compatible. The set of all smooth functions from M to $\mathbb{R}$ is denoted as $\mathcal{C}^\infty(M)$. On top of the smooth manifolds, we can define other related manifolds like the tangent or cotangent bundles and the more general tensor bundles. From bundles, we can define the vector or covector fields and the more general tensor fields. In this work, we mainly consider the manifold with the 2-forms that are special $(0, 2)$ smooth tensor fields with antisymmetric constraints. More specifically, we mainly consider the *symplectic form* (Lee, 2013) with some other light shed on the metric tensor which is another type of $(0, 2)$ smooth tensor field.

Definition 13 (tangent vector and tangent space)**.** *Let $\gamma : I \to M$ be a smooth curve through $x \in M$ s.t. $\gamma(0) = x$ and I is an interval neighborhood of 0. The* tangent vector *is a directional derivative operator at x along γ that is the* linear map

$$\begin{aligned} v_{\gamma,x} :\ & \mathcal{C}^\infty(M) \xrightarrow{\sim} \mathbb{R} \\ & f \mapsto (f \circ \gamma)'(0). \end{aligned}$$

We also call the directional derivative operator as "velocity" of γ at 0 and denote it as $\dot{\gamma}(0)$. (More generally, the 'velocity" of γ at t is denoted at $\dot{\gamma}(t)$ which is a tangent vector at point $\gamma(t) \in M$ from the curve reparametrization trick (Fecko, 2006)). Correspondingly, the tangent space *to M at x is the* vector space *over $\mathbb{R}$ with the underlying set*

$$T_x M \coloneqq \{v_{\gamma,x} \mid \gamma \text{ is a smooth curve and } \gamma(0) = x\}. \tag{22}$$

Note for $f \in \mathcal{C}^\infty(M)$, using the chart (U, q) with $x \in U$, we have the local representation

$$\begin{aligned} v_{\gamma,x}(f) &\coloneqq (f \circ \gamma)'(0) \\ &= (f \circ q^{-1} \circ q \circ \gamma)'(0) \\ &= (q^i \circ \gamma)'(0) \cdot \partial_i \big(f \circ q^{-1}\big)\big|_{q(x)}, \end{aligned}$$

where q^i is the i-th component of q, $\circ$ is the function composition and $(\cdot)|_{q(x)}$ means evaluating $(\cdot)$ at $q(x)$. Therefore for a local chart around x, we have a basis of $T_x M$ as

$$\left\{ \left(\frac{\partial}{\partial q^1}\right)_x, \ldots, \left(\frac{\partial}{\partial q^d}\right)_x \right\}$$

and we call it the chart induced basis*, where $\left(\frac{\partial}{\partial q^1}\right)_x \coloneqq \partial_i (f \circ q^{-1})|_{q(x)}$.*

Definition 14 (tangent bundle)**.** *Given a smooth manifold M, the* tangent bundle *of M is the* disjoint union *of all the tangent spaces to M, i.e., we have*

$$TM \coloneqq \coprod_{x \in M} T_x M, \tag{23}$$

equipped with the canonical projection map

$$\begin{aligned} \tilde{\pi} :\ & TM \to M \\ & X \mapsto \tilde{\pi}(X), \end{aligned}$$

where $\tilde{\pi}(X)$ sends each vector in $T_x M$ to the point x at which it is tangent, and $\coprod$ is the disjoint union. Furthermore, the tangent bundle[4] *is a $2d$-dimensional manifold. If $X \in \tilde{\pi}^{-1}(U) \subseteq TM$ with a local chart (U, q) on M s.t. $x \in U$, then $X \in T_{\tilde{\pi}(X)} M$ from the definition. Since $\tilde{\pi}(X) \in U$, X can be written in terms of the* chart induced basis*:*

$$X = \tilde{p}^i(X) \left(\frac{\partial}{\partial q^i}\right)_{\tilde{\pi}(X)}, \tag{24}$$

where $\tilde{p}^1, \ldots, \tilde{p}^d$ are smooth scalar functions. We can then define the following map as a local chart for the manifold TM induced from the chart (U, q) on M:

$$\begin{aligned} \xi :\ & \tilde{\pi}^{-1}(U) \to q(U) \times \mathbb{R}^d \subseteq \mathbb{R}^{2d} \\ & X \mapsto (q^1(\tilde{\pi}(X)), \ldots, q^d(\tilde{\pi}(X)), \tilde{p}^1(X), \ldots, \tilde{p}^d(X)), \end{aligned}$$

and the topological structure on TM is derived from the initial topology to ensure continuity.

[4] In this paper, we may just use the word "bundle" to indicate the total space in the bundle.

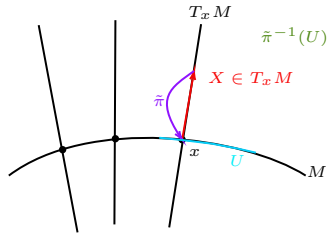

Figure 2: Visualization of a 1-dimensional manifold and its tangent bundle.

Definition 15 (vector field)**.** *A vector field on M is a smooth section (Lee, 2013) of the tangent bundle, i.e. a smooth map $\sigma : M \to TM$ such that $\tilde{\pi} \circ \sigma = id_M$, where id_M is the identity map on M.*

$$\begin{array}{c} TM \\ \sigma \uparrow \downarrow \tilde{\pi} \\ M \end{array} \tag{25}$$

We denote the set of all vector fields on M by $\Gamma(TM)$:

$$\Gamma(TM) := \{\sigma : M \to TM \mid \sigma \text{ is smooth and } \tilde{\pi} \circ \sigma = id_M\}. \tag{26}$$

Definition 16 (cotangent vector, cotangent bundle, dual basis, and covector field)**.** *For the vector space T_xM, a continuous linear functional from T_xM to $\mathbb{R}$ is called a cotangent vector at x. The set of all such linear maps is denoted as T_x^*M which is the dual vector space of T_xM. For $f \in \mathcal{C}^\infty(M)$, at each point x, we define the following linear operator in T_x^*M*

$$\begin{aligned} (\mathrm{d}f)_x : T_xM &\to \mathbb{R} \\ X_x &\mapsto (\mathrm{d}f)_x(X_x) := X_x(f). \end{aligned}$$

Given a chart (U, q) with $x \in U$ and its chart induced basis*, the* dual basis *for the dual space T_x^*M is the set*

$$\left\{ \left(\mathrm{d}q^1\right)_x, \ldots, \left(\mathrm{d}q^d\right)_x \right\},$$

where we have $(\mathrm{d}q^a)_x \left(\left(\frac{\partial}{\partial q^b} \right)_x \right) = \left(\frac{\partial}{\partial q^b} \right)_x (q^a) = \delta_b^a$ with $\delta_b^a = 1$ iff $a = b$ and $\delta_b^a = 0$ otherwise. We call it the chart induced dual basis *. Analogous to the above definition of the vector field, we can define the cotangent bundle of M as*

$$T^*M := \coprod_{x \in M} T_x^*M \tag{27}$$

which is again a $2d$-dimensional manifold equipped with the canonical projection map

$$\begin{aligned} \pi : T^*M &\to M \\ \omega &\mapsto \pi(\omega), \end{aligned} \tag{28}$$

where $\pi(\omega)$ sends each vector in T_x^*M to the point x at which it is cotangent. If $\omega \in \pi^{-1}(U) \subseteq T^*M$ with a local chart (U, q) s.t. $x \in U$, then $\omega \in T_{\pi(\omega)}^*M$ from the definition. Since $\pi(\omega) \in U$, ω can be written in terms of the chart induced dual basis*:*

$$\omega = p_i(\omega)(\mathrm{d}q^i)_{\pi(\omega)}, \tag{29}$$

where $p_1, \ldots, p_d$ are smooth scalar functions. We can then define the following map as a local chart for manifold T^*M induced from the chart (U, q) on M:

$$\begin{aligned} \xi : \pi^{-1}(U) &\to q(U) \times \mathbb{R}^d \subseteq \mathbb{R}^{2d} \\ \omega &\mapsto (q^1(\pi(\omega)), \ldots, q^d(\pi(\omega)), p_1(\omega), \ldots, p_d(\omega)), \end{aligned}$$

and the topological structure on T^*M is derived from the initial topology to ensure continuity. The covector fields are smooth sections of T^*M. The set of all covector fields is denoted as $\Gamma(T^*M)$.

Definition 17 (tensor field and 2-form)**.** *The tensor field can be defined using the smooth sections on tensor bundles analogously to the vector fields or the covector fields. We refer readers to (Lee, 2013) for more details. Here, instead of a rigorous definition, we show some basic properties of the tensor fields. For a (r, s) tensor field τ, at $x \in M$, it is a multilinear map*

$$\tau_x : \underbrace{T_x^*M \times \cdots \times T_x^*M}_{r \text{ copies}} \times \underbrace{T_xM \times \cdots \times T_xM}_{s \text{ copies}} \to \mathbb{R}.$$

The differential k-form ω is the $(0, k)$ tensor field that admits alternating (Lee, 2013). Specifically, for the 2-form ω, at each point x, ω_x is a antisymmetric $(0, 2)$ *tensor*

$$\omega_x : T_xM \times T_xM \to \mathbb{R} \text{ s.t. } \omega_x(X_1, X_2) = -\omega_x(X_2, X_1) \,\forall\, X_1, X_2 \in T_xM \tag{30}$$

which in other words, ω satisfies

$$\omega : \Gamma(TM) \times \Gamma(TM) \to \mathcal{C}^\infty(M) \text{ s.t. } \omega(X, Y) = -\omega(Y, X) \,\forall\, X, Y \in \Gamma(TM) \tag{31}$$

In local chart representation, we have that every k-form ω can be expressed locally *on U as*

$$\omega = \omega_{a_1 \cdots a_k} \, \mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}, \tag{32}$$

where $\omega_{a_1 \cdots a_k} \in \mathcal{C}^\infty(U)$, $1 \leq a_1 < \cdots < a_k \leq \dim M$ are increasing sequences and $\mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}$ is the wedge product (Lee, 2013). Here we could abstractly view the set $\{\mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}\}_{a_1, \ldots, a_k}$, with a_i enumerated from 1 to d, abstractly as a basis without more illustrations of the wedge product (We refer readers to (Lee, 2013) for more details).

Definition 18 (integral curve)**.** *Given a vector field X on M, an integral curve of X is a differentiable curve $\gamma : I \to M$, where $I \subseteq \mathbb{R}$ is an interval, whose velocity at each point is equal to the value of X at that point:*

$$\dot{\gamma}(t) = X_{\gamma(t)} \quad \textit{for all } t \in I. \tag{33}$$

If $0 \in J$, the point $\gamma(0)$ is called the starting point of γ. From Picard's theorem (Hartman, 2002), we know that locally we always have an interval I on which the solution exists and is necessarily unique.

Definition 19 (exterior derivative and closed form)**.** *The exterior derivative is a linear operator that maps k-forms to $k + 1$-forms. In local chart representation, we have that if ω is a k-form on M with the local representation as*

$$\omega = \omega_{a_1 \cdots a_k} \mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}. \tag{34}$$

Then, we have the exterior derivative

$$\begin{aligned} \mathrm{d}\omega &= \mathrm{d}\omega_{a_1 \cdots a_k} \wedge \mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k} \\ &= \partial_b \omega_{a_1 \cdots a_k} \mathrm{d}x^b \wedge \mathrm{d}x^{a_1} \wedge \cdots \wedge \mathrm{d}x^{a_k}. \end{aligned} \tag{35}$$

A form ω is called closed *if $\mathrm{d}\omega = 0$.*

Theorem 2. *From (Rudin et al., 1976; Lee, 2013), we know that*

$$\mathrm{d} \circ \mathrm{d} \equiv 0, \tag{36}$$

which is a dual statement that the boundary of the boundary of a manifold is empty from Stokes' theorem.

A.2 Hamiltonian Vector Fields on Symplectic Cotangent Bundle

Definition 20 (symplectic vs. Riemannian)**.** *Let M be a smooth manifold.*

1. *symplectic form: A* 2-form *(so it is antisymmetric) ω is said to be a* symplectic form *on M if it is closed, i.e,*

$$\mathrm{d}\omega = 0,$$

and it is non-degenerate*, i.e,*

$$(\forall\, Y \in \Gamma(TM) : \omega(X, Y) = 0) \Rightarrow X = 0. \tag{37}$$

2. *metric tensor: A* $(0,2)$ tensor field g *is said to be a* Riemannian metric *on* M *if it is* non-degenerate *and symmetric at each* x*, i.e.,*

$$g_x : T_xM \times T_xM \to \mathbb{R} \text{ s.t. } g_x(X_1, X_2) = g_x(X_2, X_1) \,\forall\, X_1, X_2 \in T_xM \tag{38}$$

Remark 3. *A manifold equipped with a symplectic form* ω *is called a* symplectic manifold*, while a manifold equipped with a metric tensor* g *is called a* pseudo-Riemannian manifold*. The Riemannian metric* g *is a* $(0,2)$*-tensor field measuring the norms of tangent vectors and the angles between them. To some extent, the "shape structure" of the manifold* M *is only available if we equipped* M *with a metric* g*.*

- *From the above definition, we know the symplectic form and the metric tensor are both nondegenerate bilinear* $(0,2)$ *tensor fields. One difference is the symplectic form is anti-symmetric while the metric tensor is symmetric. At each point* $x \in M$*, if we use the local chart coordinate representation, the* $(0,2)$ *tensor can be represented as the following matrix multiplication*

$$\tilde{p}^{\intercal} W_x \tilde{p}$$

  where $d \times d$ *matrix* W *is symmetric for metric tensor and is skew-symmetric for symplectic form.* $\tilde{p}^{\intercal}$ *is the transpose of the velocity representation (which is a numerical vector) in a local chart.*

- *Because of* non-degenerate*, on a symplectic manifold* M*, we can define an isomorphism between* $\Gamma(TM)$ *and* $\Gamma(T^*M)$ *by mapping a vector field* $X \in \Gamma(TM)$ *to a 1-form* $\eta_V \in \Gamma(T^*M)$*, where*

$$\eta_X(\cdot) \coloneqq \omega^2(\cdot, X) \tag{39}$$

  Similarly, on a pseudo-Riemannian manifold, we can define an isomorphism between $\Gamma(TM)$ *and* $\Gamma(T^*M)$ *by mapping a vector field* $X \in \Gamma(TM)$ *to a 1-form* $\in \Gamma(T^*M)$*, where*

$$\alpha_g : TM \longrightarrow T^*M. \tag{40}$$

  In a local chart coordinate representation, $\alpha_g = g_{ij}$ *and its inverse* $\alpha_g^{-1} = g^{ij}$ *with* $\sum_{j=1}^m g_{ij}g^{jk} = \delta_i^k$ *and* $\delta_i^k = 1$ *iff* $i = k$ *and* 0 *otherwise. Note the components of the metric and the inverse metric are all taken in a given chart without explicitly mentioning them.*

Definition 21 (Hamiltonian flow and orbit)**.** *For a general symplectic manifold* M *with a symplectic form* ω^2*, if we have* $H \in \mathcal{C}^\infty(M)$*, then* $\mathrm{d}H$ *is a differential* 1*-form on* M*. We define the vector field called the* ***Hamiltonian flow*** X_H ***associated to the Hamiltonian*** H*, which satisfies that*

$$\eta_{X_H}(\cdot) = \mathrm{d}H(\cdot).$$

The integral curves of are called ***Hamiltonian orbits of*** H*:*

$$\dot{\gamma}(t) = (X_H)_{\gamma(t)} \quad \textit{for all } t \in I. \tag{41}$$

where $(X_H)_{\gamma(t)}$ *is the tangent vector at* $\gamma(t) \in M$*.*

Definition 22 (Poincaré 1-form and 2-form)**.** *On the cotangent bundle* T^*M *of a manifold* M*, we have a natural symplectic form, called the* Poincaré 1-form

$$\theta^1_{\text{Poincaré}} = p_i \mathrm{d}q^i. \tag{42}$$

Therefore, by the exterior derivative, we have the Poincaré 2-form

$$\omega^2_{\text{Poincaré}} = \mathrm{d}\theta^1 = \mathrm{d}(p_i \mathrm{d}q^i) = \sum_i \mathrm{d}p_i \wedge \mathrm{d}q^i \tag{43}$$

which is closed from (36)*. Therefore* Poincaré 2-form *is a symplectic form on the cotangent bundle and cotangent bundles are the natural* ***phase spaces*** *of classical mechanics (De León & Rodrigues, 2011).*

For more general symplectic forms on the cotangent bundle, we can again use (36) to construct the closed 2-form from a 1-form which is potentially symplectic:

Corollary 3 ((Chen et al., 2021))**.** *According to* (36)*, on the cotangent bundle T^*M of a manifold M, from a* 1*-form,*

$$\theta^1 = f_i \mathrm{d}q^i,$$

we derive a closed 2*-form using the exterior derivative* (35)*, and its local representation is given by the following*

$$\omega^2 = \mathrm{d}\theta^1 = \mathrm{d}(f_i \mathrm{d}q^i) = \sum_{i<j} (\partial_i f_j - \partial_j f_i)\, \mathrm{d}p_i \wedge \mathrm{d}q^j$$

Remark 4. *Note, strictly speaking, we only can get the necessary "closed" condition for ω^2 to be potentially symplectic. However, it is enough for our use in our proposed framework.*

We now state one of the most fundamental results in symplectic geometry that links the general symplectic form to the special Poincaré 2-form.

Theorem 3 (Darboux (Lee, 2013))**.** *Let $(\tilde{M}, \tilde{\omega}^2)$ be a $2d$-dimensional symplectic manifold. For any $x \in \tilde{M}$, there are smooth coordinates $\left(\tilde{q}^1, \ldots, \tilde{q}^d, \tilde{p}^1, \ldots, \tilde{p}^n\right)$ centered at x in which ω^2 has the coordinate representation*

$$\tilde{\omega}^2 = \sum_{i=1}^{n} \mathrm{d}\tilde{q}^i \wedge \mathrm{d}\tilde{p}^i. \tag{44}$$

as a Poincaré 2*-form, any coordinates satisfying* (44) *are called Darboux or symplectic coordinates.*

Remark 5. *Therefore, for any symplectic form w^2 on the above cotangent bundle T^*M of M, we can always find a symplectic coordinate with the* Poincaré 2-form*.*

Corollary 4 ((Da Silva & Da Salva, 2008))**.** *According to* (31)*, from the antisymmetric of the* 2*-forms, we have*

$$\omega^2(X_H, X_H) = 0 \tag{45}$$

which implies that hamiltonian vector fields preserve their hamiltonian functions H.

Remark 6. *In physics, hamiltonian functions are typically energy functions for a physical system. Corollary 4 indicates if the system updates its status according to the hamiltonian vector fields, the time-evolution of the system follows the law of conservation of energy.*

Definition 23 (Hamiltonian orbit generated from Poincaré 2-form (Lee, 2013))**.** *The Hamiltonian orbit generated by the Hamiltonian flow X_H on the cotangent bundle equipped with the Poincaré* 2*-form given in local coordinates (q, p) is given by*

$$\dot{q}^i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q^i}. \tag{46}$$

Definition 24 (cogeodesic orbits)**.** *If additionally M is equipped with a metric tensor g, i.e, if M is a pseudo-Riemannian manifold with metric g, and if we set the hamiltonian H on T^*M as*

$$H(q, p) = \frac{1}{2} g^{ij}(q) p_i p_j,$$

the Hamiltonian orbit generated from Poincaré 2-form is given by

$$\dot{q}^i = \frac{\partial H}{\partial p_i} = g^{ij} p_j, \quad \dot{p}_i = -\frac{\partial H}{\partial q^i} = -\frac{1}{2} \partial_i g^{jk} p_j p_k. \tag{47}$$

It is called the ***cogeodesic orbits of*** *(M, g). The canonical projection of cogeodesic orbits under π* (28) *is called* geodesic *on the base manifold M which generalizes the notion of a "straight or shortest line" to manifold where the length is measured by the metric tensor.*

B Some Formulations and More Hamiltonian Flows

In section, we first present the formulations which are not shown in detail in the main paper due to space constraints. More Hamiltonian-related flows are also presented, which however do not strictly follow the Hamiltonian flows on the cotangent bundle T^*M.

B.1 W IN SECTION 4.1.1

$$W = \begin{pmatrix} 0 & \partial_1 f_{2,\mathrm{net}} - \partial_2 f_{1,\mathrm{net}} & \partial_1 f_{3,\mathrm{net}} - \partial_3 f_{1,\mathrm{net}} & \cdots \\ \partial_2 f_{1,\mathrm{net}} - \partial_1 f_{2,\mathrm{net}} & 0 & \partial_2 f_{3,\mathrm{net}} - \partial_3 f_{2,\mathrm{net}} & \cdots \\ \partial_3 f_{1,\mathrm{net}} - \partial_1 f_{3,\mathrm{net}} & \partial_3 f_{2,\mathrm{net}} - \partial_2 f_{3,\mathrm{net}} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{48}$$

B.2 POINCARÉ 2-FORM $\omega^2_{\mathrm{Poincssaré}}$ AND LEARNABLE METRIC g_{net} WITH RELAXATION

Similar to Section 4.1.4, we now impose additional system biases along the curve compared to the cogeodesic orbits Section 4.1.5,

$$\dot{q}^i = g^{ij}_{\mathrm{net}} p_j, \quad \dot{p}_i = -\frac{1}{2}\partial_i g^{jk}_{\mathrm{net}} p_j p_k + f_{\mathrm{net}}(q). \tag{49}$$

Therefore, the projection of the curve from (49) now no longer follows the geodesic curve along the base manifold equipped with metric g_{net}.

B.3 HAMILTONIAN RELAXATION FLOW WITH HIGHER DIMENSIONAL "MOMENTUM"

In the paper main context, we present a new type of Hamiltonian-related flow, which does not strictly follow the Hamiltonian equations. Inspired from the work (Haber & Ruthotto, 2017), we now associate to each node $q \in \mathbb{R}^d$ an additional a *learnable* momentum vector $p \in \mathbb{R}^k$ which however is not strictly a cotangent vector of the manifold if $d \neq k$. We update the node features using the following equations

$$\begin{aligned} \dot{q} &= \phi\left(h^1_{\mathrm{net}}(p) - \rho q\right), \\ \dot{p} &= \phi\left(h^2_{\mathrm{net}}(q) - \rho p\right). \end{aligned} \tag{50}$$

where h^1_{net} and h^2_{net} are neural networks with d-dimensional output and k-dimensional output respectively, ϕ is a non-linear activation function and ρ is a scalar hyper-parameter.

C EXPERIMENTS SETTING

We select the citation networks Cora(McCallum et al., 2004), Citeseer(Sen et al., 2008), and Pubmed(Namata et al., 2012), and the low-hyperbolicity (Chami et al., 2019) Disease, Airport as the benchmark datasets. The citation dataset is widely used in graph representation learning tasks. We randomly generate 20 samples per class for training, 500 samples for validation, and 1000 samples for testing in each dataset as the same setting in Kipf & Welling (2017a). The low-hyperbolicity datasets Disease and Airport are proposed in Chami et al. (2019), where the Euclidean GNN models cannot learn the node embeddings effectively. We follow the same data splitting and pre-processing in Chami et al. (2019) for Disease and Airport datasets.

We adjust the model parameters in HamGNN based on the results from the validation data. We use the ADAM optimizer (Kingma & Ba, 2014) with the weight decay as 0.001. We set the learning rate as 0.01 for citation networks and 0.001 for Disease and Airport datasets. The results presented in Table 1 are under the 3 layers HamGNN setting. We report the results by running the experiments over 5 times with different initial random seeds. The q in Hamiltonian flow is initialized by the node embeddings in the graph. The p is implemented by the linear transform from q, or we can initialize p as a learnable parameter. Table 5 shows the implementation details of H_{net} in HamGNN.

We employ the ODE solver (Chen, 2018) in the implementation of HamGNN. For computation efficiency and performance effectiveness, the fixed-step explicit Euler solver (Chen et al., 2018a) is used in HamGNN. We also compare the influence of ODE solvers and report the results in Table 3.

ODE solver	Euler	Euler	Implicit Adams	Implicit Adams	Dopri5
Step size	0.1	0.5	0.1	0.5	-
Cora	81.10±1.13	81.52±1.27	81.62±0.58	81.40±0.77	81.62±0.58

Table 3: Node classification accuracy(%) under different ODE solvers in HamGNN (15).

Dataset	Models	3 layers	5 layers	10 layers	20 layers
Cora	GCN	80.29±2.29	69.87±1.12	26.50±4.68	23.97±5.42
	HGCN	78.70±0.96	38.13±6.20	31.90±0.00	26.23±9.87
	HamGNN (15)	81.52±1.27	**81.58±0.73**	79.00±2.17	76.20±0.13
	HamGNN (17)	81.84±0.88	81.08±0.16	**81.40±0.44**	**80.58±0.30**
	HamGNN (17) type 2	**82.10±0.80**	81.10±0.10	81.06±1.49	79.26±1.07
Pubmed	GCN	77.83±0.77	76.00±0.87	77.53±1.06	56.50±12.79
	HGCN	76.38±0.81	77.20±1.05	65.90±9.44	42.16±2.54
	HamGNN (15)	78.18±0.54	77.86±1.45	77.73±1.15	76.13±0.80
	HamGNN (17)	78.83±0.46	78.43±0.25	78.50±0.61	77.20±0.69
	HamGNN (17) type 2	**79.03±0.58**	**78.46±0.11**	**78.53±0.31**	**77.50±0.44**

Table 4: Node classification accuracy(%) when increasing the number of layers on the Pubmed dataset.

D More Ablation Studies and Experiments

D.1 Vanilla ODE

To demonstrate the advantage of HamGNN's design, we also conduct more experiments that replace the Hamiltonian layer in HamGNN with a vanilla ODE as follows:

$$\dot{q}(t) = \tilde{f}_{\text{net}}(q(t)) \tag{51}$$

where the $\tilde{f}_{\text{net}}(q(t))$ is composed of two FC layers and a non-linear activation function. Compare to the Hamiltonian flows in Section 4.1, the equation (51) does not include the learnable "momentum" vector for each node and does not follow the Hamiltonian flows on the cotangent bundle T^*M.

D.2 Over-Smoothing

We continue from Section 5.2 to conduct more experiments on the Cora and Pubmed datasets to demonstrate the resilience of HamGNN against over-smoothing. Moreover, to further show that with the convex H_{net} in (17) HamGNN can perform better than the vanilla H_{net} in (15) against over-smoothing, we now include more choices of convex network H_{net} with different layer sizes and different activation functions (as along as the layer weights from the second layer in H_{net} are non-negative, and all activation functions in H_{net} are convex and non-decreasing). The network details of H_{net} are given in Table 5 (see Method (17)). The experiment results are shown in Table 4.

We clearly observe that for different convex functions and on different datasets, HamGNN with convex H_{net} nearly keeps the full node classification ability even we have stacked 20 Hamiltonian layers.

Fig. 3 shows how the node features evolve over 10 layers. For a detailed description of the figure setting, please refer to Fig. 4. We can observe from Fig. 3 that HamGNN has its learned node features change steadily and slowly, while the node features learned by GIL and GCN change abruptly, especially for the feature phases. The features from two nodes of different classes, learned by GIL and GCN, are converging to each other much faster than HamGNN.

D.3 Stability

It is well known that a Hamiltonian system is an energy-conservative system. Our HamGNN inherits this conservative or stability property. To show that, we sample a node (from the test set) in the

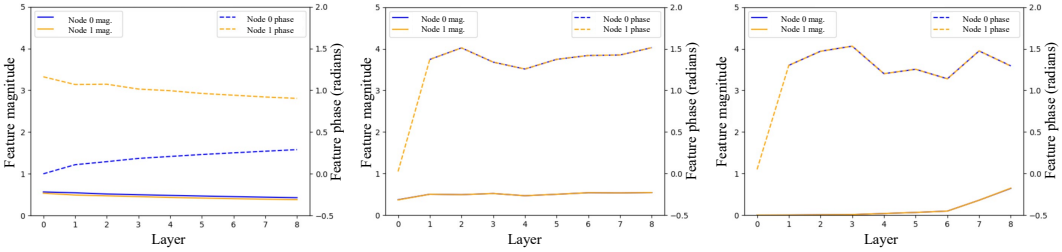

Figure 3: Two nodes from different classes and the evolution of their feature vectors over layers. Left: HamGNN, middle: GIL, right: GCN.

PubMed graph data and pass it to three neural ODE-based GNN models, which are 1) HamGNN, 2) one ODE using a positive-definite linear layer as the ODE function, and 3) one ODE using a negative-definite linear layer as the ODE function. These GNNs all contain 10 layers and we compute the node feature magnitude, i.e., $L2$-norm of the feature vector, and the node feature phase, which is the cosine similarity between the output feature at the current layer and the input feature to the first layer. The results are shown in Fig. 4. The trends are very obvious, where the feature magnitude (you may think of it as feature energy) learned by HamGNN is conserved over layers, while the feature magnitude learned by the positive-definite explodes after a few layers, and the feature magnitude learned by the negative-definite ODE is very close to zero. Regarding the feature phase, the feature phases from two nodes are steadily and slowly approaching each other when using HamGNN, while the feature phases learned by the other neural ODEs change abruptly and the difference between two nodes using negative-definite ODE is negligibly small.

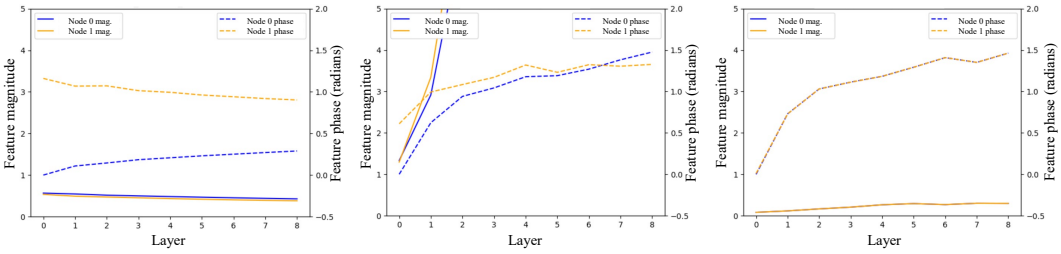

Figure 4: Two nodes from different classes and the evolution of their feature vectors over layers. Left: HamGNN, middle: neural ODE using a positive-definite linear layer, right: neural ODE using a negative-definite linear layer.

Table 5: Neural network modules and the parameters, where $\kappa(x) = 0.5x^2$ if $x > 0$ and $= \exp(x) - 1$ if $x < 0$

Methods	Modules	Activation	Output Channels
(15)	Linear tanh Linear	tanh	1
(17)	ReHU(Amos et al., 2017) Linear ReHU Linear ReHU Linear ReHU	ReHU	1
(17) type 2	κ Linear κ Linear κ Linear κ	ReHU	1
(50)	Linear act1 Linear act1	act1	576
(18)	Linear tanh Linear	tanh	1
(49)	Linear tanh Linear Sigmoid	tanh Sigmoid	64
(14)	Linear sin Linear sin	sin	H:1 W:128