

Isometric Manifold Learning Using Hierarchical Flow

Ziqi Pan, Jianfu Zhang*, Li Niu*, Liqing Zhang

MoE Key Lab of Artificial Intelligence, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{panziqi_ai, c.sis, ustcnewly}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract

We propose the Hierarchical Flow (HF) model constrained by isometric regularizations for manifold learning that combines manifold learning goals such as *dimensionality reduction*, *inference*, *sampling*, *projection* and *density estimation* into one unified framework. Our proposed HF model is regularized to not only produce embeddings preserving the geometric structure of the manifold, but also project samples onto the manifold in a manner conforming to the rigorous definition of projection. Theoretical guarantees are provided for our HF model to satisfy the two desired properties. In order to detect the real dimensionality of the manifold, we also propose a two-stage dimensionality reduction algorithm, which is a time-efficient algorithm thanks to the hierarchical architecture design of our HF model. Experimental results justify our theoretical analysis, demonstrate the superiority of our dimensionality reduction algorithm in cost of training time, and verify the effect of the aforementioned properties in improving performances on downstream tasks such as *anomaly detection*.

1 Introduction

Manifold learning (Pless and Souvenir 2009) is the task aiming to recover low-dimensional embeddings from nonlinear high-dimensional data that preserve the geometric structure of data manifolds (Fefferman, Mitter, and Narayanan 2016), which is a fundamental research topic in the field of machine learning. Applications of manifold learning include *nonlinear dimensionality reduction* (Gisbrecht and Hammer 2015), *denoising* (Buades, Coll, and Morel 2005), *anomaly detection* (Chandola, Banerjee, and Kumar 2009), *etc.*

The research on manifold learning has a long history, and the manifold learning methods range from classical algorithms such as *LLE* (Rowes 2000), *Local Preserving Projections* (He and Niyogi 2003), *Semidefinite Embedding* (Weinberger and Saul 2006), and *Isomap* (Tenenbaum, Silva, and Langford 2000), to modern deep learning methods based on neural network frameworks such as *Generative Adversarial Nets* (Goodfellow et al. 2014) (GANs), *Variational Autoencoders* (Kingma and Welling 2013) (VAEs), and *Normalizing Flow* (Dinh, Krueger, and Bengio 2014) (NF) based generative models. Classical spectral methods (*e.g.*, *Isomap*

and *LLE*) are commonly non-parametric methods by utilizing the *Multidimensional Scaling* (Carroll and Arabie 1998) (MDS) algorithm and eigendecomposition. However, classical manifold learning algorithms are hard to apply to modern datasets, since these datasets are typically complex and high-dimensional (Verleysen and François 2005). Since non-parametric methods do not produce an embedding function, the inference of the embedding can not be generalized to unseen samples. To tackle this problem, some parametric methods are proposed such as *SNE* (Van Der Maaten 2009), *t-SNE* (Van der Maaten and Hinton 2008), *DrLIM* (Hadsell, Chopra, and LeCun 2006) and so on (Gong et al. 2006; Chui and Mhaskar 2018; Mishne et al. 2019), which can perform out-of-sample-extension. There are also methods combining classical algorithms and neural networks for manifold learning (Pai et al. 2019; Geng et al. 2020). In terms of the neural network methods (Basri and Jacobs 2017), GANs and VAEs as generative models can also be regarded as manifold learning framework, which is able to efficiently sample from the manifold by using *ancestral sampling*. However, GANs and VAEs lack the ability to estimate densities on the manifold. Compared to GANs and VAEs, since NFs induce an explicit density on the manifold based on the *change-of-variable formula* and are intrinsic nonlinear bijective mappings between the embeddings and the manifold, many NF-based manifold learning models are proposed, including NFs on a prescribed manifold (Gemicic, Rezende, and Mohamed 2016; Bose et al. 2020) such as tori and spheres (Rezende et al. 2020), and the *M-flow* (Brehmer and Cranmer 2020) for learning a manifold topologically equivalent to Euclidean space. In this paper, we propose a *Hierarchical Flow* (HF) model by extending the *M-flow* into a hierarchical architecture constrained by isometric regularizations for manifold learning.

Given a set of training samples from a manifold $\mathcal{M} \subset \mathbb{R}^D$ whose ground-truth dimensionality K^* is unknown, our proposed HF model combines the following manifold learning goals in one unified framework:

- *Dimensionality reduction*. The model is able to detect the ground-truth dimensionality K^* of \mathcal{M} efficiently.
- *Sampling*. The model can efficiently sample from \mathcal{M} .
- *Inference*: Given a manifold sample $x \in \mathcal{M}$, the model can infer the embedding $u(x) \in \mathbb{R}^{K^*}$ of x that is faithful to the geometric structure of \mathcal{M} . Specifically, $\forall x_1, x_2 \in$

*Corresponding authors

\mathcal{M} , the Euclidean distance between $u(x_1)$ and $u(x_2)$ is equal to the manifold distance (*i.e.*, the geodesic length) between x_1 and x_2 (Petersen 2006), which we refer to as the *distance preserving* property.

- *Projection.* Given an off-manifold sample $x \notin \mathcal{M}$ from the ambient space \mathbb{R}^D close to \mathcal{M} , the model can project x onto \mathcal{M} in a manner that conforms to the rigorous definition of projection, namely the projection $\tilde{x} \in \mathcal{M}$ of x satisfies that $\tilde{x} = \operatorname{argmin}_{x' \in \mathcal{M}} d(x', x)$, and the distance from x to \mathcal{M} is $d(\tilde{x}, x)$, where $d: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is the Euclidean distance in \mathbb{R}^D . We refer to such a property of the model as the *rigorous projection* property.
- *Density estimation.* Given $x \in \mathcal{M}$, the model is capable of estimating the explicit density at x .

Existing manifold learning methods differ in their ability to achieve the above goals, see Tbl. 1 for comparison. In terms of *projection*, although existing methods such as \mathcal{M} -flow are able to project given samples onto the manifold, the projections are not guaranteed to conform to the rigorous definition as we mentioned above, which could affect the performance on downstream tasks such as *anomaly detection*, as we show in Sec. 4 and supplementary. For our HF model, we provide theoretical guarantees on the *rigorous projection* property of the generator constrained by our proposed isometric regularizations, and show through experimental results that our HF model performs better on downstream tasks such as *anomaly detection* due to the *rigorous projection* property. Moreover, thanks to the hierarchical architecture of our HF model, we can develop a *two-stage dimensionality reduction* algorithm to detect the ground-truth dimensionality of the manifold in a time-efficient manner. Compared with a brute-force search algorithm, our proposed algorithm is greatly superior in cost of training time. Our contributions are threefold:

- We propose the *Hierarchical Flow* (HF) model for manifold learning, which is constrained by isometric regularizations for satisfying the properties of *distance preserving* and *rigorous projection* with theoretical guarantees. Our proposed HF model combines the manifold learning goals mentioned above in one unified framework.
- We propose a two-stage dimensionality reduction algorithm based on the hierarchical architecture design of our HF model, which allows us to detect the ground-truth dimensionality of the manifold in a time-efficient manner.
- Experimental results justify our theoretical analysis, and show the time-efficiency of our dimensionality reduction algorithm and the superiority of our proposed HF model in performance on downstream tasks such as *anomaly detection*, thanks to the aforementioned desired properties.

The remainder of this paper is organized as follows. Sec. 2 provides details of our manifold learning method, including our HF model, the properties of *distance preserving* and *rigorous projection* with theoretical guarantees, and the training objectives and algorithms for manifold learning and dimensionality reduction. Sec. 3 relates our method to prior work. We provide experimental results in Sec. 4 to justify our theoretical analysis, and conclude our paper in Sec. 5.

2 Methodology

In this section, we introduce the details of our proposed HF model and its training objectives and algorithms.

2.1 Hierarchical Generator

Given $K \leq D$, we propose to employ a generator $g: \mathbb{R}^K \rightarrow \mathbb{R}^D$ with a hierarchical architecture to characterize the data manifold \mathcal{M} using K -dimensional embedding. Specifically, $g: \mathbb{R}^K \rightarrow \mathbb{R}^D$ comprised of L layers can be represented as

$$g = \underbrace{f_L \circ p_L}_{m_L} \circ \underbrace{f_{L-1} \circ p_{L-1}}_{m_{L-1}} \circ \cdots \circ \underbrace{f_1 \circ p_1}_{m_1}, \quad (1)$$

where $m_i = f_i \circ p_i: \mathbb{R}^{K_{i-1}} \rightarrow \mathbb{R}^{K_i}$ is the i -th layer of g , which is comprised of a *flow* module $f_i: \mathbb{R}^{K_i} \rightarrow \mathbb{R}^{K_i}$ and a *padding* module $p_i: \mathbb{R}^{K_{i-1}} \rightarrow \mathbb{R}^{K_i}$, where $K = K_0 \leq K_1 \leq \cdots \leq K_L = D$. The flow module $f_i: \mathbb{R}^{K_i} \rightarrow \mathbb{R}^{K_i}$ is a nonlinear bijective mapping with an explicit inversion f_i^{-1} and a tractable Jacobian determinant, which is comprised of several *coupling* layers (Dinh, Krueger, and Bengio 2014) and 1×1 convolution layers. In terms of the padding module $p_i: \mathbb{R}^{K_{i-1}} \rightarrow \mathbb{R}^{K_i}$, given an input feature $x \in \mathbb{R}^{K_{i-1}}$, p_i pads $K_i - K_{i-1}$ zeros at the end of x

$$p_i(x) = [x; 0] \in \mathbb{R}^{K_i}, \quad (2)$$

and the pseudo inverse $p_i^\dagger: \mathbb{R}^{K_i} \rightarrow \mathbb{R}^{K_{i-1}}$ is also referred to as a *projection* module that drops the last $K_i - K_{i-1}$ elements of an input feature x (Brehmer and Cranmer 2020)

$$p_i^\dagger(x) = [x_1, x_2, \cdots, x_{K_{i-1}}] \in \mathbb{R}^{K_{i-1}}, \quad (3)$$

where x_j is the j -th element of x . Hence the pseudo inverse of g , *i.e.*, $g^\dagger: \mathbb{R}^D \rightarrow \mathbb{R}^K$, is given as

$$g^\dagger = \underbrace{p_1^\dagger \circ f_1^{-1}}_{m_1^\dagger} \circ \cdots \circ \underbrace{p_{L-1}^\dagger \circ f_{L-1}^{-1}}_{m_{L-1}^\dagger} \circ \underbrace{p_L^\dagger \circ f_L^{-1}}_{m_L^\dagger}, \quad (4)$$

which is essentially an encoder that can be used to infer the embedding $u(x) \in \mathbb{R}^K$ of a sample $x \in \mathbb{R}^D$. For the details of the implementation of g , see supplementary.

We firstly introduce mathematical notations used throughout the paper, and then provide the theoretical guarantees for satisfying the properties of *distance preserving* and *rigorous projection* for the generator g . Given $x \in \mathbb{R}^D$, we denote

$$[u^i(x); v^i(x)] \triangleq f_i^{-1}(u^{i+1}(x)), \quad i \in [L], \quad (5)$$

where $u^i(x) \in \mathbb{R}^{K_{i-1}}$ and $v^i(x) \in \mathbb{R}^{K_i - K_{i-1}}$, and therefore $p_i^\dagger([u^i(x); v^i(x)]) = u^i(x)$. We use $u^{L+1}(x) \triangleq x$, and denote the embedding $u(x)$ of x as $u(x) \triangleq u^1(x)$. We also use $[N] \triangleq \{1, 2, \cdots, N\}$ for a positive integer N . From the above, running inference for $x \in \mathbb{R}^D$ using g^\dagger produces $\{u^i(x)\}_{i=1}^L$ and $\{v^i(x)\}_{i=1}^L$. Given a manifold $\mathcal{M} \subset \mathbb{R}^D$, for $i \in [L]$ and some $r > 0$, we denote

$$\mathcal{U}^i(\mathcal{M}) \triangleq \{u^i(x) \mid x \in \mathcal{M}\}, \quad (6)$$

$$\mathcal{V}^i(r) \triangleq \{v \in \mathbb{R}^{K_i - K_{i-1}} \mid \|v\|_2 \leq r\}, \quad (7)$$

and denote

$$\mathcal{A}^i(\mathcal{M}; r) \triangleq \{f_i([u; v]) \mid u \in \mathcal{A}^{i-1}(\mathcal{M}; r), v \in \mathcal{V}^i(r)\}, \quad (8)$$

$$\tilde{\mathcal{A}}^i(\mathcal{M}; r) \triangleq \{f_i([u; 0]) \mid u \in \mathcal{A}^{i-1}(\mathcal{M}; r)\}, \quad (9)$$

where we use $\mathcal{A}^0(\mathcal{M}; r) \triangleq \mathcal{U}^1(\mathcal{M})$. For brevity of description, we also stipulate that $f_i(u) \triangleq f_i([u; 0]) = f_i \circ p_i(u)$ if the given u is a K_{i-1} dimensional vector. We also present an intuitive understanding for $\mathcal{U}^i(\mathcal{M})$, $\mathcal{V}^i(r)$, $\mathcal{A}^i(\mathcal{M}; r)$ and $\tilde{\mathcal{A}}^i(\mathcal{M}; r)$ by using Fig. 2.

The Distance Preserving Property We formally state the conditions for the generator g to satisfy the *distance preserving* property in the following Prop. 1 (see supplementary for proof) with theoretical guarantees, where an orthonormal Jacobian is encouraged for the generator g . Recent works also show that an isometric autoencoder preserves the geometric structure of the data manifold (Gropp, Atzmon, and Lipman 2020; Yonghyeon et al. 2021). We present an intuitive illustration for the *distance preserving* property in Fig. 1.

Proposition 1 (*Distance preserving property*). Assume that $\mathcal{U}^1(\mathcal{M})$ is a convex set, and $v^i(x) = 0$ for $\forall i \in [L], x \in \mathcal{M}$. The length of the geodesic between $g(u_1)$ and $g(u_2)$ on \mathcal{M} for $\forall u_1, u_2 \in \mathcal{U}^1(\mathcal{M})$ equals to $\|u_1 - u_2\|_2$, if $J_g(u)$ is orthonormal for $\forall u \in \mathcal{U}^1(\mathcal{M})$, where $J_g(u)$ is the Jacobian matrix of the generator g at u .

The Rigorous Projection Property We also formally state the conditions for each flow module f_i to satisfy the *rigorous projection* property in the following Prop. 2 (see supplementary for proof) with theoretical guarantees. To intuitively see how satisfying the conditions in Prop. 2 leads to a flow module f_i that satisfies the *rigorous projection* property, refer to supplementary. In terms of the *rigorous projection* property of the whole generator g , we directly learn from Prop. 2 that a generator g comprised of only one layer (i.e., $g = f_1 \circ p_1$) satisfies the *rigorous projection* property, which is formally stated in Cor. 1. For g comprised of more than one layer, the *rigorous projection* property of g is approximately satisfied over an ambient space close to \mathcal{M} , see Rem. 1 and Fig. 2.

Proposition 2 (*Rigorous projection property*). Assume that $v^i(x) = 0$ for $\forall x \in \mathcal{M}$. Given $x \in \mathcal{A}^i(\mathcal{M}; r)$, by denoting $[\alpha; \beta] \triangleq f_i^{-1}(x)$ where $\alpha \in \mathbb{R}^{K_{i-1}}$ and $\beta \in \mathbb{R}^{K_i - K_{i-1}}$, the projection of x onto $\tilde{\mathcal{A}}^i(\mathcal{M}; r)$ is $\tilde{x} \triangleq f_i(\alpha)$, and the distance from x to $\tilde{\mathcal{A}}^i(\mathcal{M}; r)$, $d_{\mathcal{E}}(x, \tilde{x})$, equals to $\|\beta\|_2$, if $\forall u \in \mathcal{A}^{i-1}(\mathcal{M}; r), v \in \mathcal{V}^i(r)$,

$$I([u; v]) \triangleq J([u; v])^T J([u; v]) \in \mathbb{R}^{K_i \times K_i} \quad (10)$$

is a diagonal matrix, and $I_{jj}([u; v])$ equals to 1 for $K_{i-1} + 1 \leq j \leq K_i$, where $J([u; v]) \in \mathbb{R}^{K_i \times K_i}$ is the Jacobian of f_i at $[u; v]$, I_{jj} is the (j, j) -th element of I , and $d_{\mathcal{E}}(\cdot, \cdot)$ is the Euclidean distance in \mathbb{R}^{K_i} .

Corollary 1. Given $g = f_1 \circ p_1$, assume that $v^1(x) = 0$ for $\forall x \in \mathcal{M}$. Given $x \in \mathcal{A}^1(\mathcal{M}; r)$, the projection of x onto \mathcal{M} is $\tilde{x} \triangleq g(u^1(x)) = g \circ g^\dagger(x)$, and the distance from x to \mathcal{M} , $d_{\mathcal{E}}(x, \tilde{x})$, is $\|v^1(x)\|_2$, if f_1 satisfies the conditions for the *rigorous projection* property in Prop. 2.

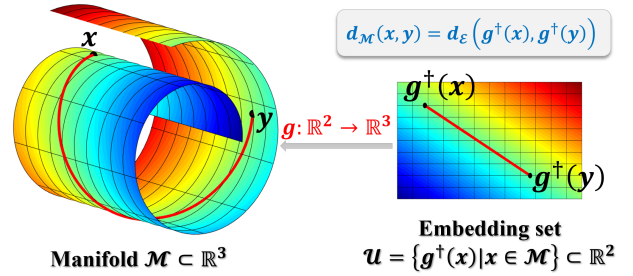


Figure 1: An intuitive illustration for the *distance preserving* property by using the exemplar manifold ‘‘Swiss roll’’ (also see Fig.3 of (Tenenbaum, Silva, and Langford 2000)). \mathcal{U} is obtained by a generator g whose Jacobian J_g is orthonormal. The correspondence between \mathcal{M} and \mathcal{U} is indicated by color, and the orthonormality of J_g can be shown by the coordinate lines of \mathcal{M} . The geodesic between x and y is plotted by using the red line on \mathcal{M} , and the geodesic length $d_{\mathcal{M}}(x, y)$ is equal to the Euclidean distance $d_{\mathcal{E}}(g^\dagger(x), g^\dagger(y))$ between the corresponding embeddings $g^\dagger(x)$ and $g^\dagger(y)$.

Remark 1. Given g comprised of $L \geq 2$ layers, assume that $v^i(x) = 0$ for $\forall x \in \mathcal{M}, i \in [L]$. Given $x \in \mathcal{A}^L(\mathcal{M}; r)$, let $\tilde{x}^i \triangleq f_i(u^i(x)), i \in [L]$, and $\tilde{x} \triangleq g(u^1(x)) = g \circ g^\dagger(x)$. Assume that r is small such that $\mathcal{A}^L(\mathcal{M}; r)$ is an ambient space close to \mathcal{M} . Given f_i that satisfies the *rigorous projection* property for $i \in [L]$, we have

- The projection of $u^{i+1}(x)$ onto $\tilde{\mathcal{A}}^i(\mathcal{M}; r)$ is \tilde{x}^i , and the distance from $u^{i+1}(x)$ to $\tilde{\mathcal{A}}^i(\mathcal{M}; r)$, $d_{\mathcal{E}}(u^{i+1}(x), \tilde{x}^i)$, equals to $\|v^i(x)\|_2$, where $i \in [L]$.
- The projection of x onto \mathcal{M} is \tilde{x} , and the distance from x to \mathcal{M} , $d_{\mathcal{E}}(x, \tilde{x})$, is approximately $\sqrt{\sum_{i=1}^L \|v^i(x)\|_2^2}$.

See Fig. 2 and supplementary for detailed analysis.

2.2 Isometric Regularizations

Based on the aforementioned Prop. 1-2, we propose two isometric regularizations for the generator g to satisfy the *distance preserving* property and the *rigorous projection* property, respectively. Both isometric regularizations are realized to pursue an orthonormal Jacobian.

Orthonormal Regularization on $\mathcal{A}^{i-1}(\mathcal{M}; r)$ To satisfy the *distance preserving* property for g , according to Prop. 1 and the following Prop. 3 (see supplementary for proof), we can constrain the Jacobian J_{f_i} of f_i to be orthonormal over $\mathcal{A}^{i-1}(\mathcal{M}; r)$ for $i \in [L]$, namely

$$J_{f_i}(u) \text{ is orthonormal, } \forall u \in \mathcal{A}^{i-1}(\mathcal{M}; r). \quad (11)$$

In practice, to ensure the generator expressiveness, we propose to constrain J_{f_i} to be orthonormal up to a global scalar $\zeta > 0$. To achieve this, we propose to utilize a sample efficient regularization on J_{f_i} based on the recently proposed *Linearized Transpose* (Pan, Niu, and Zhang 2022) (LT) technique that is used to efficiently estimate the spectral norm of the Jacobian. Specifically, we exploit the property of f_i that

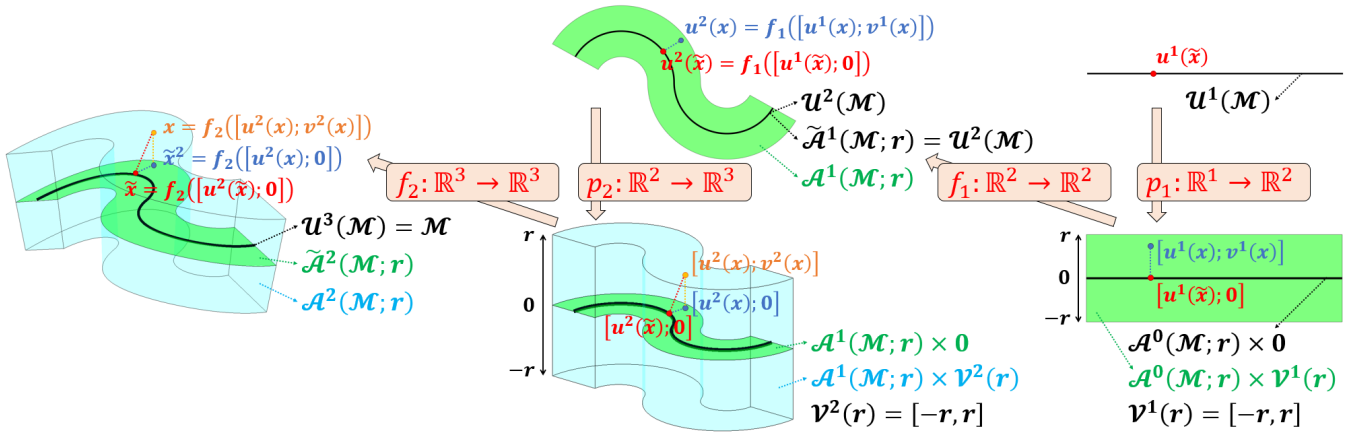


Figure 2: An intuitive illustration of the math notations and the rigorous projection for a hierarchical generator $g = f_2 \circ p_2 \circ f_1 \circ p_1 : \mathbb{R}^1 \rightarrow \mathbb{R}^3$. The projection of $x \in \mathcal{A}^2(\mathcal{M}; r)$ onto $\tilde{\mathcal{A}}^2(\mathcal{M}; r)$ and \mathcal{M} are $\tilde{x}^2 \triangleq f_2 \circ p_2 \circ p_2^\dagger \circ f_2^{-1}(x) = f_2([u^2(x); 0])$ and $\tilde{x} \triangleq g \circ g^\dagger(x)$, respectively. Assume that \tilde{x} is in a local region around \tilde{x}^2 , i.e., it can be regarded that $\tilde{x} \in T_{\tilde{x}^2} \tilde{\mathcal{A}}^2(\mathcal{M}; r)$ and $d_{\tilde{\mathcal{A}}^2(\mathcal{M}; r)}(\tilde{x}, \tilde{x}^2) \approx d_{\mathcal{E}}(\tilde{x}, \tilde{x}^2)$, where $T_{\tilde{x}^2} \tilde{\mathcal{A}}^2(\mathcal{M}; r)$ represents the tangent space of $\tilde{\mathcal{A}}^2(\mathcal{M}; r)$ at \tilde{x}^2 . Because f_2 satisfies the *rigorous projection* property, from Prop. 2 and the rigorous definition of projection, we have $d_{\mathcal{E}}(x, \tilde{x}^2) = \|v^2(x)\|_2$ and $\text{vec}(x, \tilde{x}^2) \perp T_{\tilde{x}^2} \tilde{\mathcal{A}}^2(\mathcal{M}; r)$, where $\text{vec}(x, \tilde{x}^2)$ denotes the vector from x to \tilde{x}^2 . Hence according to the Pythagorean theorem, we have $d_{\mathcal{E}}^2(x, \tilde{x}) = d_{\mathcal{E}}^2(x, \tilde{x}^2) + d_{\mathcal{E}}^2(\tilde{x}^2, \tilde{x}) \approx \|v^2(x)\|_2^2 + d_{\tilde{\mathcal{A}}^2(\mathcal{M}; r)}^2(\tilde{x}^2, \tilde{x})$. Because J_{f_2} is constrained to be orthonormal over $\mathcal{A}^1(\mathcal{M}; r)$, we have $d_{\tilde{\mathcal{A}}^2(\mathcal{M}; r)}(\tilde{x}, \tilde{x}^2) = d_{\mathcal{E}}(u^2(\tilde{x}), u^2(x))$. Given that f_1 satisfies the *rigorous projection* property, we know that $u^2(\tilde{x})$ is the projection of $u^2(x) \in \mathcal{A}^1(\mathcal{M}; r)$ onto $\tilde{\mathcal{A}}^1(\mathcal{M}; r)$ according to Prop. 2, since we know $u^1(x) = u^1(\tilde{x})$ from $\tilde{x} = g \circ g^\dagger(x)$. Hence $d_{\mathcal{E}}(u^2(\tilde{x}), u^2(x)) = \|v^1(x)\|_2$, and $d_{\mathcal{E}}(x, \tilde{x}) \approx \sqrt{\|v^1(x)\|_2^2 + \|v^2(x)\|_2^2}$.

$\det J_{f_i}$ is tractable. Because $\det J_{f_i}$ equals to the product of all singular values of J_{f_i} , we know that all singular values of J_{f_i} are equal and hence J_{f_i} is orthonormal up to a scalar, if we constrain the maximum singular value (i.e., the spectral norm) of J_{f_i} to equal $\sqrt{\det J_{f_i}}$. Therefore, in practice, we estimate the spectral norm of J_{f_i} by employing LT, and then constrain $\text{LT}_{f_i}(u) = \sqrt{\det J_{f_i}(u)} = \zeta$, where $\text{LT}_{f_i}(u)$ is the estimated spectral norm of J_{f_i} at $\forall u \in \mathcal{A}^{i-1}(\mathcal{M}; r)$.

Proposition 3. $J_g(z)$ is orthonormal for $\forall z \in \mathcal{U}^1(\mathcal{M})$, if $J_{f_i}(u)$ is orthonormal for $\forall u \in \mathcal{A}^{i-1}(\mathcal{M}; r), \forall i \in [L]$.

Proxy Regularization on $\mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$ Based on Prop. 2 and Rem. 1, to satisfy the *rigorous projection* property for g , an approach is to trivially constrain J_{f_i} to be orthonormal over $\mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$ by using LT. Although we can verify that this is a special case satisfying the conditions in Prop. 2, the flexibility of f_i is greatly reduced (see supplementary for detailed discussion). To tackle this problem, we observe that for a given $z \in \mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$, by using an auxiliary module $s_i : \mathbb{R}^{K_i} \rightarrow \mathbb{R}^{K_i}$, the Jacobian of the following composed mapping

$$\phi_i^z(\xi) \triangleq f_i(z - s_i(z) + s_i(z) \otimes \xi), \quad \xi \in \mathbb{R}^{K_i} \quad (12)$$

at $\xi = \mathbb{1}$, i.e., $J_{\phi_i^z}(\mathbb{1}) = \left. \frac{\partial \phi_i^z}{\partial \xi} \right|_{\xi=\mathbb{1}}$, equals to

$$J_{f_i}(z) \text{diag}(\omega_1, \omega_2, \dots, \omega_{K_i}), \quad (13)$$

where \otimes denotes the elementwise multiplication, $\mathbb{1} \in \mathbb{R}^{K_i}$ is a vector with all elements being 1, and ω_j denotes the j -th element of $s_i(z)$. From Prop. 4, we know that the following objective leads to an f_i satisfying the conditions for the *rigorous projection* in Prop. 2 (see supplementary for proof):

constrain all singular values of $J_{\phi_i^z}(\mathbb{1})$ to be equal,

$$\text{s.t. } \omega_j = \sqrt{\left(\prod_{l=1}^{K_{i-1}} \omega_l \right) \det J_{f_i}}, \quad K_{i-1} + 1 \leq j \leq K_i \quad (14)$$

In practice, we constrain $\text{LT}_{\phi_i^z}(\mathbb{1}) = \sqrt{\det J_{\phi_i^z}(\mathbb{1})}$ by using LT for $\forall z \in \mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$, which is a sample efficient regularization while giving flexibility to f_i . We refer to such a method as the proxy regularization for f_i to satisfy the *rigorous projection* property. We refer to the module s_i as the *singular values predictor* for f_i .

Proposition 4. All singular values of $J_{\phi_i^z}(\mathbb{1})$ equal a > 0 $\Leftrightarrow J_{f_i}(z)^\top J_{f_i}(z) = \text{diag}\left(\frac{a^2}{\omega_1^2}, \dots, \frac{a^2}{\omega_{K_i}^2}\right)$, and $\left\{\frac{a}{\omega_j}\right\}_{j=1}^{K_i}$ are the singular values of $J_{f_i}(z)$.

2.3 Training Objectives and Algorithms

We present the training objectives and algorithms for manifold learning given a training dataset $\mathcal{X} \triangleq \{x^{(i)} \in \mathcal{M}\}_{i=1}^N$

from a manifold $\mathcal{M} \subset \mathbb{R}^D$. We adopt the training strategy of \mathcal{M} -flow (Brehmer and Cranmer 2020) which separates the manifold and density training. We first introduce the training objectives for the manifold and density training respectively, then introduce the training algorithms, where a two-stage dimensionality reduction algorithm for detecting the ground-truth dimensionality K^* of \mathcal{M} is included.

Manifold Training Objectives During the manifold training phase, the following loss function \mathcal{L}_m is minimized

$$\mathcal{L}_m = \lambda_{\text{dist}} \mathcal{L}_{\text{dist}} + \lambda_{\text{proj}} \mathcal{L}_{\text{proj}} + \lambda_v \mathcal{L}_v, \quad (15)$$

where λ are balancing hyper-parameters. We then introduce each loss function in detail as follows.

$\mathcal{L}_{\text{dist}}$ is the isometric loss function used to satisfy the *distance preserving* property for the generator g , which is realized by constraining all the Jacobians of f_i to be orthonormal up to ζ as we introduced in Sec. 2.2. Specifically,

$$\mathcal{L}_{\text{dist}} = \frac{1}{L} \sum_{i=1}^L \mathbb{E}_{u^i \sim \mathcal{A}^{i-1}(\mathcal{M}; r)} \left\{ (\text{LT}_{f_i}(u^i) - \zeta)^2 + (\det J_{f_i}(u^i) - \zeta)^2 \right\}. \quad (16)$$

Note that $\mathcal{L}_{\text{dist}}$ involves sampling u^i from $\mathcal{A}^{i-1}(\mathcal{M}; r)$. For $i = 1$, since $\mathcal{A}^0(\mathcal{M}; r) = \mathcal{U}^1(\mathcal{M})$, we sample u^1 from the K -dimensional normal distribution $\mathcal{N}(\mu, \sigma^2)$, where u and σ^2 are the running mean and running variance of $u^1(x)$ for $x \in \mathcal{X}$. For $2 \leq i \leq L$, according to Eq. (8), we sample u^{i-1} from $\mathcal{A}^{i-2}(\mathcal{M}; r)$ and v^{i-1} from $\mathcal{V}^{i-1}(r)$, respectively, and then obtain $u^i = f_{i-1}([u^{i-1}; v^{i-1}])$.

$\mathcal{L}_{\text{proj}}$ is the isometric loss function used to satisfy the *rigorous projection* property for the generator g , which is realized by the proxy regularization over $\mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$ as we introduced in Sec. 2.2. Specifically,

$$\mathcal{L}_{\text{proj}} = \frac{1}{L} \sum_{i=1}^L \mathbb{E}_{z \sim \mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)} \left(\text{LT}_{\phi_i^z}(\mathbb{1}) - \det J_{f_i}(z) \prod_{j=1}^{K_i} \omega_j \right)^2, \quad (17)$$

where we sample z from $\mathcal{A}^{i-1}(\mathcal{M}; r) \times \mathcal{V}^i(r)$ by using the *ancestral sampling* method involved in $\mathcal{L}_{\text{dist}}$. Note that from Eq. (13) we learn that $\det J_{\phi_i^z}(\mathbb{1}) = \det J_{f_i}(z) \prod_{j=1}^{K_i} \omega_j$.

\mathcal{L}_v is the reconstruction loss function, which constrains g to satisfy $g \circ g^\dagger(x) = x$ for $x \in \mathcal{X}$. Instead of adopting the commonly used loss function as follows

$$\frac{1}{N} \sum_{i=1}^N \left\| g \circ g^\dagger(x^{(i)}) - x^{(i)} \right\|_2^2, \quad (18)$$

we can use the following loss function

$$\mathcal{L}_v = \frac{1}{N} \sum_{k=1}^N \sum_{i=1}^L \left\| v^i(x^{(k)}) \right\|_2^2. \quad (19)$$

Since \mathcal{L}_v depends on $\{v^i(x)\}_{i=1}^L$ to compute the loss for a given $x \in \mathcal{X}$, compared with Eq. (18), \mathcal{L}_v requires only the

encoding process (i.e., $g^\dagger(x)$) and no decoding process (i.e., $g(u(x))$ where $u(x)$ is the embedding of x), which reduces the computational cost and hence speed up training. In order to see how \mathcal{L}_v is related to a reconstruction loss function for \mathcal{X} , according to Rem. 1 we learn that the distance from x to the generated manifold $\widetilde{\mathcal{M}}$ of g is about $\sqrt{\sum_{i=1}^L \|v^i(x)\|_2^2}$.

Therefore, minimizing \mathcal{L}_v brings \mathcal{X} and $\widetilde{\mathcal{M}}$ closer, which is equivalent to encouraging reconstruction on \mathcal{X} .

The overall loss function \mathcal{L}_m can be regarded as a *Regularized Autoencoder* (Ghosh et al. 2020) (RAE), where \mathcal{L}_v corresponds to the reconstruction loss of RAE, and the other losses (i.e., $\mathcal{L}_{\text{dist}}$ and $\mathcal{L}_{\text{proj}}$) are regularizers for the decoder.

Density Training Objectives In order to perform density estimation on \mathcal{M} , similar to \mathcal{M} -flow (Brehmer and Cranmer 2020), we employ a *density estimator* $h: \mathbb{R}^K \rightarrow \mathbb{R}^K$ which is trained to estimate the probability density on \mathcal{M} after the manifold training phase is done. Given a sample $x \in \mathcal{M}$, its density $p_{\mathcal{M}}(x)$ on \mathcal{M} is explicitly derived according to the following *change-of-variable formula* (Dinh, Krueger, and Bengio 2014; Kobyzev, Prince, and Brubaker 2020)

$$p_{\mathcal{M}}(x) = p_{\mathcal{U}}(u(x)) \left| \det J_g^\top(u(x)) J_g(u(x)) \right|^{-\frac{1}{2}} \quad (20)$$

$$= p_{\mathcal{U}}(u(x)), \quad (21)$$

where $u(x) = g^\dagger(x)$ is the embedding of x , and Eq. (21) is derived because J_g is orthonormal (see Prop. 1). The density estimator h is further involved in the following formula

$$p_{\mathcal{U}}(u) = p_{\widetilde{\mathcal{U}}}(h^{-1}(u)) \left| \det J_h(h^{-1}(u)) \right|^{-1}, \quad (22)$$

where h is a flow module (Brehmer and Cranmer 2020) that is a bijection with explicit inversion h^{-1} and tractable Jacobian determinant $\det J_h$, and the prior distribution $p_{\widetilde{\mathcal{U}}}$ is the standard normal distribution. Hence given $u(x)$, the density $p_{\mathcal{U}}(u(x))$ is tractable, and hence $p_{\mathcal{M}}(x) = p_{\mathcal{U}}(u(x))$ can be explicitly estimated. In terms of the training objective, we use maximum likelihood training, and the loss function is

$$\mathcal{L}_d = - \sum_{i=1}^N \log p_{\mathcal{U}}(u(x^{(i)})). \quad (23)$$

Algorithms We present the algorithms for manifold learning using our proposed HF model. As mentioned above, the training process consists of two phases, namely the manifold phase and the density phase. During the manifold phase, we detect the ground-truth dimensionality K^* of \mathcal{M} by utilizing our proposed two-stage dimensionality reduction algorithm, see Alg. 1. Specifically, for the first stage, we set an embedding dimensionality \widetilde{K} that is much lower than D but higher than K^* , then train the first stage generator $g_1: \mathbb{R}^{\widetilde{K}} \rightarrow \mathbb{R}^D$ using \mathcal{L}_m in Eq. (15). For the second stage, given a reconstruction error threshold t , we use the *Binary Search Algorithm* (BSA) to search the minimum dimensionality \widetilde{K} for the second stage generator $g_2: \mathbb{R}^{\widetilde{K}} \rightarrow \mathbb{R}^K$ such that the reconstruction error on \mathcal{X} of $\widetilde{g} = g_1 \circ g_2$ is not larger than t , where g_1 is not involved in the second stage training, and g_2 is trained on the embedding set $\{g_1^\dagger(x) | x \in \mathcal{X}\}$. After the

Algorithm 1: Two-stage Dimensionality Reduction

Require: Training dataset $\mathcal{X} = \{x^{(i)}\}_{i=1}^N \subset \mathbb{R}^D$, dimensionality K of the first stage, threshold t
 /* Stage 1 */

- 1: Initialize a generator $g_1 : \mathbb{R}^K \rightarrow \mathbb{R}^D$
- 2: Train g_1 using \mathcal{L}_m , then obtain $\mathcal{U} = \{u(x^{(i)})\}_{i=1}^N \subset \mathbb{R}^K$, where $u(x) = g_1^\dagger(x)$ is the embedding of x
 /* Stage 2 */
- 3: Initialize $K_1 \leftarrow 0, K_2 \leftarrow K, \tilde{K} \leftarrow \lfloor \frac{K_1+K_2}{2} \rfloor, \tilde{g} \leftarrow g_1$
- 4: **while** $K_1 < \tilde{K}$ **do**
- 5: Initialize another generator $g_2 : \mathbb{R}^{\tilde{K}} \rightarrow \mathbb{R}^{K_2}$
- 6: Train g_2 on \mathcal{U} using \mathcal{L}_m , then obtain $\hat{g} \leftarrow g_1 \circ g_2$
- 7: Estimate the mean reconstruction error \mathcal{E} as

$$\mathcal{E} = \frac{1}{N} \sum_{i=1}^N \left\| \hat{g} \circ \tilde{g}^\dagger(x^{(i)}) - x^{(i)} \right\| \quad (24)$$

- 8: If $\mathcal{E} > t$, update $K_2 \leftarrow \tilde{K}, g_1 \leftarrow \hat{g}, \tilde{g} \leftarrow \hat{g}, \mathcal{U} \leftarrow \{g_2^\dagger(u) | u \in \mathcal{U}\}$, otherwise update $K_1 \leftarrow \tilde{K}$
 - 9: Update $\tilde{K} \leftarrow \lfloor \frac{K_1+K_2}{2} \rfloor$
 - 10: **end while**
 - 11: Obtain the detected manifold dimensionality \tilde{K} as well as the final generator $\tilde{g} : \mathbb{R}^{\tilde{K}} \rightarrow \mathbb{R}^D$
-

manifold phase training is completed, the density estimator h is trained on the embedding set $\{\tilde{g}^\dagger(x) | x \in \mathcal{X}\}$ using \mathcal{L}_d . See supplementary for the overall training algorithm.

3 Relation to Prior Work

Generative Manifold Learning Our proposed HF model is related to generative models for manifold learning (Gropp, Atzmon, and Lipman 2020; Beitler, Sosnovik, and Smeulders 2021; Teng and Choromanska 2019; Lou et al. 2020; Caterini et al. 2021; Silvestri, Roos, and Ambrogioni 2022), see supplementary for discussion. Our proposed HF model can be regarded as a multi-layer extension of \mathcal{M} -flow, which is similar regarding the hierarchical architecture to the *Pseudo Invertible Encoders* (Beitler, Sosnovik, and Smeulders 2021) (PIEs). The hierarchical architecture allows us to detect the ground-truth dimensionality of the data manifold in a time-efficient manner by utilizing Alg. 1. The *Isometric Autoencoders* (Gropp, Atzmon, and Lipman 2020) (IAEs) propose an isometric loss for manifold learning, and we also propose isometric regularizations to constrain our HF model to satisfy the properties of *distance preserving* and *rigorous projection* with theoretical guarantees. Moreover, thanks to the *rigorous projection* property, we can replace the classical reconstruction loss based on the *encoding-decoding* process with \mathcal{L}_v that only involves the *encoding* process, which can reduce computational cost and hence speed up training. We compare a variety of models in their ability to achieve manifold learning goals as in Tbl. 1. As we can see, most methods lack the ability to satisfy the aforementioned two properties, which may affect the performance on downstream tasks such

Methods	Sampling	Infer.	Proj.	Density Est.
GAN	✓	×	×	×
VAE	✓	○	○	×
IAE	✓	✓	○	×
PIE	✓	○	○	✓
\mathcal{M} -flow	✓	○	○	✓
HF	✓	✓	✓	✓

Table 1: Comparison between different models in their ability to achieve a variety of manifold goals. Regarding *inference*, ✓ (resp., ○) means that the model is able to infer embeddings that satisfy (resp., are not guaranteed to satisfy) the *distance preserving* property. Regarding *projection*, ✓ (resp., ○) means that the model is able to project samples onto the manifold in a manner that satisfies (resp., is not guaranteed to satisfy) the *rigorous projection* property.

as *anomaly detection*, as we show in experimental results.

Dimensionality Reduction The dimensionality of embeddings for most manifold learning methods is prescribed, and is not guaranteed to be equal to the ground-truth dimensionality of the manifold. For real-world datasets such as human faces (Liu et al. 2015; Karras, Laine, and Aila 2019), the real dimensionality K^* of the manifold may never be known. In existing literatures (Brehmer and Cranmer 2020), K^* is typically obtained by a searching process based on the criterion that a drop in performance (e.g., the reconstruction error) is expected when the model manifold dimensionality becomes smaller than K^* . Therefore, a naive approach to detect K^* is to train model for each possible value of K^* . For our proposed Alg. 1, the searching is performed in the second stage for \tilde{K} . Compared to a brute-force search of \tilde{K} , the time complexity of the binary search algorithm is $\mathcal{O}(\log K)$, which is much superior to $\mathcal{O}(K)$ of brute-force search given that K is large. On the other hand, our theoretical analysis guarantees that the composed generator $\hat{g} = g_1 \circ g_2$ of the first and the second stage generators g_1, g_2 still satisfies the desired properties for manifold learning. Given that D is very large, although g_1 is a nonlinear mapping over high-dimensional spaces with high training time cost, g_1 is trained only once in the first stage. Though the searching process in the second stage involves multiple training of g_2 , each training can be quickly done since g_2 operates in low-dimensional spaces when \tilde{K} is small. Hence combining the training of the first and second stage results in a time-efficient algorithm for dimensionality reduction. See supplementary for experiments.

4 Experiments

We provide experimental results on a synthetic manifold for intuitive illustration in this section, and leave more results on natural image datasets in supplementary due to space limitation. All the implementation details for our experiments can also be found in supplementary.

For intuitive illustration, we use a 1-dimensional manifold $\mathcal{M} \triangleq \{(\cos \theta, \sin \theta) | \theta \in [\frac{\pi}{6}, \frac{5\pi}{6}]\}$ (i.e., a curve) residing in the 2-dimensional Euclidean space. We draw $N = 1,000$

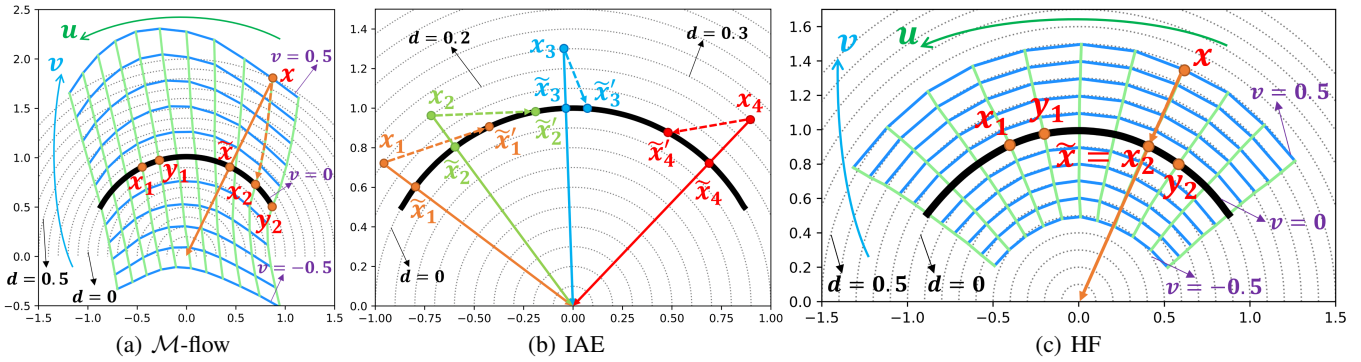


Figure 3: Qualitative results of different models trained on \mathcal{X} . For each model, the recovered manifold is depicted by using the thick black solid line, and we can intuitively observe that all the models perfectly recover the manifold curve \mathcal{M} . In the ambient space of \mathcal{M} , the contour lines with respect to distances to \mathcal{M} are plotted by using the thin grey dotted lines, namely samples on the same contour line have equal distances to \mathcal{M} . We use $d = \alpha$ to denote a contour line with distance to \mathcal{M} being α . We also plot the coordinate charts of the flow module $f_1 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ of the generator $g = f_1 \circ p_1$ for \mathcal{M} -flow (in Fig. 3(a)) and HF (in Fig. 3(c)), where both \mathcal{M} -flow and HF use a one-layer generator g . Specifically, the two embedding dimensions u and v of f_1 are represented in green and blue, respectively, namely samples on the same green (*resp.*, blue) line corresponds to the same u (*resp.*, v). Hence for \mathcal{M} -flow and HF, given an off-manifold sample x , the projection x_2 of x onto \mathcal{M} inferred by the model is given by the coordinate line of f_1 connecting x and the inferred projection x_2 , which is denoted by the dashed arrow line. The ground-truth projection \tilde{x} of x onto \mathcal{M} is given by the solid arrow line, which is the straight line connecting x and the origin $(0, 0)$, considering radial is the direction perpendicular to the tangent space of the arc \mathcal{M} . In terms of IAE, because the decoder $g : \mathbb{R}^1 \rightarrow \mathbb{R}^2$ does not induce a coordinate chart like \mathcal{M} -flow and HF, the projection \tilde{x}' of a given sample x onto \mathcal{M} inferred by the model is obtained by an *encoding-decoding* process, *i.e.*, $\tilde{x}' = g \circ e(x)$, where e is the encoder. In Fig. 3(b), we show the projections of $\{x_i\}_{i=1}^4$, where the ground-truth and the inferred projection of x_i onto \mathcal{M} are \tilde{x}_i and \tilde{x}'_i , respectively.

samples from a Gaussian density $\mathcal{N}(\frac{\pi}{2}, 1)$ that is restricted to \mathcal{M} , obtaining a training dataset

$$\mathcal{X} \triangleq \left\{ x^{(i)} \sim \mathcal{N}\left(\frac{\pi}{2}, 1\right) \wedge x^{(i)} \in \mathcal{M} \right\}_{i=1}^N \quad (25)$$

which we use to train different models. We only show qualitative results as in Fig. 3 due to space limitation, and present all quantitative results in supplementary.

Distance Preserving In order to evaluate \mathcal{M} -flow and HF intuitively, in Fig. 3(a) and Fig. 3(c), we refer to $\forall x, y \in \mathcal{M}$ that are intersections of two adjacent green coordinate lines with \mathcal{M} as a *pair*. Note that we depict coordinate charts for \mathcal{M} -flow and HF such that $|u(x) - u(y)|$ is constant for any pair x, y . Therefore, the *distance preserving* property of \mathcal{M} -flow and HF can be intuitively evaluated by observing if the manifold distance on \mathcal{M} (*i.e.*, the arc length) between x and y is constant for any pair x, y . From Fig. 3(a), we learn that \mathcal{M} -flow is not guaranteed to satisfy the *distance preserving* property due to $d_{\mathcal{M}}(x_1, y_1) \neq d_{\mathcal{M}}(x_2, y_2)$. However, from Fig. 3(c) we learn that our HF satisfies this property because we can intuitively observe that $d_{\mathcal{M}}(x_1, y_1) = d_{\mathcal{M}}(x_2, y_2)$ for any pairs x_1, y_1 and x_2, y_2 . In terms of IAE, the *distance preserving* property is not reflected in Fig. 3(b), see supplementary for quantitative evaluations.

Rigorous Projection Given an off-manifold x , we visualize the difference between \mathcal{M} -flow and HF in terms of projecting x onto \mathcal{M} . By comparing Fig. 3(a) with Fig. 3(c), we can see that the projection x_2 inferred by \mathcal{M} -flow deviates from the ground-truth projection \tilde{x} , meanwhile our HF gives

the correct projection $x_2 = \tilde{x}$. Moreover, because x is on the blue coordinate line $v = 0.5$ as we visualize in Fig. 3(a) and Fig. 3(c), we know that the distance from x to the projection x_2 given by the model is 0.5. On the other hand, x is on the contour line $d = 1.5$ (*resp.*, $d = 0.5$) for \mathcal{M} -flow (*resp.*, our HF), which means that \mathcal{M} -flow (*resp.*, our HF) gives biased (*resp.*, correct) distance from x to its inferred projection x_2 . From Fig. 3(b), we also learn that projections given by IAE are biased. Hence both \mathcal{M} -flow and IAE are not guaranteed to satisfy the *rigorous projection* property. As visualized in Fig. 3(c), the green coordinate lines are radial while the blue coordinate lines coincides with the contour lines, which suggests that HF satisfies the *rigorous projection* property. The dissatisfaction of the *rigorous projection* property affects the performance of the model on *anomaly detection* based on the inferred projection distance, see supplementary.

5 Conclusion

In this paper, we propose the *Hierarchical Flow* (HF) model constrained by isometric regularizations to combine several manifold learning goals in one unified framework. The HF model is theoretically guaranteed to satisfy the properties of *distance preserving* and *rigorous projection*, which helps in manifold learning and downstream tasks. We also propose a time-efficient two-stage algorithm for dimensionality reduction based on the hierarchical architecture of our HF model. Experimental results justify our theoretical analysis and verify the effectiveness of our proposed method.

Acknowledgements

The work was supported by the National Science Foundation of China (62076162), and the Shanghai Municipal Science and Technology Major/Key Project, China (2021SHZDZX0102, 20511100300).

References

- Basri, R.; and Jacobs, D. 2017. Efficient representation of low-dimensional manifolds using deep networks. In *ICLR*.
- Beitler, J. J.; Sosnovik, I.; and Smeulders, A. 2021. PIE: Pseudo-Invertible Encoder. *arXiv preprint arXiv:2111.00619*.
- Bose, J.; Smofsky, A.; Liao, R.; Panangaden, P.; and Hamilton, W. 2020. Latent variable modelling with hyperbolic normalizing flows. In *ICML*.
- Brehmer, J.; and Cranmer, K. 2020. Flows for simultaneous manifold learning and density estimation. In *NeurIPS*.
- Buades, A.; Coll, B.; and Morel, J.-M. 2005. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*.
- Carroll, J. D.; and Arabie, P. 1998. Multidimensional scaling. *Measurement, Judgment and Decision Making*.
- Caterini, A. L.; Loaiza-Ganem, G.; Pleiss, G.; and Cunningham, J. P. 2021. Rectangular flows for manifold learning. In *NeurIPS*.
- Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys*.
- Chui, C. K.; and Mhaskar, H. N. 2018. Deep nets for local manifold learning. *Frontiers in Applied Mathematics and Statistics*.
- Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.
- Fefferman, C.; Mitter, S.; and Narayanan, H. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society*.
- Gemici, M. C.; Rezende, D.; and Mohamed, S. 2016. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*.
- Geng, C.; Wang, J.; Chen, L.; Bao, W.; Chu, C.; and Gao, Z. 2020. Uniform interpolation constrained geodesic learning on data manifold. *arXiv preprint arXiv:2002.04829*.
- Ghosh, P.; Sajjadi, M. S.; Vergari, A.; Black, M.; and Schölkopf, B. 2020. From variational to deterministic autoencoders. In *ICLR*.
- Gisbrecht, A.; and Hammer, B. 2015. Data visualization by nonlinear dimensionality reduction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.
- Gong, H.; Pan, C.; Yang, Q.; Lu, H.; and Ma, S. 2006. Neural Network Modeling of Spectral Embedding. In *BMVC*.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.
- Gropp, A.; Atzmon, M.; and Lipman, Y. 2020. Isometric autoencoders. *arXiv preprint arXiv:2006.09289*.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*.
- He, X.; and Niyogi, P. 2003. Locality preserving projections. In *NeurIPS*.
- Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kobyzev, I.; Prince, S.; and Brubaker, M. 2020. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*.
- Lou, A.; Lim, D.; Katsman, I.; Huang, L.; Jiang, Q.; Lim, S. N.; and De Sa, C. M. 2020. Neural manifold ordinary differential equations. In *NeurIPS*.
- Mishne, G.; Shaham, U.; Cloninger, A.; and Cohen, I. 2019. Diffusion nets. *Applied and Computational Harmonic Analysis*.
- Pai, G.; Talmon, R.; Bronstein, A.; and Kimmel, R. 2019. Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *WACV*.
- Pan, Z.; Niu, L.; and Zhang, L. 2022. Unigan: reducing mode collapse using a uniform generator. In *NeurIPS*.
- Petersen, P. 2006. *Riemannian geometry*. Springer.
- Pless, R.; and Souvenir, R. 2009. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*.
- Rezende, D. J.; Papamakarios, G.; Racaniere, S.; Albergo, M.; Kanwar, G.; Shanahan, P.; and Cranmer, K. 2020. Normalizing flows on tori and spheres. In *ICML*.
- Rowes, S. T. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*.
- Silvestri, G.; Roos, D.; and Ambrogioni, L. 2022. Closing the gap: Exact maximum likelihood training of generative autoencoders using invertible layers. *arXiv preprint arXiv:2205.09546*.
- Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*.
- Teng, Y.; and Choromanska, A. 2019. Invertible autoencoder for domain adaptation. *Computation*.
- Van Der Maaten, L. 2009. Learning a parametric embedding by preserving local structure. In *AISTATS*.
- Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*.
- Verleysen, M.; and François, D. 2005. The curse of dimensionality in data mining and time series prediction. In *IWANN*.
- Weinberger, K. Q.; and Saul, L. K. 2006. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*.
- Yonghyeon, L.; Yoon, S.; Son, M.; and Park, F. C. 2021. Regularized Autoencoders for Isometric Representation Learning. In *ICLR*.