



Deep nonparametric estimation of intrinsic data structures by chart autoencoders: Generalization error and robustness [☆]

Hao Liu ^a, Alex Havrilla ^b, Rongjie Lai ^{c,*}, Wenjing Liao ^{b,*}

^a Math department, Hong Kong Baptist University, Hong Kong

^b School of Math, Georgia Tech, USA

^c Department of Mathematics, Purdue University, USA



ARTICLE INFO

Communicated by Radu Balan

Keywords:

Chart autoencoder
Deep learning theory
Generalization error
Dimension reduction
Manifold model

ABSTRACT

Autoencoders have demonstrated remarkable success in learning low-dimensional latent features of high-dimensional data across various applications. Assuming that data are sampled near a low-dimensional manifold, we employ chart autoencoders, which encode data into low-dimensional latent features on a collection of charts, preserving the topology and geometry of the data manifold. Our paper establishes statistical guarantees on the generalization error of chart autoencoders, and we demonstrate their denoising capabilities by considering n noisy training samples, along with their noise-free counterparts, on a d -dimensional manifold. By training autoencoders, we show that chart autoencoders can effectively denoise the input data with normal noise. We prove that, under proper network architectures, chart autoencoders achieve a squared generalization error in the order of $n^{-\frac{2}{d+2}} \log^4 n$, which depends on the intrinsic dimension of the manifold and only weakly depends on the ambient dimension and noise level. We further extend our theory on data with noise containing both normal and tangential components, where chart autoencoders still exhibit a denoising effect for the normal component. As a special case, our theory also applies to classical autoencoders, as long as the data manifold has a global parametrization. Our results provide a solid theoretical foundation for the effectiveness of autoencoders, which is further validated through several numerical experiments.

1. Introduction

High-dimensional data arise in many real-world machine learning problems, presenting new difficulties for both researchers and practitioners. For example, the ImageNet classification task [17] involves data points with 150,528 dimensions, derived from images of size $224 \times 224 \times 3$. Similarly, the MS-COCO object detection task [33] tackles data points with 921600 dimensions, stemming from images of size $480 \times 640 \times 3$. The well-known phenomenon of the curse of dimensionality states that, in many statistical learning and inference tasks, the required sample size for training must grow exponentially with respect to the dimensionality of the data, unless further assumptions are made. Due to this curse, directly working with high-dimensional datasets can result in subpar performance for many machine learning methods.

[☆] This research is partially supported by HKBU 179356, NSFC 12201530, HKRGC ECS 22302123, NSF DMS-2012652, NSF DMS-2145167 and NSF DMS-2134168.

* Corresponding authors.

E-mail addresses: haoliu@hkbu.edu.hk (H. Liu), ahavrilla3@gatech.edu (A. Havrilla), lairj@purdue.edu (R. Lai), wliaoj60@gatech.edu (W. Liao).

<https://doi.org/10.1016/j.acha.2023.101602>

Received 26 March 2023; Received in revised form 13 September 2023; Accepted 30 September 2023

Available online 12 October 2023

1063-5203/© 2023 Elsevier Inc. All rights reserved.

Fortunately, many real-world data are embedded in a high-dimensional space while exhibiting low-dimensional structures due to local regularities, global symmetries, or repetitive patterns. It has been shown in Pope et al. [44] that many benchmark datasets such as MNIST, CIFAR-10, MS-COCO and ImageNet have low intrinsic dimensions. In literature, a well-known mathematical model to capture such low-dimensional geometric structures in datasets is the manifold model, where data is assumed to be sampled on or near a low-dimensional manifold [56,48,19]. A series of works on manifold learning have been effective on linear dimension reduction of data, including IsoMap [56], Locally Linear Embedding [48,63], Laplacian Eigenmap [3], Diffusion map [16], t-SNE [60], Geometric Multi-Resolution Analysis [2,32] and many others [1]. As extensions, the noisy manifold setting has been studied in [39,21,20,45].

In recent years, deep learning has made significant successes on various machine learning tasks with high-dimensional data sets. Unlike traditional manifold learning methods which estimate the data manifold first and then perform statistical inference on the manifold, it is a common belief that deep neural networks can automatically capture the low-dimensional structures of the data manifold and utilize them for statistical inference. In order to justify the performance of deep neural networks, many mathematical theories have been established on function approximation [25,62,52,49,53,11,15,40,35,38], regression [14,12,41,23], classification [36], operator learning [37] and causal inference on a low-dimensional manifold [13]. In many of these works, a proper network architecture is constructed to approximate certain class of functions supported on a manifold. Regression, classification, operator learning and causal inference are further achieved with the constructed network architecture. The sample complexity critically depends on the intrinsic dimension of the manifold and only weakly depends on the ambient dimension.

Autoencoder is a special designed deep learning method to effectively learn low-dimensional features of data [9,30,24,34]. The conventional autoencoder consists of two subnetworks, an encoder and a decoder. The encoder transforms the high-dimensional input data into a lower-dimensional latent representation, capturing the intrinsic parameters of the data in a compact form. The decoder then maps these latent features to reconstruct the original input in the high-dimensional space. Inspired by the traditional autoencoder [4,46], many variants of autoencoder have been proposed. The most well-known variant of autoencoders is Variational Auto-Encoder (VAE) [26,27,47], which introduces a prior distribution in the latent space as a regularizer. This regularization ensures a better control of the distribution of latent features and helps to avoid overfitting. Recently, the excess risk of VAE via empirical Bayes estimation was analyzed in Tang and Yang [55]. The Denoising Auto-Encoder (DAE) [61,5] was proposed to denoise the input data in the process of feature extraction. By intentionally corrupting the input training data by noise, DAE has a denoising effect on the noisy test data and therefore has improved the robustness over the traditional autoencoders.

Although autoencoders have demonstrated great success in feature extraction and dimension reduction, its mathematical and statistical theories are still very limited. More importantly, the aforementioned autoencoders aim to globally map the data manifold to a subset in \mathbb{R}^d where d is the intrinsic dimension of the manifold. However, a global mapping may not always exist for manifolds with nontrivial geometry and topology. To address this issue, Schonsheck et al. [50] showed that conventional auto encoders using a flat Euclidean space can not represent manifolds with nontrivial topology, thus introduced a Chart Auto-Encoder (CAE) to capture local latent features. Instead of using a global mapping, CAE uses a collection of open sets to cover the manifold where each set is associated with a local mapping. Their numerical experiments have demonstrated that CAE can preserve the geometry and topology of data manifolds. They also obtained an approximation error of CAE in the noise-free setting. Specifically, Schonsheck et al. [50] constructed an encoder \mathcal{E} and a decoder \mathcal{D} that can optimize the empirical loss with the approximation error satisfying $\sup_{v \in M} \|v - \mathcal{D} \circ \mathcal{E}(v)\|_2 \leq \epsilon$, where M represents the low-dimensional manifold. In a recent work [51], CAE has been extended to semi-supervised manifold learning and has demonstrated great performances in differentiating data on nearby but disjoint manifolds.

In this paper, our focus is on CAE and we aim to extend previous results in two ways. Firstly, we establish statistical guarantees on the generalization error for the trained encoders and decoders, which are given by the global minimizer of the empirical loss. Secondly, our analysis considers data sampled on a manifold corrupted by noise, which is a more practical scenario. To the best of our knowledge, this type of analysis has not been conducted previously. The generalization error analysis is crucial in understanding the sample complexity of autoencoders. Additionally, the inclusion of noise in the error analysis is significant as it allows us to examine the impact of noise on CAE.

We briefly summarize our results as follows. To demonstrate the robustness of CAE, we allow for data sampled on a manifold corrupted by noise. Namely, we assume n pairs of clean and noisy data for training, where the clean data are sampled on a d -dimensional manifold, and the noisy data are perturbed from the clean data by noise. This setting is practically meaningful and has been considered in DAE [61,5] and multi-fidelity simulations [29,6,43]. We show that CAE results in an encoder $\hat{\mathcal{E}}$ and a decoder $\hat{\mathcal{D}}$ that have a denoising effect for the normal noise. That is, for any noisy test data \mathbf{x} , the output of $\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x})$ is close to its clean counterpart $\pi(\mathbf{x})$, which is the orthogonal projection of \mathbf{x} onto the manifold M . Our results, as summarized in Theorem 2, can be stated informally as follows:

Theorem (Informal). *Let M be a d -dimensional compact smooth Riemannian manifold isometrically embedded in \mathbb{R}^D with reach $\tau > 0$. Given a fixed noise level $q \in [0, \tau)$, we consider a training data set $S = \{(\mathbf{x}_i, \mathbf{v}_i)\}_{i=1}^n$ where the \mathbf{v}_i 's are i.i.d. samples from a probability measure on M , and $\mathbf{x}_i = \mathbf{v}_i + \mathbf{w}_i$'s are perturbed from the \mathbf{v}_i 's with independent random noise $\mathbf{w}_i \in T_{\mathbf{v}_i}^{\perp} M$ (the normal space of M at \mathbf{v}_i) whose distribution satisfies $\|\mathbf{w}\|_2 \leq q$. We denote the distribution of all \mathbf{x}_i by γ . Using proper network architectures, the encoder $\mathcal{E} : \mathbb{R}^D \rightarrow \mathbb{R}^{O(d)}$ and the decoder $\mathcal{D} : \mathbb{R}^{O(d)} \rightarrow \mathbb{R}^D$, we solve the empirical risk minimization problem in (8) to obtain the global minimizer $\hat{\mathcal{E}}$ and $\hat{\mathcal{D}}$. Then the expected generalization error of CAE satisfies*

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \leq C D^2 \log^3 D n^{-\frac{2}{d+2}} \log^4 n \quad (1)$$

where C is a constant independent of n and D .

This theorem highlights the robustness and effectiveness of CAE in learning the underlying data manifold. More specifically, with increasing sample size n , the generalization error converges to zero at a fast rate and its exponent depends only on the intrinsic dimension d , not the ambient dimension D . The theorem also shows that autoencoders have a strong denoising ability when dealing with noise on the normal directions, as the error approaches zero as n increases. The latent feature in every chart has a dimension of d , with the number of charts being dependent on the complexity of the manifold \mathcal{M} . In special cases where the manifold is globally homeomorphic to a subset of \mathbb{R}^d , this result can be applied to conventional autoencoders as described in Section 3.2.

Besides the case of normal noise in the aforementioned theorem, we also consider a general setting where the noise contains both normal and tangential components. In Theorem 3, we prove that CAE can denoise the normal component of the noise. Specifically, the squared generalization error is upper bounded by

$$C(D^2 \log^3 D) n^{-\frac{2}{2+d}} \log^4 n + C_1 \sigma^2,$$

where σ^2 is the second moment of the tangential component of the noise. Our result is consistent with the existing works in manifold learning [21,45] which demonstrates that denoising is possible for normal noise but impossible for tangential noise. A detailed explanation is given at the end of Section 3.3.

The rest of the paper is organized as follows: In Section 2, we introduce background related to manifolds and neural networks to be used in this paper. In Section 3, we present our problem setting and main results including single chart case, multi-chart case and extension to general noise. We defer theoretical proof in Section 5. We validate our network architectures and theories by several experiments in Section 4. We conclude the paper in Section 6.

Notation: We use lower-case letters to denote scalars, lower-case bold letters to denote vectors, upper-case letters to denote matrices and constants, calligraphic letters to denote manifolds, sets and function classes. For a vector valued function $\mathbf{f} = [f_1, \dots, f_d]^\top$ defined on Ω , we let $\|\mathbf{f}\|_{L^{\infty,\infty}} := \sup_{\mathbf{x} \in \Omega} \max_k |f_k(\mathbf{x})|$.

2. Preliminary

In this section, we briefly introduce the preliminaries on manifolds and neural networks to be used in this paper.

2.1. Manifolds

We first introduce some definitions and notations about manifolds. More details can be found in [58,31]. Let \mathcal{M} be a d -dimensional Riemannian manifold isometrically embedded in \mathbb{R}^D . A chart of \mathcal{M} defines a local neighborhood and coordinates on \mathcal{M} .

Definition 1. A chart of \mathcal{M} is a pair (U, ϕ) where $U \subset \mathcal{M}$ is an open set, and $\phi : U \rightarrow \mathbb{R}^d$ is a homeomorphism, i.e., ϕ is bijective and both ϕ and ϕ^{-1} are continuous. A C^s atlas of \mathcal{M} is a collection of charts $\{(U_k, \phi_k)\}_{k \in \mathcal{K}}$ which satisfies $\cup_{k \in \mathcal{K}} U_k = \mathcal{M}$, and are pairwise C^s compatible:

$$\phi_{k_1} \circ \phi_{k_2}^{-1} : \phi_{k_2}(U_{k_1} \cap U_{k_2}) \rightarrow \phi_{k_1}(U_{k_1} \cap U_{k_2}) \quad \text{and} \quad \phi_{k_2} \circ \phi_{k_1}^{-1} : \phi_{k_1}(U_{k_1} \cap U_{k_2}) \rightarrow \phi_{k_2}(U_{k_1} \cap U_{k_2})$$

are both C^s for any $k_1, k_2 \in \mathcal{K}$. An atlas is called finite if it contains finite many charts. Here C^s denotes the space of functions with continuous derivatives up to s order.

A smooth manifold is a manifold with a C^∞ atlas. Commonly used smooth manifolds include the Euclidean space, torus, sphere and Grassmannian. C^s functions on a smooth manifold \mathcal{M} can be defined as follows:

Definition 2 (C^s functions on a smooth manifold). Let \mathcal{M} be a smooth manifold and $f : \mathcal{M} \rightarrow \mathbb{R}$ be a function on \mathcal{M} . The function f is a C^s function on \mathcal{M} if for every chart (U, ϕ) of \mathcal{M} , the function $f \circ \phi^{-1} : \phi(U) \rightarrow \mathbb{R}$ is a C^s function.

We next define the C^∞ partition of unity of \mathcal{M} .

Definition 3 (Partition of unity). A C^∞ partition of unity of a manifold \mathcal{M} is a collection of C^∞ functions $\{\rho_k\}_{k \in \mathcal{K}}$ with $\rho_k : \mathcal{M} \rightarrow [0, 1]$ such that for any $\mathbf{x} \in \mathcal{M}$,

1. there is a neighborhood of \mathbf{x} where only a finite number of the functions in $\{\rho_k\}_{k \in \mathcal{K}}$ are nonzero, and
2. $\sum_{k \in \mathcal{K}} \rho_k(\mathbf{x}) = 1$.

We say an open cover is locally finite if every $\mathbf{v} \in \mathcal{M}$ has a neighborhood that intersects with a finite number of sets in the cover. It is well-known that for any locally finite cover of \mathcal{M} , a C^∞ partition of unity that subordinates to this cover always exists [54, Chapter 2, Theorem 15].

Reach is an important quantity of a manifold that is related to curvature. For any $\mathbf{x} \in \mathbb{R}^D$, we write $d(\mathbf{x}, \mathcal{M}) := \inf_{\mathbf{v} \in \mathcal{M}} \|\mathbf{x} - \mathbf{v}\|_2$ the distance from \mathbf{x} to \mathcal{M} . Reach is defined as follows:

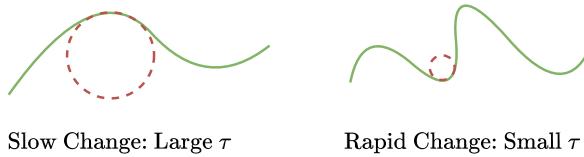


Fig. 1. Illustration of manifolds with large and small reach.

Definition 4 (Reach [18,42]). The reach of \mathcal{M} is defined as

$$\tau = \inf_{v \in \mathcal{M}} \inf_{x \in G} \|x - v\|_2, \quad (2)$$

where $G = \{x \in \mathbb{R}^D : \exists \text{ distinct } p, q \in \mathcal{M} \text{ such that } d(x, \mathcal{M}) = \|x - p\|_2 = \|x - q\|_2\}$ is the medial axis of \mathcal{M} .

Roughly speaking, a manifold with a small reach can “bend” faster than that with a large reach. For example, a plane has a reach equal to infinity. A hyper-sphere with radius r has a reach r . We illustrate manifolds with a large reach and small reach in Fig. 1.

We denote the reach of \mathcal{M} by τ , the tangent plane of \mathcal{M} at $v \in \mathcal{M}$ by $T_v \mathcal{M}$ and its orthogonal space by $T_v^\perp \mathcal{M}$. Define the q -neighborhood of \mathcal{M} by

$$\mathcal{M}(q) = \{x \in \mathbb{R}^D : \inf_{v \in \mathcal{M}} \|x - v\|_2 \leq q\}. \quad (3)$$

When $q < \tau$, by the property of reach, every $x \in \mathcal{M}(q)$ has a unique decomposition

$$x = \pi(x) + w \quad (4)$$

where $\pi(x) = \operatorname{argmin}_{v \in \mathcal{M}} \|v - x\|_2$ and $w \in T_v^\perp \mathcal{M}$ [42,15].

2.2. Neural networks

In this paper, we consider feedforward neural networks (FNN) with the rectified linear unit $\operatorname{ReLU}(a) = \max\{a, 0\}$. An FNN with L layers is defined as

$$f(x) = W_L \cdot \operatorname{ReLU}(W_{L-1} \cdots \operatorname{ReLU}(W_1 x + b_1) + \cdots + b_{L-1}) + b_L, \quad (5)$$

where the W_i ’s are weight matrices, the b_i ’s are bias vectors, and ReLU is applied element-wisely. We define a class of neural networks with inputs in \mathbb{R}^D and outputs in \mathbb{R}^d as

$$\mathcal{F}(D, d; L, p, K, \kappa, R) = \{f : \mathbb{R}^D \rightarrow \mathbb{R}^d \mid f \text{ has the form of (5) with } L \text{ layers and width bounded by } p,$$

$$\begin{aligned} \|f\|_\infty &\leq R, \sum_{i=1}^L \|W_i\|_0 + \|b_i\|_0 \leq K, \\ \|W_i\|_{\infty, \infty} &\leq \kappa, \|b_i\|_\infty \leq \kappa \text{ for } i = 1, \dots, L, \end{aligned}$$

where $\|H\|_{\infty, \infty} = \max_{i,j} |H_{ij}|$ for a matrix H and $\|\cdot\|_0$ denotes the number of non-zero elements of its argument. Above, the width of a network is the largest output dimension among all layers.

3. Main results

3.1. Problem setup for bounded normal noise

We consider the noisy setting where training data contain n pairs of clean and noisy data:

Setting 1. Let \mathcal{M} be a d -dimensional compact smooth Riemannian manifold isometrically embedded in \mathbb{R}^D with reach τ . Given a fixed noise level $q \in [0, \tau]$, we consider a training data set $S = \{(x_i, v_i)\}_{i=1}^n$ where the v_i ’s are i.i.d. samples from a probability measure on \mathcal{M} , and the x_i ’s are perturbed from the v_i ’s according to the model such that

$$x = v + w \quad (6)$$

where $w \in T_v^\perp \mathcal{M}$ (the normal space of \mathcal{M} at v) is a random vector satisfying $\|w\|_2 \leq q$. We denote the distribution of x by γ . In particular, we have $x_i = v_i + w_i$, where the w_i ’s are independent.

Setting 1 has two important implications:

(i) \mathcal{M} is bounded: there exists a constant $B > 0$ such that for any $\mathbf{x} \in \mathcal{M}(q)$,

$$\|\mathbf{x}\|_\infty \leq B. \quad (7)$$

(ii) \mathcal{M} has a positive reach [57, Proposition 14], denoted by $\tau > 0$.

The \mathbf{v}_i 's in Setting 1 represent the noise-free training data, and the \mathbf{x}_i 's are the noisy data perturbed by the normal noise \mathbf{w}_i 's. This noisy setting shares some similarity to the Denoising Auto-Encoder (DAE) [61,5] and multifidelity simulations [29,6,43]. DAE is widely used in image processing to train autoencoders with a denoising effect. In the DAE setting, one has clean samples, and then manually adds noise to the clean samples to obtain noisy samples. During training, the noisy samples are taken as the inputs and the clean samples are the outputs, such that the autoencoder is trained to denoise the noisy samples. In uncertainty quantification and prediction of random fields, it is expensive to simulate high-fidelity solutions. A popular strategy is to use a cheaper low-fidelity simulation as a surrogate and then a correction step is applied to modify the surrogate towards high-fidelity data. The correction operations are determined using both low-fidelity and high-fidelity data. Such a strategy is similar to our Setting 1: one can take the high-fidelity data as noise-free data and low-fidelity data as noisy data.

We first consider normal noise on the manifold \mathcal{M} in Setting 1. Given a training data set $S = \{(\mathbf{x}_i, \mathbf{v}_i)\}_{i=1}^n$, our goal is to theoretically analyze how the manifold structure of data can be learned based an encoder $\hat{\mathcal{E}} : \mathcal{M}(q) \rightarrow \mathbb{R}^{O(d)}$ and the corresponding decoder $\hat{\mathcal{D}} : \mathbb{R}^{O(d)} \rightarrow \mathbb{R}^D$ by minimizing the empirical mean squared loss

$$(\hat{\mathcal{D}}, \hat{\mathcal{E}}) = \underset{\mathcal{D} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}, \mathcal{E} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{v}_i - \mathcal{D} \circ \mathcal{E}(\mathbf{x}_i)\|_2^2, \quad (8)$$

for properly designed network architectures $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$. We evaluate the performance of $(\hat{\mathcal{D}}, \hat{\mathcal{E}})$ through the squared generalization error

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2, \quad (9)$$

at a noisy test point \mathbf{x} sampled from the same distribution γ as the training data. This paper establishes upper bounds on the squared generalization error of CAE with properly chosen network architectures. We first consider the single-chart case in Section 3.2 where \mathcal{M} is globally homeomorphic to a subset of \mathbb{R}^d . The general multi-chart case is studied in Section 3.3. In Section 3.4, we will study a more general setting that allows high-dimensional noise in the ambient space, under Setting 2.

3.2. Single-chart case

We start from a simple case where \mathcal{M} has a global low-dimensional parametrization. In other words, data on \mathcal{M} can be encoded to a d -dimensional latent feature through a global mapping.

Assumption 1 (Single-chart case). Assume \mathcal{M} has a global d -dimensional parameterization: There exist $\Lambda > 0$ and smooth maps $\mathbf{f} : \mathcal{M} \rightarrow [-\Lambda, \Lambda]^d$ and $\mathbf{g} : [-\Lambda, \Lambda]^d \rightarrow \mathcal{M}$ such that

$$\mathbf{v} = \mathbf{g} \circ \mathbf{f}(\mathbf{v}), \quad (10)$$

for any $\mathbf{v} \in \mathcal{M}$.

Assumption 1 implies that there exists an atlas of \mathcal{M} consisting of only one chart $(\mathcal{M}, \mathbf{f})$. This single-chart case serves as the mathematical model of autoencoders where one can learn a global low-dimensional representation of data without losing much information.

Our first result gives an upper bound on the generalization error (9) with properly chosen network architectures.

Theorem 1. In Setting 1, suppose Assumption 1 holds. Let $\hat{\mathcal{E}}, \hat{\mathcal{D}}$ be a global minimizer in (8) with the network classes $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, d; L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(d, D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ where

$$L_{\mathcal{E}} = O\left(\log^2 n + \log D\right), \quad p_{\mathcal{E}} = O\left(Dn^{\frac{d}{d+2}}\right), \quad K_{\mathcal{E}} = O\left(Dn^{\frac{d}{d+2}} \log^2 n + D \log D\right),$$

$$\kappa_{\mathcal{E}} = O\left(n^{\frac{2}{d+2}}\right), \quad R_{\mathcal{E}} = \Lambda, \quad (11)$$

$$L_{\mathcal{D}} = O(\log n), \quad p_{\mathcal{D}} = O\left(n^{\frac{d}{d+2}}\right), \quad K_{\mathcal{D}} = O\left(n^{\frac{d}{d+2}} \log^2 n\right), \quad \kappa_{\mathcal{D}} = O\left(n^{\frac{1}{d+2}}\right), \quad R_{\mathcal{D}} = B. \quad (12)$$

Then, we have the following upper bound of the squared generalization error

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \leq C (D^2 \log^2 D) n^{-\frac{2}{d+2}} \log^4 n \quad (13)$$

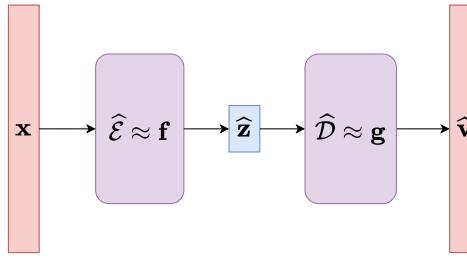


Fig. 2. An illustration of the network architectures in Theorem 1, where $\hat{\mathcal{E}}$ is the estimated encoder, $\hat{\mathcal{D}}$ is the estimated decoder, $\hat{\mathbf{z}}$ is the encoded latent feature, $\hat{\mathbf{v}}$ is the decoded vector.

for some constant C depending on d, B, τ, q , the Lipschitz constant of \mathbf{f} and \mathbf{g} , and the volume of \mathcal{M} . The constants hidden in the O depend on d, B, Λ, τ, q , the volume of \mathcal{M} and the Lipschitz constant of \mathbf{f} and \mathbf{g} .

We defer the detailed proof of Theorem 1 in Section 5.1. Theorem 1 has several implications:

- (i) **Fast convergence of the squared generalization error and the denoising effect:** When the network architectures are properly set, we can learn an autoencoder and the corresponding decoder so that the squared generalization error converges at a fast rate in the order of $n^{-\frac{2}{d+2}} \log^4 n$. Such a rate crucially depends on the intrinsic dimension d instead of the ambient dimension D , and therefore mitigates the curse of ambient space dimensionality. In addition, the error bound also suggests that the autoencoder has a denoising effect as the network output $\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x})$ converges to its clean counterpart $\pi(\mathbf{x})$ as n increases.
- (ii) **Geometric representation of data:** When the manifold \mathcal{M} has a global d -dimensional parameterization, the autoencoder $\hat{\mathcal{E}}$ outputs a d -dimensional latent feature, which serves as a geometric representation of the high-dimensional input \mathbf{x} .
- (iii) **Network size:** The network size critically depends on d , and weakly depends on D .

Remark 1. We remark that the constant hidden in the upper bound in Theorem 1 (and for the constant in Theorem 2) depends on $1/(\tau - q)$: as q gets closer to τ , the constant factor becomes larger. This is easy to understand: if q is very close to τ , some data are close to the medial axis of \mathcal{M} (see Definition 4). Assume $\mathbf{x} = \mathbf{v} + \mathbf{w}$ for $\mathbf{v} \in \mathcal{M}$ and $\mathbf{w} \in T_{\mathbf{v}}^{\perp} \mathcal{M}$ with $\|\mathbf{w}\|_2$ being very close to τ . Then there exists $\mathbf{v}' (\neq \mathbf{v})$ on \mathcal{M} such that $\|\mathbf{v}' - \mathbf{x}\|_2$ is very close to $\|\mathbf{v} - \mathbf{x}\|_2$. Thus, a small perturbation in \mathbf{x} might lead to a big change in $\pi(\mathbf{x})$, which makes the projection unstable.

We briefly introduce the proof idea of Theorem 1 in four steps.

Step 1: Decomposing the error. We decompose the squared generalization error (9) into a squared bias term and a variance term. The bias term captures the network's approximation error and the variance term captures the stochastic error.

Step 2: Bounding the bias term. To derive an upper bound of the bias term, we first define the oracle encoder and decoder. According to Assumption 1, \mathbf{f} is an encoder and \mathbf{g} is a decoder of \mathcal{M} . However, since the input data is in $\mathcal{M}(q)$, we cannot directly use \mathbf{f} as the encoder since \mathbf{f} is defined on \mathcal{M} . Utilizing the projection operator π , we define the oracle encoder as $\mathcal{E} = \mathbf{f} \circ \pi$, and simply define the oracle decoder as $\mathcal{D} = \mathbf{g}$. Based on Cloninger and Klock [15], we design the encoder network to approximate the oracle encoder $\mathbf{f} \circ \pi$. The decoder network is designed to approximate \mathbf{g} . Our network architecture is illustrated in Fig. 2. Based on our network construction, we derive an upper bound of the bias term showing that our encoder and decoder networks can approximate the oracles \mathcal{E} and \mathcal{D} respectively to an arbitrary accuracy ϵ (see Lemma 1).

Step 3: Bounding the variance term. The upper bound for the variance term is derived using metric entropy arguments [59,22], which depends on the network size (see Lemma 2).

Step 4: Putting the upper bound for both terms together. We finally put the upper bounds of the squared bias and variance term together. After balancing the approximation error and the network size, we can prove Theorem 1.

3.3. Multi-chart case

We next consider a more general case where the manifold has a complicated topology requiring multiple charts in an atlas. Consider an atlas $\{(U_k, \phi_k)\}_{k=1}^{C_{\mathcal{M}}}$ of \mathcal{M} so that each U_k is a local neighborhood on the manifold homeomorphic to a subset of \mathbb{R}^d . Here $C_{\mathcal{M}}$ denotes the number of charts in this atlas. In this atlas, each U_k has a d -dimensional parametrization of \mathcal{M} and data can be locally represented by d -dimensional latent features. In this general case, we prove the following upper bound of the squared generalization error (9) for CAE.

Theorem 2. Consider Setting 1. Let $\hat{\mathcal{E}}, \hat{\mathcal{D}}$ be a global minimizer of (8) with the network classes $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(C_{\mathcal{M}}(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ where $C_{\mathcal{M}} = O((d \log d)(4/\tau)^d)$,

$$L_{\mathcal{E}} = O(\log^2 n + \log D), \quad p_{\mathcal{E}} = O(D n^{\frac{d}{d+2}}), \quad K_{\mathcal{E}} = O((D \log D) n^{\frac{d}{d+2}} \log^2 n),$$

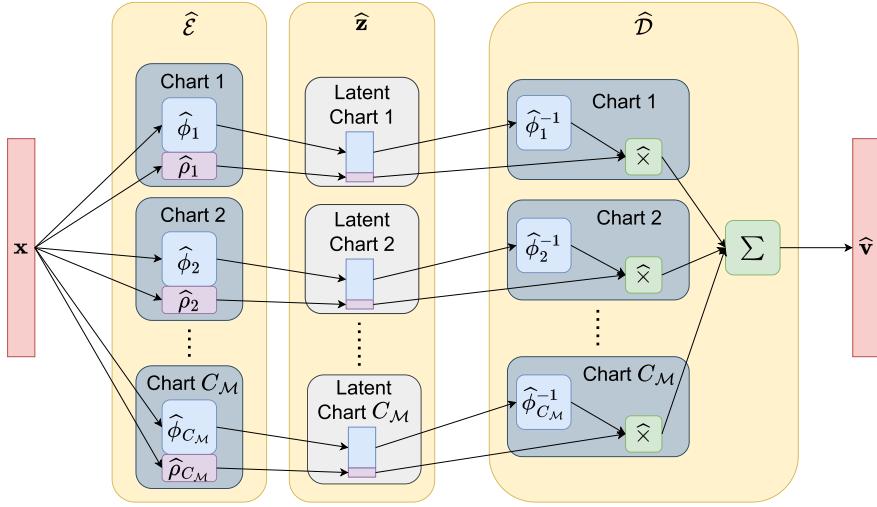


Fig. 3. Illustration of the network architectures in Theorem 2. Here $\hat{\mathcal{E}}$ is the estimated encoder, $\hat{\mathcal{D}}$ is the estimated decoder. The output of $\hat{\mathcal{E}}$ gives rise to the latent feature $\hat{\mathbf{z}}$, and the final output of the decoder $\hat{\mathcal{D}}$ gives rise to $\hat{\mathbf{v}}$ as an approximation of the clean sample \mathbf{v} .

$$\kappa_{\mathcal{E}} = O(n^{\frac{2}{d+2}}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}, \quad (14)$$

$$L_{\mathcal{D}} = O(\log^2 n + \log D), \quad p_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D),$$

$$\kappa_{\mathcal{D}} = O(n^{\frac{1}{d+2}}), \quad R_{\mathcal{D}} = B. \quad (15)$$

Then, we have the following upper bound of the squared generalization error

$$\mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \leq C(D^2 \log^3 D) n^{-\frac{2}{d+2}} \log^4 n \quad (16)$$

for some constant C depending on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

Theorem 2 indicates that for a general smooth manifold, the squared generalization error converges in the order of $n^{-\frac{2}{d+2}} \log^4 n$. In the case of multiple charts, the manifold has a more complicated structure than the single-chart case. Compared to the network architectures in Theorem 1, the network architecture specified in Theorem 2 has the following changes:

- The output of encoder has dimension $C_{\mathcal{M}}(d+1)$ instead of d . Such an increment in dimension is due to that $C_{\mathcal{M}}$ charts are needed to cover complicated manifolds.
- The encoder network uses more parameters: The number of nonzero parameters K has an additional factor $\log D$.
- The decoder network is deeper, wider and uses more parameters: The depth has an additional factor $\log n$, the width has an additional factor D , and the number of nonzero parameters has an additional factor D .

We remark that the number of charts $C_{\mathcal{M}} = O((d \log d)(4/\tau)^d)$ occurs in the worst case scenario. If \mathcal{M} has some good properties so that fewer charts are needed to cover \mathcal{M} , the result in Theorem 2 holds by replacing $C_{\mathcal{M}}$ by the actual number of charts needed.

We defer the detailed proof of Theorem 2 to Section 5.2. The proof idea of Theorem 2 is similar to that of Theorem 1. We also decompose the squared generalization error into a squared bias term and a variance term. The bias term is controlled by neural network approximation theories of the oracles. We briefly discuss the oracles and our network construction in Theorem 2 here. Let $\{\mathbf{U}_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$ be an atlas of \mathcal{M} and $\{\rho_j\}_{j=1}^{C_{\mathcal{M}}}$ be the partition of unity that subordinates to the atlas. For any $\mathbf{x} \in \mathcal{M}(q)$, we have

$$\mathbf{v} = \pi(\mathbf{x}) = \sum_{j=1}^{C_{\mathcal{M}}} \left[\phi_j^{-1} \circ \phi_j \circ \pi(\mathbf{x}) \right] \times [\rho_j \circ \pi(\mathbf{x})]. \quad (17)$$

If $\pi(\mathbf{x})$ is on the j th chart, (17) gives rise to an encoding of $\mathbf{x} \in \mathcal{M}(q)$ to the local coordinate $\phi_j \circ \pi(\mathbf{x}) \in \mathbb{R}^d$ and the partition of unity value $\rho_j \circ \pi(\mathbf{x}) \in \mathbb{R}$. We consider $[(\phi_j \circ \pi)^T \quad \rho_j \circ \pi]^T$ as the oracle encoder on the j th chart, and the collection of $\{[(\phi_j \circ \pi)^T \quad \rho_j \circ \pi]^T\}_{j=1}^{C_{\mathcal{M}}}$ as the global encoder. The latent feature on a single chart is of dimension $d+1$, and the latent feature on the whole manifold is of dimension $C_{\mathcal{M}}(d+1)$ with $C_{\mathcal{M}}$ charts. For any latent feature $\{\mathbf{z}_j \in \mathbb{R}^{d+1}\}_{j=1}^{C_{\mathcal{M}}}$, we consider the oracle decoder $\sum_{j=1}^{C_{\mathcal{M}}} \phi_j^{-1}((\mathbf{z}_j)_{1:d}) \times (\mathbf{z}_j)_{d+1}$ where $(\mathbf{z}_j)_{1:d}$ and $(\mathbf{z}_j)_{d+1}$ represent the first d and the $(d+1)$ th entries of \mathbf{z}_j respectively. To bound the bias in Theorem 2, we design neural networks to approximate the oracle encoder and decoder with an arbitrary accuracy ε (see Lemma 4 and its proof). The overall architecture of the encoder and decoder is illustrated in Fig. 3.

Table 1

Summary of the results on manifold learning with noise-free data. In the table, $\widehat{\mathcal{M}}$ denotes the manifold learned from training data, $\text{dist}(x, \mathcal{M})$ denotes the distance from x to \mathcal{M} , $\text{Proj}_{\widehat{\mathcal{M}}}(x)$ denotes the projection of x onto $\widehat{\mathcal{M}}$.

Reference	Method	Error measurements	Upper bound
[10]	k-Means	$\mathbb{E}_S \mathbb{E}_{x \in \mathcal{M}} [\text{dist}^2(x, \widehat{\mathcal{M}})]$	$n^{-\frac{1}{d+2}}$
[10]	k-Flats	$\mathbb{E}_S \mathbb{E}_{x \in \mathcal{M}} [\text{dist}^2(x, \widehat{\mathcal{M}})]$	$n^{-\frac{2}{d+4}}$
[32]	Multiscale	$\mathbb{E}_S \mathbb{E}_{x \in \mathcal{M}} [\ \text{Proj}_{\widehat{\mathcal{M}}}(x) - x\ ^2]$	$n^{-\frac{4}{d+2}}$
[50]	Chart Auto-Encoder	$\sup_{x \in \mathcal{M}} \ \widehat{\mathcal{D}} \circ \widehat{\mathcal{E}}(x) - x\ _2^2$	$n^{-\frac{1}{d+1}}$
Theorem 2	Chart Auto-Encoder	$\mathbb{E}_S \mathbb{E}_{x \sim \gamma} \ \widehat{\mathcal{D}} \circ \widehat{\mathcal{E}}(x) - x\ _2^2$	$n^{-\frac{2}{d+2}}$

We next discuss the connection between our results and some existing works. Low-dimensional approximations of manifolds have been studied in the manifold learning literature, with classical methods, such as k-means and k-flats [10], multiscale linear approximations [39,32]. Canas et al. [10], Liao and Maggioni [32] and [50] consider the noise-free setting, where training and test data are exactly located on a low-dimensional manifold. This is comparable to our noise-free setting with $q = 0$. We summarize the upper bounds in these works and our Theorem 2 in Table 1. While the rate from Liao and Maggioni [32] is faster than ours, $O\left(n^{\frac{1}{d+2}}\right)$ local tangent planes are used to approximate \mathcal{M} . In comparison, our Theorem 2 requires a fixed number $C_{\mathcal{M}}$ (at most $O((d \log d)(4/\tau)^d)$) local pieces (charts), which is independent of the sample size n . [50] first considers a Chart Auto-Encoder where their analysis is for an approximation error. Given n data samples uniformly distributed on \mathcal{M} , Schonsheck et al. [50] explicitly constructs the encoder $\widehat{\mathcal{E}}$ and decoder $\widehat{\mathcal{D}}$ and shows that with high probability, the constructed autoencoder gives rise to the approximation error in Table 1. Our analysis in this paper extends the theory to the noisy setting and establishes a statistical estimation theory with an improved rate of convergence on the mean squared generalization error, which is beyond the approximation error analysis.

Our noisy setting shares some similarities with [21] and [45], which focuses on manifold learning from noisy data. [21] assumes that the training data are corrupted by normal noise. In their setting, the noise follows a uniform distribution along the manifold normal direction and only noisy data (without the clean counterparts) are given for training. The authors proved that the lower bound measured by Hausdorff distance is in the order of $n^{-\frac{2}{d+4}}$, while no efficient algorithm is proposed to achieve this error bound. Recently, [45] considers a more general distribution of noise (not restricted to normal noise) but assumes the noise magnitude decays with a certain rate as n increases. In comparison, our work and [21] do not require the noise magnitude to decay as n increases. The great advantage of [45], as well as [21], is that only noisy data are required for training. [45] also derived a lower bound measured by Hausdorff distance in the order of $\frac{q^2 b^2}{\tau^3} \vee \tau^{-1} \left(\frac{q^2 \tau^2 \log n}{n} \right)^{\frac{2}{d+4}}$, where b denotes an upper bound of the magnitude of the tangential component of noise. The lower bound is different from the one in [21], because the existence of tangential noise, even with very small magnitude. Our work requires both clean and noisy data for training, which is possible when the training data are well-controlled. Our goal is to establish a theoretical foundation for the widely used autoencoders.

3.4. Extension to general noise with bounded normal components

We consider a more general setting in which the noise includes both normal and tangential components.

Setting 2. Let \mathcal{M} be a d -dimensional compact smooth Riemannian manifold isometrically embedded in \mathbb{R}^D with reach τ . Given a fixed noise level $q \in [0, \tau]$, we consider a training data set $S = \{(x_i, v_i)\}_{i=1}^n$ where the v_i 's are i.i.d. samples from a probability measure on \mathcal{M} , and the x_i 's are perturbed from the v_i 's according to the model such that

$$x = v + n \tag{18}$$

where $n \in \mathbb{R}^D$ is a random vector satisfying

$$\|\text{Proj}_{T_v^\perp \mathcal{M}}(n)\|_2 \leq q, \quad \mathbb{E} \left[\|\text{Proj}_{T_v^\perp \mathcal{M}}(n)\|_2^2 | v \right] \leq \sigma^2$$

with $\sigma \geq 0$. Here $\text{Proj}_{T_v \mathcal{M}}(n)$ and $\text{Proj}_{T_v^\perp \mathcal{M}}(n)$ denote the orthogonal projections of n onto the tangent space $T_v \mathcal{M}$ and the normal space $T_v^\perp \mathcal{M}$ respectively. We denote the distribution of x by γ . In particular, we have $x_i = v_i + n_i$, where the n_i 's are independent.

For any noise vector n , we can decompose it into the normal component $\text{Proj}_{T_v^\perp \mathcal{M}}(n)$ and the tangential component $\text{Proj}_{T_v \mathcal{M}}(n)$:

$$n = \text{Proj}_{T_v^\perp \mathcal{M}}(n) + \text{Proj}_{T_v \mathcal{M}}(n).$$

Setting 2 requires the magnitude of the normal component to be bounded by q , and for any $v \in \mathcal{M}$, the second moment of the tangential component is bounded by σ^2 . In particular, if we further have $\mathbb{E} [\text{Proj}_{T_v \mathcal{M}}(n) | v] = 0$, Setting 2 implies for any $v \in \mathcal{M}$, the tangential component has variance no larger than σ^2 .

Table 2
Comparison of the network architectures in Theorem 1, 2 and 3.

	Theorem 1	Theorem 2, 3
$\mathcal{F}_{\text{NN}}^{\mathcal{E}}$	dim. of input, output	D, d
	$L_{\mathcal{E}}$	$O(\log^2 n + \log D)$
	$p_{\mathcal{E}}$	$O(Dn^{\frac{d}{d+2}})$
	$K_{\mathcal{E}}$	$O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D)$
	$\kappa_{\mathcal{E}}$	$O(n^{\frac{2}{d+2}})$
$\mathcal{F}_{\text{NN}}^{\mathcal{D}}$	$R_{\mathcal{E}}$	Λ
	dim. of input, output	d, D
	$L_{\mathcal{D}}$	$O(\log n)$
	$p_{\mathcal{D}}$	$O(n^{\frac{2}{d+2}})$
	$K_{\mathcal{D}}$	$O(n^{\frac{d}{d+2}} \log^2 n)$
$\mathcal{F}_{\text{NN}}^{\mathcal{D}}$	$\kappa_{\mathcal{D}}$	$O(n^{\frac{1}{d+2}})$
	$R_{\mathcal{D}}$	B
		$\max\{\tau/4, 1\}$

Under Setting 2, the squared generalization error (9) is not appropriate any more, since our goal is to recover \mathbf{v} and $\pi(\mathbf{x})$ is not necessarily equal to \mathbf{v} . Instead, we consider the following squared generalization error

$$\mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \mathbf{v}\|_2^2.$$

We have the following upper bound on the squared generalization error

Theorem 3. Consider Setting 2. Let $\hat{\mathcal{E}}, \hat{\mathcal{D}}$ be a global minimizer of (8) with the network classes $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(C_{\mathcal{M}}(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ where $C_{\mathcal{M}} = O((d \log d)(4/\tau)^d)$,

$$\begin{aligned} L_{\mathcal{E}} &= O(\log^2 n + \log D), \quad p_{\mathcal{E}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{E}} = O((D \log D)n^{\frac{d}{d+2}} \log^2 n), \\ \kappa_{\mathcal{E}} &= O(n^{\frac{2}{d+2}}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}, \end{aligned} \tag{19}$$

$$\begin{aligned} L_{\mathcal{D}} &= O(\log^2 n + \log D), \quad p_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D), \\ \kappa_{\mathcal{D}} &= O(n^{\frac{1}{d+2}}), \quad R_{\mathcal{D}} = B. \end{aligned} \tag{20}$$

We have

$$\mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x} \sim \gamma} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}) - \mathbf{v}\|_2^2 \leq C(D^2 \log^3 D)n^{-\frac{2}{d+2}} \log^4 n + C_1 \sigma^2 \tag{21}$$

for some constant C depending on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} , and C_1 depending on τ, q . The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

Theorem 3 is proved in Section 5.3. Theorem 3 is a straightforward extension of Theorem 2 to Setting 2 with general noise. The network architecture in Theorem 3 has a similar size as that in Theorem 2. We summarize the network architectures specified in Theorem 1, 2 and 3 in Table 2. Compared to the upper bound in Theorem 2, Theorem 3 has an additional term $C_1 \sigma^2$ which comes from the tangential component of noise. If tangential noise exists, a given point \mathbf{x} may correspond to multiple (and probably infinitely many) points on \mathcal{M} . Therefore the generalization error cannot converge to 0 as the sample size increases. This fundamental difficulty of high-dimensional noise is also demonstrated by our numerical experiments.

4. Numerical experiments

In this section, we conduct a series of experiments on simulated data to numerically verify our theoretically analysis. We consider three surfaces listed in the first column of Fig. 4. The noisy data with normal noise are displayed in the middle column of Fig. 4. We use the code in [50] to implement chart autoencoders. It is important to prescribe a reasonable number of charts to appropriately reflect the topology of the manifold. During training, chart autoencoders segment manifolds into largely non-overlapping charts. Further in some cases, chart autoencoders perform automatic chart pruning if too many charts are prescribed. This is done by contracting excess charts to a trivial or nearly trivial patch when there is already a sufficient number of charts present to capture the manifold's topology. The number of charts is prescribed to be 4 for the sphere and Genus-2 double torus and 8 for the Genus-3 pyramid. In Fig. 4, we visualize the sphere as decomposed by 2 charts, a Genus-3 pyramid as decomposed by 6 charts, and a Genus-2 double torus as decomposed by 4 charts after training. We use the network architecture such that the encoder is composed of 3 linear layers with ReLU activations, and the decoder has 3 linear layers with ReLU activations and a width (hidden dimension) of 50. In training, the batch size is 512, the learning rate is $3e-6$, and weight decay is $3e-1$.

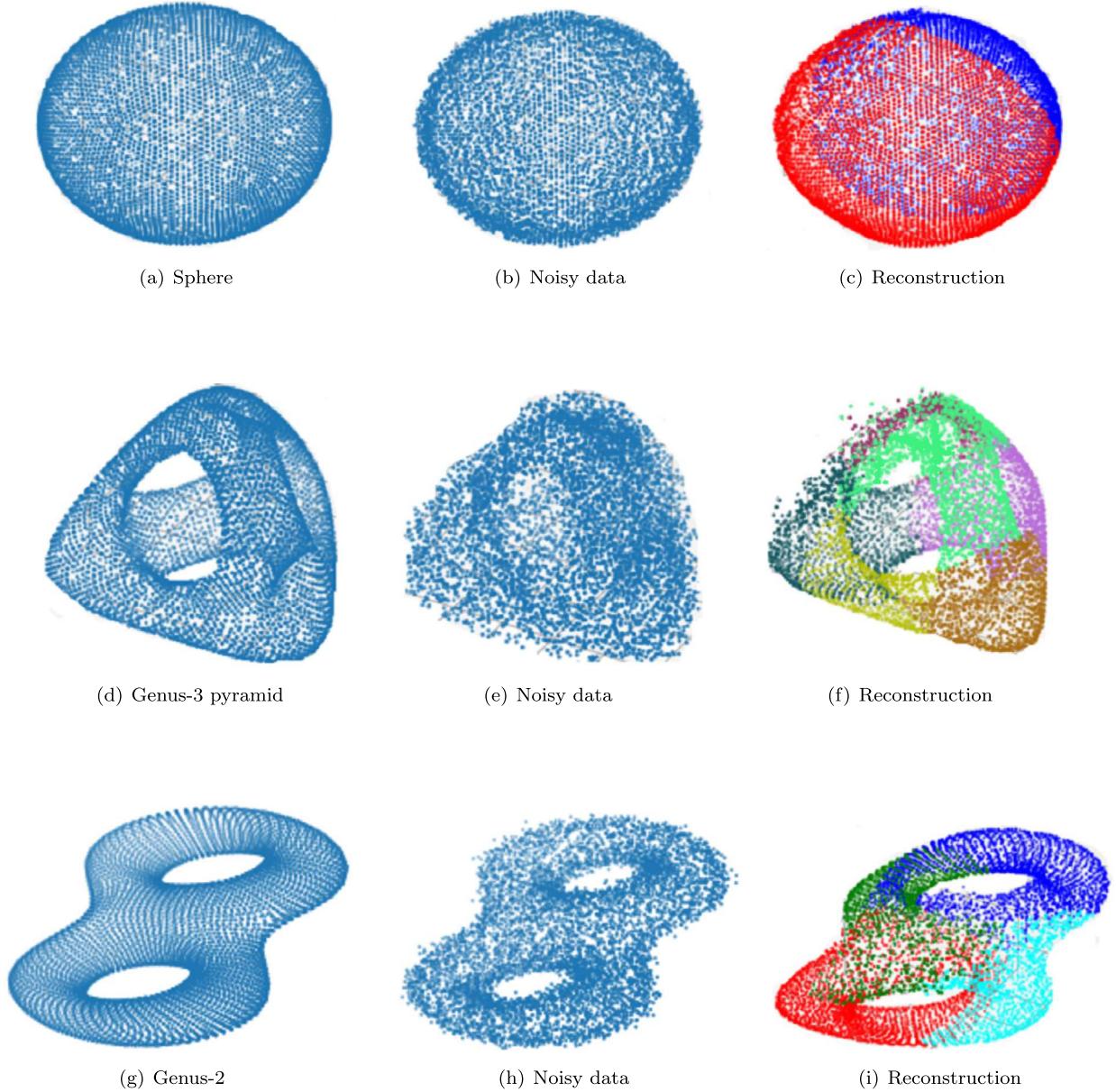


Fig. 4. Reconstructions of three noisy manifolds rescaled to lie inside the $[-50, 50]^3$ cube centered at the origin.

4.1. Sample complexity

We first investigate the sample complexity of chart autoencoders. We train chart autoencoders with n training points randomly sampled from the Genus-3 pyramid and evaluate the squared generalization error on held-out test data. The training data contain clean and noisy pairs and the test data are noisy. Let $\{(\mathbf{x}_j, \mathbf{v}_j)\}_{j=1}^{n_{\text{test}}}$ be the set of test data where \mathbf{v}_j is the clean counterpart of \mathbf{x}_j . The squared generalization error is approximated by the following squared test error:

$$\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} \|\hat{\mathcal{D}} \circ \hat{\mathcal{E}}(\mathbf{x}_j) - \mathbf{v}_j\|_2^2.$$

For each n , we perform 5 runs of experiments and average the squared test error over these 5 runs of experiments.

According to our Theorem 2, if the training data contain clean and noisy pairs with normal noise, including the noise-free case, we have

$$\text{Squared generalization error} \leq C(D^2 \log^3 D) n^{-\frac{2}{d+2} \log^4 n}. \quad (22)$$

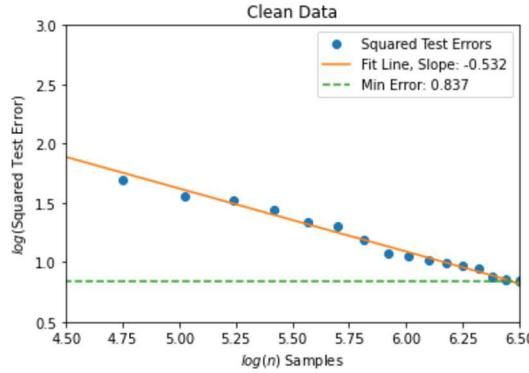


Fig. 5. Squared test error versus the training sample size n for clean data on the Genus-3 pyramid with intrinsic dimension $d = 2$ and ambient dimension $D = 3$. The horizontal line of “Min Error” represents the minimum squared testing error observed among all sample sizes.

If the training data contain clean and noisy pairs with high-dimensional noise where the second moment of the tangential component is bounded by σ^2 , Theorem 3 implies that

$$\text{Squared generalization error} \leq C_1(D^2 \log^3 D) n^{-\frac{2}{d+2} \log^4 n} + C_2 \sigma^2. \quad (23)$$

We start with the noise-free case where both training and test data are on the Genus-3 pyramid. Fig. 5 shows the log-log plot of the squared test error versus the training sample size n . Our theory in (22) implies that a least square fit of the curve has slope $-\frac{2}{d+2} = -\frac{1}{2}$ since $d = 2$. Numerically we obtain a slope of -0.532 , which is consistent with our theory. Due to the optimization error in training, we do not observe convergence to 0 in either the training or the test loss. The “Min error” is the minimum squared test error achieved among all sample sizes.

We next test the noisy case with normal noise and gaussian noise respectively. Fig. 6 displays the log-log plot of the squared test error versus the training sample size n with normal noise (left column) and gaussian noise (right column). The noise level is measured by the variance of the noise distribution. Specifically, for the normal noise \mathbf{w} in Setting 1, we set $\mathbb{E}[\mathbf{w}|\mathbf{v}] = 0$ and refer the noise level as $\mathbb{E}\|\mathbf{w}\|^2$. For the gaussian noise \mathbf{n} , we set $\mathbf{n} \sim \mathcal{N}(0, \tilde{\sigma}^2 I)$, such that $\mathbb{E}\|\mathbf{n}\|^2 = \tilde{\sigma}^2$, which is referred to be the noise level. In this experiment, we set the noise level to be 1. The Genus-3 surface is embedded in \mathbb{R}^D with $D = 3, 5, 10$ respectively. In Fig. 6, the “Min error” is the minimum squared test error achieved among all sample sizes. The “Noise Free Error” is the squared test error achieved when training on the entire clean dataset.

In the case of normal noise, we observe a convergence of the squared test error as n increases. The “Min error” is close to the “Noise Free Error”, which shows that training on noisy data almost achieves the performance of training on clean data. This demonstrates autoencoders’ denoising effect for normal noise. The slope of the line obtained from a linear fit is around -0.5 , which is consistent with our theory in (22).

In the case of gaussian noise, the squared test error first converges when n increases but then stagnates at a certain level. The “Min error” is much larger than the “Noise Free Error” which shows training on noisy data can not achieve similar performance on clean data. This is expected as our theory implies that autoencoders do not have a denoising effect for the tangential component of the noise.

4.2. Effects of the ambient dimension, the number of charts and noise levels

We next investigate how the squared generalization error of chart autoencoders depends on the ambient dimension, the number of charts, and noise levels.

Our theory in (22) shows that, when the ambient dimension D varies, the squared generalization error grows at most in $D^2 \log^3 D$. This bound may not be tight on the dependence of D . In Fig. 7 (a), we plot the squared test error for chart autoencoders with clean data on the Genus-2 and Genus-3 surfaces. We observe that, in these simulations, the squared test error almost grows linearly with respect to D . Note that the upper bounds in our theorems are for the global minimizer of the empirical loss (8). In practice, due to the complicated structure of networks, the training process may easily get stuck at a local minimizer and it is difficult to get the global minimizer. Nevertheless, our numerical results still give an approximate linear relation between the test error and D . We leave it as a future work to investigate the optimal dependence of the squared generalization error on D .

In Fig. 7 (b), we plot the squared test error for chart autoencoders with clean data on the Genus-2 and Genus-3 surfaces versus the number of charts. We observe that, when the number of charts is sufficiently large to preserve the data structure, the squared test error stays almost the same, independently of the number of charts.

In Fig. 7 (c), we plot the squared test error versus various noise levels of both normal and gaussian noise. We observe that the squared test error is much higher in the gaussian case for both manifolds.

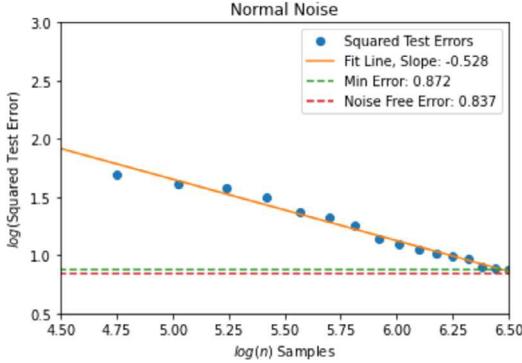
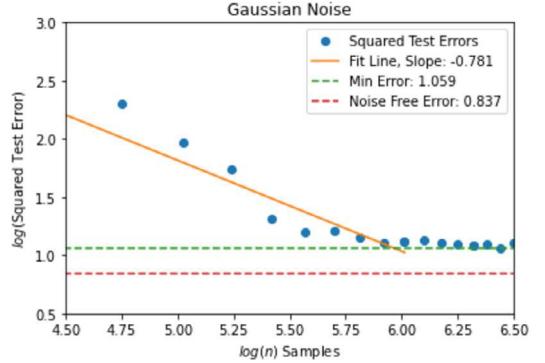
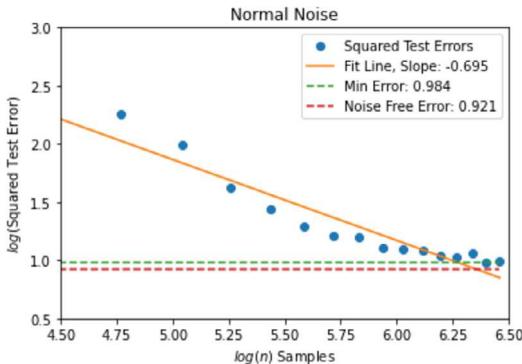
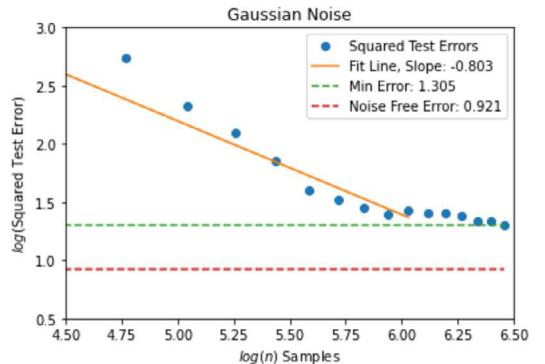
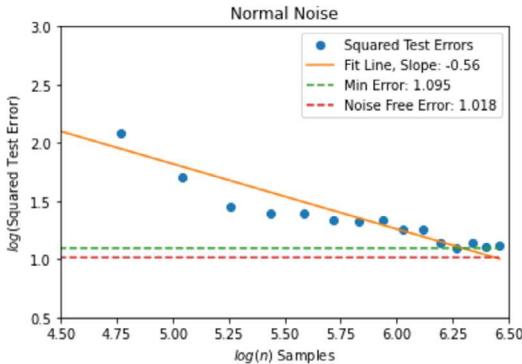
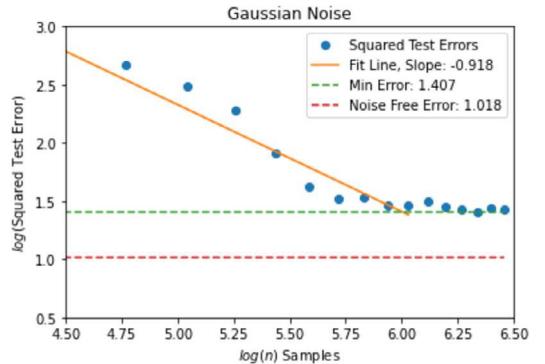
(a) Normal noise, $D = 3$ (b) Gaussian noise, $D = 3$ (c) Normal noise, $D = 5$ (d) Gaussian noise, $D = 5$ (e) Normal noise, $D = 10$ (f) Gaussian noise, $D = 10$

Fig. 6. Squared test error versus the training sample size n on the Genus-3 pyramid with intrinsic dimension $d = 2$ and ambient dimension $D \in \{3, 5, 10\}$. The left column shows the results of normal noise and the right column shows the results of Gaussian noise. The “Min error” is the minimum squared test error achieved among all sample sizes. The “Noise Free Error” is the squared test error achieved when training on the entire clean dataset.

5. Proof of main results

For the simplicity of notations, for any given $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$, we define the network class

$$\mathcal{F}_{\text{NN}}^{\mathcal{G}} = \{\mathcal{G} = \mathcal{D} \circ \mathcal{E} \mid \mathcal{D} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}, \mathcal{E} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}\} \quad (24)$$

and denote $\hat{\mathcal{G}} = \hat{\mathcal{D}} \circ \hat{\mathcal{E}}$, where $\hat{\mathcal{E}}, \hat{\mathcal{D}}$ are the global minimizers in (8).

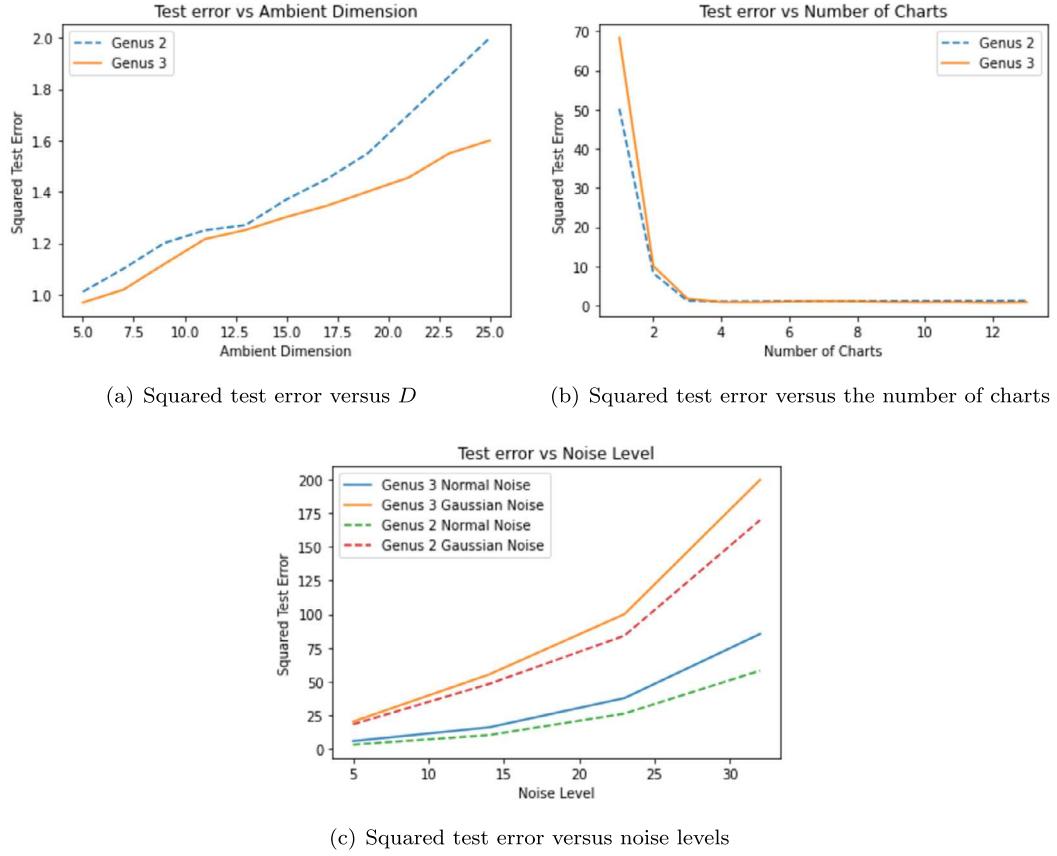


Fig. 7. Effects of the ambient dimension, the number of charts and noise levels for the Genus-3 and Genus-2 data.

5.1. Proof of Theorem 1

Proof of Theorem 1. To simplify the notation, we denote the Lipschitz constant of \mathbf{f} by C_f , i.e., for any $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{M}$,

$$\|\mathbf{f}(\mathbf{v}_1) - \mathbf{f}(\mathbf{v}_2)\|_2 \leq C_f \|\mathbf{v}_1 - \mathbf{v}_2\|_2 \leq C_f d_{\mathcal{M}}(\mathbf{v}_1, \mathbf{v}_2), \quad (25)$$

with $d_{\mathcal{M}}(\mathbf{v}_1, \mathbf{v}_2)$ denoting the geodesic distance on \mathcal{M} between \mathbf{v}_1 and \mathbf{v}_2 , and denote the Lipschitz constant of \mathbf{g} by C_g , i.e., for any $\mathbf{z}_1, \mathbf{z}_2 \in \mathbf{f}(\mathcal{M})$,

$$\|\mathbf{g}(\mathbf{z}_1) - \mathbf{g}(\mathbf{z}_2)\|_2 \leq C_g \|\mathbf{z}_1 - \mathbf{z}_2\|_2. \quad (26)$$

Our proof idea can be summarized as: We decompose the generalization error (9) into a bias term and a variance term. The bias term will be upper bounded using network approximation theory in Lemma 1. The variance will be bounded in terms of the covering number of the network class using metric entropy argument in Lemma 2.

We add and subtract two times of the empirical risk to (9) to get

$$\begin{aligned} & \mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] \\ &= \underbrace{2\mathbb{E}_{\mathcal{S}} \left[\frac{1}{n} \sum_{i=1}^n \|\hat{\mathcal{G}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2 \right]}_{T_1} + \underbrace{\mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] - 2\mathbb{E}_{\mathcal{S}} \left[\frac{1}{n} \sum_{i=1}^n \|\hat{\mathcal{G}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2 \right]}_{T_2}. \end{aligned} \quad (27)$$

The term T_1 captures the bias of the network class $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$ and T_2 captures the variance. We then derive an upper bound for each term in order.

• Bounding T_1

We derive an upper bound of T_1 using the network approximation error. We deduce

$$T_1 = 2\mathbb{E}_{\mathcal{S}} \left[\frac{1}{n} \sum_{i=1}^n \|\hat{\mathcal{G}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2 \right]$$

$$\begin{aligned}
&= 2\mathbb{E}_S \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \left[\frac{1}{n} \sum_{i=1}^n \|\mathcal{G}(\mathbf{x}_i) - \mathbf{v}_i\|_2^2 \right] \\
&\leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_S \left[\frac{1}{n} \sum_{i=1}^n \|\mathcal{G}(\mathbf{x}_i) - \pi(\mathbf{x})\|_2^2 \right] \\
&= 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2]. \tag{28}
\end{aligned}$$

The following Lemma 1 shows that by properly choosing the architecture of $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$, there exist $\tilde{\mathcal{E}} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\tilde{\mathcal{D}} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}$ so that $\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x})$ approximates $\pi(\mathbf{x})$ with high accuracy:

Lemma 1. Consider Setting 1 and suppose Assumption 1 holds. For any $0 < \varepsilon < \min\{1, \tau/2\}$, there exist two network architectures $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, d; L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(d, D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ with

$$L_{\mathcal{E}} = O(\log^2 \varepsilon^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\varepsilon^{-d}), \quad K_{\mathcal{E}} = O(D\varepsilon^{-d} \log^2 \varepsilon + D \log D), \quad \kappa_{\mathcal{E}} = O(\varepsilon^{-2}), \quad R_{\mathcal{E}} = \Lambda \tag{29}$$

and

$$L_{\mathcal{D}} = O(\log \varepsilon^{-1}), \quad p_{\mathcal{D}} = O(D\varepsilon^{-d}), \quad K_{\mathcal{D}} = O(D\varepsilon^{-d} \log \varepsilon), \quad \kappa_{\mathcal{D}} = O(\varepsilon^{-1}), \quad R_{\mathcal{D}} = B, \tag{30}$$

where the constant hidden in O depends on $d, B, \Lambda, \tau, q, C_f, C_g$ and the volume of \mathcal{M} . These network architectures give rise to $\tilde{\mathcal{E}} : \mathcal{M}(q) \rightarrow \mathbb{R}^d$ in $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\tilde{\mathcal{D}} : \mathbb{R}^d \rightarrow \mathbb{R}^D$ in $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty} \leq \varepsilon. \tag{31}$$

Lemma 1 is proved in Appendix A.1. The proof is based on the approximation theory in [15].

Let $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ be the networks in Lemma 1 with accuracy ε . We have

$$\begin{aligned}
T_1 &\leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \\
&\leq 2 \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \\
&\leq 2D \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty}^2 \\
&\leq 2D\varepsilon^2. \tag{32}
\end{aligned}$$

• **Bounding T_2 .** The term T_2 is the difference between the population risk and empirical risk of the network class $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$, except the empirical risk has a factor 2. We will derive an upper bound of T_2 using the covering number of $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$. The cover and covering number of a function class are defined as

Definition 5 (Cover). Let \mathcal{F} be a class of functions. A set of functions \mathcal{S} is a δ -cover of \mathcal{F} with respect to a norm $\|\cdot\|$ if for any $f \in \mathcal{F}$, one has

$$\inf_{u \in \mathcal{S}} \|u - f\| \leq \delta.$$

Definition 6 (Covering number, Definition 2.1.5 of [59]). Let \mathcal{F} be a class of functions. For any $\delta > 0$, the covering number of \mathcal{F} is defined as

$$\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|) = \min\{|\mathcal{S}_f| : \mathcal{S}_f \text{ is a } \delta\text{-cover of } \mathcal{F} \text{ under } \|\cdot\|\},$$

where $|\mathcal{S}_f|$ denotes the cardinality of \mathcal{S}_f .

The following lemma gives an upper bound of T_2 :

Lemma 2. Consider Setting 1. Let $0 < \delta < 1$ and T_2 be defined as in (27). We have

$$T_2 \leq \frac{35DB^2}{n} \log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \tag{33}$$

Lemma 2 is proved in Appendix A.2

• **Putting both gradients together.** Putting (32) and (33) together, we have

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \leq 2D\varepsilon^2 + \frac{35DB^2}{n} \log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \tag{34}$$

The following lemma gives an upper bound of $\mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{\infty, \infty}\right)$ in terms of the network architecture (see a proof in Appendix A.3):

Lemma 3. Let $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$ be defined in (24). The covering number of $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$ is bounded by

$$\mathcal{N}\left(\delta, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}}\right) \leq \left(\frac{2(L_{\mathcal{E}} + L_{\mathcal{D}})^2 (\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\} B + 2) (\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\})^{L_{\mathcal{E}} + L_{\mathcal{D}}} (\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\})^{L_{\mathcal{E}} + L_{\mathcal{D}} + 1}}{\delta} \right)^{2(K_{\mathcal{E}} + K_{\mathcal{D}})}. \quad (35)$$

Substituting (29) and (30) into Lemma 3, we have

$$\begin{aligned} & \log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}}\right) \\ &= O\left((L_{\mathcal{E}} + L_{\mathcal{D}})(K_{\mathcal{E}} + K_{\mathcal{D}}) (\log(L_{\mathcal{E}} + L_{\mathcal{D}}) + \log(\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\}) + \log(\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\}) + \log\delta^{-1})\right) \\ &= O\left((\log^2 \epsilon^{-1} + \log D)(D\epsilon^{-d} \log \epsilon^{-1} + D \log D) (\log \epsilon^{-1} + \log D + \log \delta^{-1})\right) \\ &\leq O\left((D \log^2 D)\epsilon^{-d} \log^4 \epsilon^{-1} + (D \log^2 D)\epsilon^{-d} \log^3 \epsilon^{-1} \log \delta^{-1}\right). \end{aligned} \quad (36)$$

Substituting (36) into (34) and setting $\delta = \epsilon = n^{-\frac{1}{d+2}}$ gives rise to

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] \leq C D^2 (\log^2 D) n^{-\frac{2}{d+2}} \log^4 n \quad (37)$$

for some constant C depending on $d, C_f, C_g, B, R_1, \tau$ and the volume of \mathcal{M} . The network sizes for $\mathcal{F}_{\text{NN}}^E$ and $\mathcal{F}_{\text{NN}}^D$ are given as

$$\begin{aligned} L_{\mathcal{E}} &= O\left(\log^2 n + \log D\right), \quad p_{\mathcal{E}} = O\left(Dn^{\frac{d}{d+2}}\right), \quad K_{\mathcal{E}} = O\left(Dn^{\frac{d}{d+2}} \log^2 n + D \log D\right), \quad \kappa_{\mathcal{E}} = O\left(n^{\frac{2}{d+2}}\right), \\ L_{\mathcal{D}} &= O(\log n), \quad p_{\mathcal{D}} = O\left(n^{\frac{d}{d+2}}\right), \quad K_{\mathcal{D}} = O\left(n^{\frac{d}{d+2}} \log^2 n\right), \quad \kappa_{\mathcal{D}} = O\left(n^{\frac{1}{d+2}}\right). \quad \square \end{aligned}$$

5.2. Proof of Theorem 2

Proof of Theorem 2. The proof of Lemma 2 is similar to that of Lemma 1, except extra efforts are needed to define the oracle encoder and decoder.

Similar to the proof of Theorem 1, we decompose the generalization error as

$$\begin{aligned} & \mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] \\ &= \underbrace{2\mathbb{E}_S \left[\frac{1}{n} \sum_{i=1}^n \|\hat{\mathcal{G}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2 \right]}_{T_1} + \underbrace{\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] - 2\mathbb{E}_S \left[\frac{1}{n} \sum_{i=1}^n \|\hat{\mathcal{G}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2 \right]}_{T_2}. \end{aligned} \quad (38)$$

• **Bounding T_1 .** We derive an upper bound of T_1 using network approximation error.

Following (28), we have

$$T_1 \leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right]. \quad (39)$$

The following Lemma shows that by properly choosing the architecture of $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$, then there exist $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ that $\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x})$ approximates $\pi(\mathbf{x})$ with high accuracy:

Lemma 4. Consider Setting 1. For any $0 < \epsilon < \min\{1, \tau/2\}$, there exist two network architectures $\mathcal{F}_{\text{NN}}^{\mathcal{E}}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ and $\mathcal{F}_{\text{NN}}^{\mathcal{D}}(C_{\mathcal{M}}(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ with

$$\begin{aligned} L_{\mathcal{E}} &= O(\log^2 \epsilon^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\epsilon^{-d}), \quad K_{\mathcal{E}} = O((D \log D)\epsilon^{-d} \log^2 \epsilon^{-1}), \\ \kappa_{\mathcal{E}} &= O(\epsilon^{-2}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}, \end{aligned} \quad (40)$$

and

$$\begin{aligned} L_{\mathcal{D}} &= O(\log^2 \epsilon^{-1} + \log D), \quad p_{\mathcal{D}} = O(D\epsilon^{-d}), \quad K_{\mathcal{D}} = O(D\epsilon^{-d} \log^2 \epsilon^{-1} + D \log D), \\ \kappa_{\mathcal{D}} &= O(\epsilon^{-1}), \quad R_{\mathcal{D}} = B. \end{aligned} \quad (41)$$

The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} . These network architectures give rise to $\tilde{\mathcal{E}} : \mathcal{M}(q) \rightarrow \mathbb{R}^d$ in $\mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\tilde{\mathcal{D}} : \mathbb{R}^d \rightarrow \mathbb{R}^D$ in $\mathcal{F}_{\text{NN}}^{\mathcal{D}}$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty} \leq \varepsilon. \quad (42)$$

Lemma 4 is proved by carefully designing an oracle encoder and decoder and showing that they can be approximated well by neural networks. The proof of Lemma 4 is presented in Appendix A.4.

Let $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ be the networks in Lemma 4 so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty} \leq \varepsilon. \quad (43)$$

We can bound T_1 as

$$\begin{aligned} T_1 &\leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \\ &\leq 2 \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \\ &\leq 2D \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty}^2 \\ &\leq 2D\varepsilon^2. \end{aligned} \quad (44)$$

• **Bounding T_2 .** By Lemma 2, we have

$$T_2 \leq \frac{35DB^2}{n} \log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \quad (45)$$

• **Putting both ingredients together.** Combining (44) and (45) gives rise to

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \leq 2D\varepsilon^2 + \frac{35DB^2}{n} \log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \quad (46)$$

The covering number can be bounded by substituting (40) and (41) into Lemma 3:

$$\begin{aligned} &\log \mathcal{N}\left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}}\right) \\ &= O\left((L_{\mathcal{E}} + L_{\mathcal{D}})(K_{\mathcal{E}} + K_{\mathcal{D}})(\log(L_{\mathcal{E}} + L_{\mathcal{D}}) + \log(\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\}) + \log(\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\}) + \log \delta^{-1})\right) \\ &= O\left((\log^2 \varepsilon^{-1} + \log D)((D \log D)\varepsilon^{-d} \log \varepsilon^{-1})(\log \varepsilon^{-1} + \log D + \log \delta^{-1})\right) \\ &= O\left((D \log^3 D)\varepsilon^{-d} \log^4 \varepsilon^{-1} + (D \log^3 D)\varepsilon^{-d} \log^3 \varepsilon^{-1} \log \delta^{-1}\right). \end{aligned} \quad (47)$$

Substituting (47) into (46) and setting $\varepsilon = \delta = n^{-\frac{1}{d+2}}$ give rise to

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \leq C(D^2 \log^3 D) n^{-\frac{2}{d+2}} \log^4 n \quad (48)$$

for some constant C depending on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

Consequently, the network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{E}}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ has

$$\begin{aligned} L_{\mathcal{E}} &= O(\log^2 n + \log D), \quad p_{\mathcal{E}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{E}} = O((D \log D)n^{\frac{d}{d+2}} \log^2 n), \\ \kappa_{\mathcal{E}} &= O(n^{\frac{2}{d+2}}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}, \end{aligned} \quad (49)$$

and the network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{D}}(C_{\mathcal{M}}(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ has

$$\begin{aligned} L_{\mathcal{D}} &= O(\log^2 n + \log D), \quad p_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D), \\ \kappa_{\mathcal{D}} &= O(n^{\frac{1}{d+2}}), \quad R_{\mathcal{D}} = B. \end{aligned} \quad (50)$$

The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} . \square

5.3. Proof of Theorem 3

Proof of Theorem 3. Theorem 3 can be proved by following the proof of Theorem 2. We decompose the generalization error as

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\hat{\mathcal{G}}(\mathbf{x}) - \mathbf{v}\|_2^2]$$

$$= 2\mathbb{E}_S \underbrace{\left[\frac{1}{n} \sum_{i=1}^n \|\widehat{\mathcal{G}}(\mathbf{x}_i) - \mathbf{v}_i\|_2^2 \right]}_{T_1} + \mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} \left[\|\widehat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right] - 2\mathbb{E}_S \underbrace{\left[\frac{1}{n} \sum_{i=1}^n \|\widehat{\mathcal{G}}(\mathbf{x}_i) - \mathbf{v}_i\|_2^2 \right]}_{T_2}. \quad (51)$$

• **Bounding T_1 .** Denote $\mathbf{w} = \text{Proj}_{T_{\mathbf{v}}^{\perp} \mathcal{M}}(\mathbf{n})$ as the component of \mathbf{n} that is normal to \mathcal{M} at \mathbf{v} , and $\mathbf{u} = \text{Proj}_{T_{\mathbf{v}} \mathcal{M}}(\mathbf{n})$ as the component that is in the tangent space of \mathcal{M} at \mathbf{v} . Using Lemma 4 and the data model in Setting 2, we have

$$\begin{aligned} T_1 &\leq 2 \inf_{\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}} \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\mathcal{G}(\mathbf{x}) - \mathbf{v}\|_2^2] \\ &\leq 2\mathbb{E}_{\mathbf{x} \sim \gamma} \left[2\|\widetilde{\mathcal{D}} \circ \widetilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 + 2\|\pi(\mathbf{x}) - \mathbf{v}\|_2^2 \right] \\ &\leq 4D\|\widetilde{\mathcal{D}} \circ \widetilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty}^2 + 4\mathbb{E}_{\mathbf{x} \sim \gamma} [\|\pi(\mathbf{x}) - \mathbf{v}\|_2^2] \\ &\leq 4D\epsilon^2 + 4\mathbb{E}_{\mathbf{x} \sim \gamma} [\|\pi(\mathbf{v} + \mathbf{w} + \mathbf{u}) - \pi(\mathbf{v} + \mathbf{w})\|_2^2] \\ &\leq 4D\epsilon^2 + 4L_{\pi} \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\mathbf{u}\|_2^2] \\ &= 4D\epsilon^2 + 4L_{\pi}^2 \mathbb{E}_{\mathbf{v}} \mathbb{E}_{\mathbf{n}} [\|\text{Proj}_{T_{\mathbf{v}} \mathcal{M}}(\mathbf{n})\|_2^2 |\mathbf{v}] \\ &\leq 4D\epsilon^2 + 4L_{\pi}^2 \sigma^2, \end{aligned} \quad (52)$$

where L_{π} denotes the Lipschitz constant of π . In (52), the fourth inequality uses Lemma 4, the fifth inequality uses the fact that $\pi(\mathbf{v} + \mathbf{w}) = \pi(\mathbf{v})$.

The following Lemma gives an upper bound of L_{π} :

Lemma 5 (Lemma 2.1 of [15]). *Let \mathcal{M} be a connected, compact, d -dimensional Riemannian manifold embedded in \mathbb{R}^D with a reach $\tau > 0$. Let π be the orthogonal projection onto \mathcal{M} . For any $q \in [0, \tau]$, we have*

$$\|\pi(\mathbf{x}_1) - \pi(\mathbf{x}_2)\|_2 \leq \frac{1}{1 - q/\tau} \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \quad (53)$$

for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{M}(q)$.

According to Lemma 5, L_{π} only depends on τ and q . Therefore we have

$$T_1 \leq 4D\epsilon^2 + C_1 \sigma^2 \quad (54)$$

for some C_1 depending on τ and q .

• **Bounding T_2 .** By Lemma 2, we have

$$T_2 \leq \frac{35DB^2}{n} \log \mathcal{N} \left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}} \right) + 6\delta. \quad (55)$$

• **Putting both ingredients together.** Combining (54) and (55) gives rise to

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\widehat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \leq 4D\epsilon^2 + \frac{35DB^2}{n} \log \mathcal{N} \left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}} \right) + 6\delta + C_1 \sigma^2. \quad (56)$$

An upper bound of the covering number is given in (47). Substituting (47) into (56) and setting $\epsilon = n^{-\frac{1}{d+2}}$, $\delta = \frac{1}{n}$ give rise to

$$\mathbb{E}_S \mathbb{E}_{\mathbf{x} \sim \gamma} [\|\widehat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2] \leq C(D^2 \log^3 D) n^{-\frac{2}{d+2}} \log^4 n + C_1 \sigma^2 \quad (57)$$

for some constant C depending on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

Consequently, the network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{G}}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{G}}, p_{\mathcal{G}}, K_{\mathcal{G}}, \kappa_{\mathcal{G}}, R_{\mathcal{G}})$ has

$$\begin{aligned} L_{\mathcal{G}} &= O(\log^2 n + \log D), \quad p_{\mathcal{G}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{G}} = O((D \log D)n^{\frac{d}{d+2}} \log^2 n), \\ \kappa_{\mathcal{G}} &= O(n^{\frac{2}{d+2}}), \quad R_{\mathcal{G}} = \max\{\tau/4, 1\}, \end{aligned} \quad (58)$$

and the network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{D}}(C_{\mathcal{M}}(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ has

$$\begin{aligned} L_{\mathcal{D}} &= O(\log^2 n + \log D), \quad p_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}}), \quad K_{\mathcal{D}} = O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D), \\ \kappa_{\mathcal{D}} &= O(n^{\frac{1}{d+2}}), \quad R_{\mathcal{D}} = B. \end{aligned} \quad (59)$$

The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} . \square

6. Conclusion

This paper studies the generalization error of Chart Auto-Encoders (CAE), when the noisy data are concentrated around a d -dimensional manifold \mathcal{M} embedded in \mathbb{R}^D . We assume that the training data are well controlled such that both the noisy data and their clean counterparts are available. When the noise is along the normal directions of \mathcal{M} , we prove that the squared generalization error converges to 0 at a fast rate in the order of $n^{-\frac{2}{2+d}} \log^4 n$. When the noise contains both normal and tangential components, we prove that the squared generalization error converges to a value proportional to the second moment of the tangential noise. Our results are supported by experimental validation. Our findings provide evidence that deep neural networks are capable of extracting low-dimensional nonlinear latent features from data, contributing to the understanding of the success of autoencoders.

Data availability

Data will be made available on request.

Appendix A. Proof of lemmas

A.1. Proof of Lemma 1

Proof of Lemma 1. We show that there exist $\tilde{\mathcal{E}} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\tilde{\mathcal{D}} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}$ that approximate $\mathbf{f} \circ \pi$ and \mathbf{g} with the given accuracy ε . The following lemma shows the existence of a network architecture with which a network $\tilde{\mathcal{E}}$ approximates $\mathbf{f} \circ \pi$ with high accuracy.

Lemma 6. Consider Setting 1 and suppose Assumption 1 holds. For any $0 < \varepsilon < \tau/2$, there exists a network architecture $\tilde{\mathbf{f}} \in \mathcal{F}(D, d; L, p, K, \kappa, R)$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathbf{f}}(\mathbf{x}) - \mathbf{f} \circ \pi(\mathbf{x})\|_{\infty} \leq \varepsilon.$$

Such a network architecture has

$$L = O(\log^2 \varepsilon^{-1} + \log D), \quad p = O(D\varepsilon^{-d}), \quad K = O(D\varepsilon^{-d} \log^2 \varepsilon + D \log D), \quad \kappa = O(\varepsilon^{-2}), \quad R = \Lambda,$$

where the constant hidden in O depends on $d, B, \Lambda, \tau, q, C_f$ and the volume of \mathcal{M} .

Lemma 6 can be proved using Cloninger and Klock [15, Theorem 2.2]. One only needs to stack d scalar-valued networks together.

To construct a network to approximate \mathbf{g} , first note that \mathbf{g} is defined on $\mathbf{f}(\mathcal{M}) \subset [-\Lambda, \Lambda]^d$. The following lemma shows that \mathbf{g} can be extended to $[-\Lambda, \Lambda]^d$ while keeping the same Lipschitz constant:

Lemma 7 (Kirschbraun theorem [28]). If $E \subset \mathbb{R}^d$, then any Lipschitz function $\mathbf{g} : E \rightarrow \mathbb{R}^D$ can be extended to the whole \mathbb{R}^d keeping the Lipschitz constant of the original function.

By Lemma 7, we extend \mathbf{g} to $[-\Lambda, \Lambda]^d$ so that the extended function is Lipschitz continuous with Lipschitz constant C_g . When there is no ambiguity, we still use \mathbf{g} to denote the extended function. The following lemma shows the existence of a network architecture with which a network $\tilde{\mathcal{D}}$ approximates \mathbf{g} on $[-\Lambda, \Lambda]^d$ with high accuracy.

Lemma 8. Consider Setting 1 and suppose Assumption 1 holds. For any $0 < \varepsilon < 1$, there exists a network architecture $\tilde{\mathbf{g}} \in \mathcal{F}(d, D; L, p, K, \kappa, R)$ so that

$$\sup_{\mathbf{z} \in [-\Lambda, \Lambda]^d} \|\tilde{\mathbf{g}}(\mathbf{z}) - \mathbf{g}(\mathbf{z})\|_{\infty} \leq \varepsilon. \quad (60)$$

Such a network architecture has

$$L = O(\log \varepsilon^{-1}), \quad p = O(D\varepsilon^{-d}), \quad K = O(D\varepsilon^{-d} \log \varepsilon), \quad \kappa = O(\varepsilon^{-1}), \quad R = B, \quad (61)$$

where the constant hidden in O depends on d, C_g, B, Λ .

Lemma 8 can be proved using [62, Theorem 1]. One only needs to stack D scalar-valued networks together.

For a constant $\varepsilon_1 \in (0, \min\{1, \tau/2\})$, we choose $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, d; L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ with

$$L_{\mathcal{E}} = O(\log^2 \varepsilon_1^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\varepsilon_1^{-d}), \quad K_{\mathcal{E}} = O(D\varepsilon_1^{-d} \log^2 \varepsilon_1 + D \log D), \quad \kappa_{\mathcal{E}} = \varepsilon_1^{-2}, \quad R_{\mathcal{E}} = \Lambda, \quad (62)$$

and $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(d, D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ with

$$L_{\mathcal{D}} = O(\log \varepsilon_1^{-1}), \quad p_{\mathcal{D}} = O(D\varepsilon_1^{-d}), \quad K_{\mathcal{D}} = O(D\varepsilon_1^{-d} \log \varepsilon_1), \quad \kappa_{\mathcal{D}} = \varepsilon_1^{-1}, \quad R_{\mathcal{D}} = B. \quad (63)$$

According to Lemma 6, there exists $\tilde{\mathcal{E}} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}$ such that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{E}}(\mathbf{x}) - \mathbf{f} \circ \pi(\mathbf{x})\|_{\infty} \leq \varepsilon_1. \quad (64)$$

According to Lemma 8, there exists $\tilde{\mathcal{D}} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}$ such that

$$\sup_{\mathbf{z} \in [-\Lambda, \Lambda]^d} \|\tilde{\mathcal{D}}(\mathbf{z}) - \mathbf{g}(\mathbf{z})\|_{\infty} \leq \varepsilon_1. \quad (65)$$

Putting (64) and (65) together and setting $\varepsilon_1 = \frac{\varepsilon}{1+C_g\sqrt{d}}$, give rise to

$$\begin{aligned} \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty} &= \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - g \circ \mathbf{f} \circ \pi(\mathbf{x})\|_{\infty} \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \left(\|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - g \circ \tilde{\mathcal{E}}(\mathbf{x})\|_{\infty} + \|g \circ \tilde{\mathcal{E}}(\mathbf{x}) - g \circ \mathbf{f} \circ \pi(\mathbf{x})\|_{\infty} \right) \\ &\leq \varepsilon_1 + C_g \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{E}}(\mathbf{x}) - \mathbf{f} \circ \pi(\mathbf{x})\|_2 \\ &\leq \varepsilon_1 + C_g \sqrt{d} \varepsilon_1 \\ &= \varepsilon. \end{aligned}$$

The lemma is proved. \square

A.2. Proof of Lemma 2

Proof of Lemma 2.

Denote

$$\hat{h}(\mathbf{x}) = \|\hat{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2.$$

We have $\hat{h}(\mathbf{x}) \leq 4DB^2$ for any $\mathbf{x} \in \mathcal{M}(q)$ due to the definition of $\hat{\mathcal{G}}$. We bound T_2 as

$$\begin{aligned} T_2 &= \mathbb{E}_S \left[\mathbb{E}_{\mathbf{x}} [\hat{h}(\mathbf{x})] - \frac{2}{n} \sum_{i=1}^n \hat{h}(\mathbf{x}_i) \right] \\ &= 2\mathbb{E}_S \left[\mathbb{E}_{\mathbf{x}} [\hat{h}(\mathbf{x})] - \frac{1}{n} \sum_{i=1}^n \hat{h}(\mathbf{x}_i) - \frac{1}{2} \mathbb{E}_{\mathbf{x}} [\hat{h}(\mathbf{x})] \right] \\ &\leq 2\mathbb{E}_S \left[\mathbb{E}_{\mathbf{x}} [\hat{h}(\mathbf{x})] - \frac{1}{n} \sum_{i=1}^n \hat{h}(\mathbf{x}_i) - \frac{1}{8DB^2} \mathbb{E}_{\mathbf{x}} [\hat{h}^2(\mathbf{x})] \right], \end{aligned} \quad (66)$$

where in the last inequality we used $\hat{h}(\mathbf{x}) \leq 4DB^2$ and

$$\frac{1}{4DB^2} \mathbb{E}_{\mathbf{x}} [\hat{h}^2(\mathbf{x})] \leq \mathbb{E}_{\mathbf{x}} [\hat{h}(\mathbf{x})].$$

Let $\tilde{\mathcal{S}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^n$ be a ghost sample set that is independent to \mathcal{S} . Define the set

$$\mathcal{H} = \{h : h(\mathbf{x}) = \|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \text{ for } \mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}\}.$$

We have

$$\begin{aligned} T_2 &\leq 2\mathbb{E}_S \sup_{h \in \mathcal{H}} \left[\mathbb{E}_{\mathbf{x}} [h(\mathbf{x})] - \frac{1}{n} \sum_{i=1}^n h(\mathbf{x}_i) - \frac{1}{8DB^2} \mathbb{E}_{\mathbf{x}} [h^2(\mathbf{x})] \right] \\ &\leq 2\mathbb{E}_{S, \tilde{\mathcal{S}}} \sup_{h \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^n (h(\tilde{\mathbf{x}}_i) - h(\mathbf{x}_i)) - \frac{1}{16DB^2} \mathbb{E}_{\mathbf{x}, \tilde{\mathbf{x}}} [h^2(\tilde{\mathbf{x}}) + h^2(\mathbf{x})] \right] \end{aligned} \quad (67)$$

Denote the δ -covering number of \mathcal{H} by $\mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_{\infty})$ and let $\mathcal{H}^* = \{h_j^*\}_{j \in \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_{\infty})}$ be a δ -cover of \mathcal{H} , namely, for any $h \in \mathcal{H}$, there exists $h^* \in \mathcal{H}^*$ so that $\|h - h^*\|_{\infty} \leq \delta$. Therefore, we have

$$\begin{aligned} h(\tilde{\mathbf{x}}) - h(\mathbf{x}) &= h(\tilde{\mathbf{x}}) - h^*(\tilde{\mathbf{x}}) + h^*(\tilde{\mathbf{x}}) - h^*(\mathbf{x}) + h^*(\mathbf{x}) - h(\mathbf{x}) \\ &\leq h^*(\tilde{\mathbf{x}}) - h^*(\mathbf{x}) + 2\delta, \end{aligned} \quad (68)$$

and

$$\begin{aligned}
h^2(\tilde{\mathbf{x}}) + h^2(\mathbf{x}) &= [h^2(\tilde{\mathbf{x}}) - (h^*)^2(\tilde{\mathbf{x}})] + [(h^*)^2(\tilde{\mathbf{x}}) + (h^*)^2(\mathbf{x})] - [(h^*)^2(\mathbf{x}) - h^2(\mathbf{x})] \\
&= (h^*)^2(\tilde{\mathbf{x}}) + (h^*)^2(\mathbf{x}) + [h(\tilde{\mathbf{x}}) - h^*(\tilde{\mathbf{x}})] [h(\tilde{\mathbf{x}}) + h^*(\tilde{\mathbf{x}})] - [h^*(\mathbf{x}) - h(\mathbf{x})] [h^*(\mathbf{x}) + h(\mathbf{x})] \\
&\geq (h^*)^2(\tilde{\mathbf{x}}) + (h^*)^2(\mathbf{x}) - |h(\tilde{\mathbf{x}}) - h^*(\tilde{\mathbf{x}})| |h(\tilde{\mathbf{x}}) + h^*(\tilde{\mathbf{x}})| - |h^*(\mathbf{x}) - h(\mathbf{x})| |h^*(\mathbf{x}) + h(\mathbf{x})| \\
&\geq (h^*)^2(\tilde{\mathbf{x}}) + (h^*)^2(\mathbf{x}) - 16DB^2\delta
\end{aligned} \tag{69}$$

Substituting (68) and (69) into (67) gives rise to

$$\begin{aligned}
T_2 &\leq 2\mathbb{E}_{S, \tilde{S}} \sup_{h^* \in \mathcal{H}^*} \left[\frac{1}{n} \sum_{i=1}^n (h^*(\tilde{\mathbf{x}}_i) - h^*(\mathbf{x}_i)) - \frac{1}{16DB^2} \mathbb{E}_{\mathbf{x}, \tilde{\mathbf{x}}} [(h^*)^2(\tilde{\mathbf{x}}) + (h^*)^2(\mathbf{x})] \right] + 6\delta \\
&= 2\mathbb{E}_{S, \tilde{S}} \max_j \left[\frac{1}{n} \sum_{i=1}^n (h_j^*(\tilde{\mathbf{x}}_i) - h_j^*(\mathbf{x}_i)) - \frac{1}{16DB^2} \mathbb{E}_{\mathbf{x}, \tilde{\mathbf{x}}} [(h_j^*)^2(\tilde{\mathbf{x}}) + (h_j^*)^2(\mathbf{x})] \right] + 6\delta
\end{aligned} \tag{70}$$

Denote $\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) = h_j^*(\tilde{\mathbf{x}}_i) - h_j^*(\mathbf{x}_i)$. We can check that $\mathbb{E}_{S, \tilde{S}}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] = 0$ for any $j = 1, \dots, \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty)$. We compute the variance of $\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)$ as

$$\text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] = \mathbb{E}_{S, \tilde{S}} [\eta_j^2(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] = \mathbb{E}_{S, \tilde{S}} \left[\left(h_j^*(\tilde{\mathbf{x}}_i) - h_j^*(\mathbf{x}_i) \right)^2 \right] \leq 2\mathbb{E}_{S, \tilde{S}} \left[\left(h_j^* \right)^2(\tilde{\mathbf{x}}_i) + \left(h_j^* \right)^2(\mathbf{x}_i) \right]. \tag{71}$$

We thus have

$$T_2 \leq \tilde{T}_2 + 6\delta \tag{72}$$

with

$$\tilde{T}_2 = 2\mathbb{E}_{S, \tilde{S}} \max_j \left[\frac{1}{n} \sum_{i=1}^n \left(\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) - \frac{1}{32DB^2} \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right]. \tag{73}$$

We next derive the moment generating function of $\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)$. For $0 < t < \frac{3}{4DB^2}$, we have

$$\begin{aligned}
\mathbb{E}_{S, \tilde{S}} [\exp(t\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i))] &= \mathbb{E}_{S, \tilde{S}} \left[1 + t\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) + \sum_{k=2}^{\infty} \frac{t^k \eta_j^k(\tilde{\mathbf{x}}_i, \mathbf{x}_i)}{k!} \right] \\
&\leq \mathbb{E}_{S, \tilde{S}} \left[1 + t\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) + \sum_{k=2}^{\infty} \frac{t^k \eta_j^2(\tilde{\mathbf{x}}_i, \mathbf{x}_i) (4DB^2)^{k-2}}{2 \times 3^{k-2}} \right] \\
&= \mathbb{E}_{S, \tilde{S}} \left[1 + t\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) + \frac{t^2 \eta_j^2(\tilde{\mathbf{x}}_i, \mathbf{x}_i)}{2} \sum_{k=2}^{\infty} \frac{t^{k-2} (4DB^2)^{k-2}}{3^{k-2}} \right] \\
&= \mathbb{E}_{S, \tilde{S}} \left[1 + t\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) + \frac{t^2 \eta_j^2(\tilde{\mathbf{x}}_i, \mathbf{x}_i)}{2} \frac{1}{1 - 4DB^2 t / 3} \right] \\
&= 1 + t^2 \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \frac{1}{2 - 8DB^2 t / 3} \\
&\leq \exp \left(\text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \frac{3t^2}{6 - 8DB^2 t} \right),
\end{aligned} \tag{74}$$

where in the first inequality we used $|\eta_j(\tilde{\mathbf{x}}, \mathbf{x})| \leq 4DB^2$.

Set $t = \tilde{t}/n$. For $0 < t < \frac{3}{4DB^2}$, we have

$$\begin{aligned}
\exp \left(\frac{\tilde{T}_2}{2} \right) &= \exp \left(\tilde{t} \mathbb{E}_{S, \tilde{S}} \max_j \left[\frac{1}{n} \sum_{i=1}^n \left(\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) - \frac{1}{32DB^2} \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right] \right) \\
&\leq \mathbb{E}_{S, \tilde{S}} \left[\exp \left(\tilde{t} \max_j \left[\frac{1}{n} \sum_{i=1}^n \left(\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) - \frac{1}{32DB^2} \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right] \right) \right] \\
&\leq \mathbb{E}_{S, \tilde{S}} \left[\sum_j \exp \left(\tilde{t} \sum_{i=1}^n \left(\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) - \frac{1}{32DB^2} \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right) \right] \\
&= \mathbb{E}_{S, \tilde{S}} \left[\sum_j \exp \left(t \sum_{i=1}^n \left(\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i) - \frac{1}{32DB^2} \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right) \right] \\
&\leq \sum_j \exp \left(\sum_{i=1}^n \left(\text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \frac{3t^2}{6 - 8DB^2 t} - \frac{1}{32DB^2} t \text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \right) \right)
\end{aligned}$$

$$= \sum_j \exp \left(t \sum_{i=1}^n \left(\text{Var}[\eta_j(\tilde{\mathbf{x}}_i, \mathbf{x}_i)] \left(\frac{3t}{6-8DB^2t} - \frac{1}{32DB^2} \right) \right) \right), \quad (75)$$

where the first inequality follows from Jensen's inequality and the third inequality uses (74). Setting

$$\frac{3t}{6-8DB^2t} - \frac{1}{32DB^2} = 0 \quad (76)$$

gives $t = \frac{3}{52DB^2} < \frac{3}{4DB^2}$ and $\tilde{t} = \frac{3n}{52DB^2}$. Substituting the value of \tilde{t} into (75) gives rise to

$$\frac{\tilde{t}\tilde{\mathbf{T}}_2}{2} \leq \log \left(\sum_j \exp(0) \right) = \log \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty), \quad (77)$$

implying that

$$\tilde{\mathbf{T}}_2 \leq \frac{2}{\tilde{t}} \log \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty) = \frac{104DB^2}{3n} \log \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty), \quad (78)$$

and

$$\mathbf{T}_2 \leq \frac{104DB^2}{3n} \log \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty) + 6\delta \leq \frac{35DB^2}{n} \log \mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty) + 6\delta. \quad (79)$$

We next derive the relation between the covering numbers of \mathcal{H} and $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$. For any $h, h' \in \mathcal{H}$, we have

$$h(\mathbf{x}) = \|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2, \text{ and } h' = \|\mathcal{G}'(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \quad (80)$$

for some $\mathcal{G}, \mathcal{G}' \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}$. We can compute

$$\begin{aligned} \|h - h'\|_\infty &= \sup_{\mathbf{x}} \left| \|\mathcal{G}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 - \|\mathcal{G}'(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \right| \\ &= \sup_{\mathbf{x}} \left| \langle \mathcal{G}(\mathbf{x}) - \mathcal{G}'(\mathbf{x}), \mathcal{G}(\mathbf{x}) + \mathcal{G}'(\mathbf{x}) - 2\pi(\mathbf{x}) \rangle \right| \\ &\leq \sup_{\mathbf{x}} \|\mathcal{G}(\mathbf{x}) - \mathcal{G}'(\mathbf{x})\|_2 \|\mathcal{G}(\mathbf{x}) + \mathcal{G}'(\mathbf{x}) - 2\pi(\mathbf{x})\|_2 \\ &\leq \sqrt{4DB} \sup_{\mathbf{x}} \sqrt{D} \|\mathcal{G}(\mathbf{x}) - \mathcal{G}'(\mathbf{x})\|_\infty \\ &= 2DB \|\mathcal{G}(\mathbf{x}) - \mathcal{G}'(\mathbf{x})\|_{L^{\infty,\infty}}. \end{aligned} \quad (81)$$

Therefore, we have

$$\mathcal{N}(\delta, \mathcal{H}, \|\cdot\|_\infty) \leq \mathcal{N} \left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}} \right) \quad (82)$$

and

$$\mathbf{T}_2 \leq \frac{35DB^2}{n} \log \mathcal{N} \left(\frac{\delta}{2DB}, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty,\infty}} \right) + 6\delta. \quad \square \quad (83)$$

A.3. Proof of Lemma 3

Proof of Lemma 3. We first show that there exists a network architecture $\mathcal{F}(D, D; L, p, K, \kappa, R)$ so that any $\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}$ can be realized by a network with such an architecture. Then the covering number of $\mathcal{F}_{\text{NN}}^{\mathcal{G}}$ can be bounded by that of $\mathcal{F}(D, D; L, p, K, \kappa, R)$.

For any $\mathcal{G} \in \mathcal{F}_{\text{NN}}^{\mathcal{G}}$, there exist $\mathcal{E} \in \mathcal{F}_{\text{NN}}^{\mathcal{E}}$ and $\mathcal{D} \in \mathcal{F}_{\text{NN}}^{\mathcal{D}}$ so that $\mathcal{G} = \mathcal{D} \circ \mathcal{E}$. Denote the set of weights and biases of \mathcal{E} by $\{(W_k^{\mathcal{E}}, \mathbf{b}_k^{\mathcal{E}})\}_{k=1}^{L_{\mathcal{E}}}$ and the set of weights and biases of \mathcal{D} by $\{(W_k^{\mathcal{D}}, \mathbf{b}_k^{\mathcal{D}})\}_{k=1}^{L_{\mathcal{D}}}$. We construct F as

$$F(\mathbf{x}) = F_3 \circ F_2 \circ F_1(\mathbf{x}) \quad (84)$$

where

$$F_1(\mathbf{x}) = \text{ReLU} \left(W_{L-1}^{\mathcal{E}} \cdots \text{ReLU}(W_1^{\mathcal{E}} \mathbf{x} + \mathbf{b}_1) + \cdots + \mathbf{b}_{L-1}^{\mathcal{E}} \right) \quad (85)$$

consists of the first $L-1$ layers of \mathcal{E} ,

$$F_3(\mathbf{t}) = W_{L_{\mathcal{D}}}^{\mathcal{D}} \cdot \text{ReLU} \left(W_{L_{\mathcal{D}}-1}^{\mathcal{D}} \cdots \text{ReLU}(W_2^{\mathcal{D}} \mathbf{t} + \mathbf{b}_2) + \cdots + \mathbf{b}_{L-1}^{\mathcal{D}} \right) + \mathbf{b}_{L_{\mathcal{D}}}^{\mathcal{D}} \quad (86)$$

consists of the 2 to $L_{\mathcal{D}}$ layers of \mathcal{D} .

Note that we have

$$G(\mathbf{x}) = F_3 \circ \text{ReLU}(W_1^{\mathcal{D}} \cdot (W_{L_{\mathcal{E}}}^{\mathcal{E}} \cdot F_1(\mathbf{x}) + \mathbf{b}_{L_{\mathcal{E}}}^{\mathcal{E}}) + \mathbf{b}_1^{\mathcal{D}}) \quad (87)$$

We will design F_2 to realize the connection between F_1 and F_3 in \mathcal{G} while keeping similar order of the number of parameters. We construct F_2 as

$$F_2(\mathbf{r}) = \text{ReLU} \left(\begin{bmatrix} W_1^{\mathcal{D}} & -W_1^{\mathcal{D}} \end{bmatrix} \cdot \text{ReLU} \left(\begin{bmatrix} W_{L_{\mathcal{E}}}^{\mathcal{E}} \\ -W_{L_{\mathcal{E}}}^{\mathcal{E}} \end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix} \mathbf{b}_{L_{\mathcal{E}}}^{\mathcal{E}} \\ -\mathbf{b}_{L_{\mathcal{E}}}^{\mathcal{E}} \end{bmatrix} \right) + \mathbf{b}_1^{\mathcal{D}} \right). \quad (88)$$

Here F_2 is a two-layer network, with width of $O(\max\{p_{\mathcal{D}}, p_{\mathcal{E}}\})$, number of nonzero parameters of $O(\max\{K_{\mathcal{D}}, K_{\mathcal{E}}\})$, and all parameters are bounded by $\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\}$. Furthermore, we have

$$F_2 \circ F_1(\mathbf{x}) = \text{ReLU}(W_1^{\mathcal{D}} \cdot (W_{L_{\mathcal{E}}}^{\mathcal{E}} \cdot F_1(\mathbf{x}) + \mathbf{b}_{L_{\mathcal{E}}}^{\mathcal{E}}) + \mathbf{b}_1^{\mathcal{D}}), \quad (89)$$

$$F_3 \circ F_2 \circ F_1(\mathbf{x}) = G(\mathbf{x}). \quad (90)$$

We next quantify the network size:

- F_1 has depth $L_{\mathcal{E}} - 1$, width $dp_{\mathcal{E}}$, number of weight parameters no more than $K_{\mathcal{E}}$, and all parameters are bounded by $\kappa_{\mathcal{E}}$.
- F_2 has depth 2, width $\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\}$, number of weight parameters is bounded by two times the number of parameters in $(W_{L_{\mathcal{E}}}^{\mathcal{E}}, \mathbf{b}_{L_{\mathcal{E}}}^{\mathcal{E}})$ and $(W_1^{\mathcal{D}}, \mathbf{b}_1^{\mathcal{D}})$, and all parameters are bounded by $\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\}$.
- F_3 has depth $L_{\mathcal{D}} - 1$, width $Dp_{\mathcal{D}}$, number of weight parameters no more than $K_{\mathcal{D}}$, and all parameters are bounded by $\kappa_{\mathcal{D}}$.

In summary, $F \in \mathcal{F}_{\text{NN}}^F = \mathcal{F}_{\text{NN}}(D, D; L, p, K, \kappa, R)$ with

$$L = L_{\mathcal{E}} + L_{\mathcal{D}}, p = \max\{p_{\mathcal{E}}, p_{\mathcal{D}}\}, K = 2(K_{\mathcal{E}} + K_{\mathcal{D}}), \kappa = \max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\}, R = B. \quad (91)$$

Therefore $\mathcal{F}_{\text{NN}}^{\mathcal{G}} \subset \mathcal{F}_{\text{NN}}^F$ and

$$\mathcal{N}(\delta, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}}) \leq \mathcal{N}(\delta, \mathcal{F}_{\text{NN}}^F, \|\cdot\|_{L^{\infty, \infty}}). \quad (92)$$

We next derive an upper bound for $\mathcal{N}(\delta, \mathcal{F}_{\text{NN}}^F, \|\cdot\|_{L^{\infty, \infty}})$. We will use the following lemma:

Lemma 9. Let $\mathcal{F}_{\text{NN}}(d_1, d_2; L, p, K, \kappa, R)$ be a class of network: $[-B, B]^{d_1} \rightarrow [-R, R]^{d_2}$. For any $\delta > 0$, the δ -covering number of $\mathcal{F}_{\text{NN}}(d_1, d_2; L, p, K, \kappa, R)$ is bounded by

$$\mathcal{N}(\delta, \mathcal{F}_{\text{NN}}(d_1, d_2; L, p, K, \kappa, R), \|\cdot\|_{L^{\infty, \infty}}) \leq \left(\frac{2L^2(pB + 2)\kappa^L p^{L+1}}{\delta} \right)^K. \quad (93)$$

Lemma 9 can be proved by following the proof of [12, Lemma 6].

By (92) and Lemma 9, we have

$$\begin{aligned} \mathcal{N}(\delta, \mathcal{F}_{\text{NN}}^{\mathcal{G}}, \|\cdot\|_{L^{\infty, \infty}}) &\leq \\ \left(\frac{2(L_{\mathcal{E}} + L_{\mathcal{D}})^2(\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\}B + 2)(\max\{\kappa_{\mathcal{E}}, \kappa_{\mathcal{D}}\})^{L_{\mathcal{E}} + L_{\mathcal{D}}}(\max\{p_{\mathcal{E}}, p_{\mathcal{D}}\})^{L_{\mathcal{E}} + L_{\mathcal{D}} + 1}}{\delta} \right)^{2(K_{\mathcal{E}} + K_{\mathcal{D}})}. \quad \square \end{aligned} \quad (94)$$

A.4. Proof of Lemma 4

Proof of Lemma 4. We first show that there exist an encoder $\mathcal{E} : \mathcal{M}(q) \rightarrow \mathbb{R}^{C_{\mathcal{M}}(d+1)}$ and a decoder $\mathcal{D} : \mathbb{R}^{C_{\mathcal{M}}(d+1)} \rightarrow \mathcal{M}$ satisfying

$$\mathcal{D} \circ \mathcal{E}(\mathbf{x}) = \pi(\mathbf{x}) \quad (95)$$

for any $\mathbf{x} \in \mathcal{M}(q)$. We call $\mathcal{E}(\mathbf{x})$ and \mathcal{D} as the oracle encoder and decoder. Then we show that there exists networks $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ approximating \mathcal{E} and \mathcal{D} so that $\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}$ approximates $\mathcal{D} \circ \mathcal{E}$ with high accuracy.

• **Constructing \mathcal{E} and \mathcal{D} .** The construction of \mathcal{E} and \mathcal{D} relies on a proper construction of an atlas of \mathcal{M} and a partition of unity of $\mathcal{M}(q)$. We construct \mathcal{E} and \mathcal{D} using the following three steps.

Step 1. In the first step, we use the results from [15] to construct a partition of unity of $\mathcal{M}(q)$.

Define the local reach [7] of \mathcal{M} at $\mathbf{v} \in \mathcal{M}$ as

$$\tau_{\mathcal{M}}(\mathbf{v}) = \inf_{\mathbf{x} \in G} \|\mathbf{x} - \mathbf{v}\|_2, \quad (96)$$

where $G = \{\mathbf{x} \in \mathbb{R}^D : \exists \text{ distinct } \mathbf{p}, \mathbf{q} \in \mathcal{M} \text{ such that } d(\mathbf{x}, \mathcal{M}) = \|\mathbf{x} - \mathbf{p}\|_2 = \|\mathbf{x} - \mathbf{q}\|_2\}$ is the medial axis of \mathcal{M} . We have

$$\tau_{\mathcal{M}} = \inf_{\mathbf{v} \in \mathcal{M}} \tau_{\mathcal{M}}(\mathbf{v}).$$

Let $\{\mathbf{v}'_j\}_{j=1}^c$ be a δ -separated set of \mathcal{M} for some integer $c > 0$. Define $p = \frac{1}{2}(1 + q/\tau)$, $h = \frac{6}{1-q/(p\tau)}$. If δ satisfies

$$\delta < C(1 - q/\tau)^2 \tau \quad (97)$$

for some absolute constant C , Cloninger and Klock [15, Proposition 6.3] constructs a partition of unity of $\mathcal{M}(q)$, denoted by $\{\eta_j\}_{j=1}^c$, defined as

$$\bar{\eta}_j(\mathbf{x}) = \max \left\{ \left(1 - \left(\frac{\|\mathbf{x} - \mathbf{v}'_j\|_2}{p\tau_{\mathcal{M}}(\mathbf{v}'_j)} \right)^2 - \left(\frac{\|A_{\mathbf{v}'_j}^{\top}(\mathbf{x} - \mathbf{v}'_j)\|_2}{h\delta} \right)^2 \right), 0 \right\}, \quad \eta_j(\mathbf{x}) = \frac{\bar{\eta}(\mathbf{x})}{\|\bar{\eta}(\mathbf{x})\|_1}, \quad (98)$$

where $\bar{\eta} = [\bar{\eta}_1, \dots, \bar{\eta}_c]^{\top}$, and $A_{\mathbf{v}'_j}$ denotes the $D \times d$ matrix containing columnwise orthonormal basis for the tangent space \mathcal{M} at \mathbf{v}'_j . In (98), j is the index of each component of this partition of unity, and the construction of $\{\eta_j\}_{j=1}^c$ only depends on the δ -separated set $\{\mathbf{v}'_j\}_{j=1}^c$ and properties of \mathcal{M} . With this construction, the cardinality of $\{\eta_j\}_{j=1}^c$ (and thus the corresponding atlas of \mathcal{M}) depends on q , which goes to infinity as q approaches τ .

Step 2. In the second step, we apply a grouping technique to $\{\eta_j\}_{j=1}^c$ to construct a partition of unity of $\mathcal{M}(q)$ and an atlas of \mathcal{M} so that their cardinality only depends on \mathcal{M} itself.

We use the following lemma:

Lemma 10. For any \mathcal{M} in Setting 1, there exists two atlases $\{\tilde{V}_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$ and $\{\bar{V}_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$ with $C_{\mathcal{M}} = O((d \log d)(4/\tau)^d)$ so that for each $j = 1, \dots, C_{\mathcal{M}}$, it holds

- (i) $\tilde{V}_j \subset \bar{V}_j$.
- (ii) $\inf_{\bar{\mathbf{v}} \in \bar{V}_j, \tilde{\mathbf{v}} \in \partial \tilde{V}_j} d_{\mathcal{M}}(\bar{\mathbf{v}}, \tilde{\mathbf{v}}) > \tau/8$, where $d_{\mathcal{M}}(\cdot, \cdot)$ denotes the geodesic distance on \mathcal{M} .
- (iii) We have

$$\|\phi_j(\mathbf{v}_1) - \phi_j(\mathbf{v}_2)\|_{\infty} \leq \|\mathbf{v}_1 - \mathbf{v}_2\|_2 \leq d_{\mathcal{M}}(\mathbf{v}_1, \mathbf{v}_2),$$

$$\|\phi_j(\mathbf{v}_1)\|_{\infty} \leq \tau/4,$$

for any $\mathbf{v}_1, \mathbf{v}_2 \in \tilde{V}_j$.

- (iv) For any $j = 1, \dots, C_{\mathcal{M}}$ and $\mathbf{z}_1, \mathbf{z}_2 \in \phi_j(\tilde{V}_j)$, we have

$$\|\phi_j^{-1}(\mathbf{z}_1) - \phi_j^{-1}(\mathbf{z}_2)\|_{\infty} \leq 2\|\mathbf{z}_1 - \mathbf{z}_2\|_2.$$

Lemma 10 is proved in Appendix A.5.

We next construct an atlas $\{V_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$ of \mathcal{M} using Lemma 10, as well as a partition of unity $\{\rho_j\}_{j=1}^{C_{\mathcal{M}}}$ defined on $\mathcal{M}(q)$ so that $\{\rho_j|_{\mathcal{M}}\}_{j=1}^{C_{\mathcal{M}}}$ is a partition of unity subordinate to $\{V_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$.

Let U_j be the support of η_j defined in (98). Then $\{U_j\}_{j=1}^c$ forms a cover of $\mathcal{M}(q)$. According to Cloninger and Klock [15, Proposition 6.3], we have

$$(U_j \cap \mathcal{M}) \subset B_{\mathcal{M}, r}(\mathbf{v}'_j) \quad \text{with} \quad r = c_1 \delta \quad (99)$$

for some constant c_1 depending on q and $\tau_{\mathcal{M}}$, where $B_{\mathcal{M}, r}(\mathbf{v})$ denotes the geodesic ball on \mathcal{M} centered at \mathbf{v} with radius r .

For $j = 1, \dots, C_{\mathcal{M}}$, we sequentially construct

$$I_j = \{k : (U_k \cap \mathcal{M}) \cap \tilde{V}_j \neq \emptyset \text{ and } k \notin I_{j'} \text{ for all } j' < j\}, \quad V_j = \bigcup_{k \in I_j} (U_k \cap \mathcal{M}).$$

Such a construction ensures that k only belongs to one I_j . According to (99) and Lemma 10(ii), as long as $2c_1\delta < \tau/8$, we have $V_j \subset \tilde{V}_j$. Therefore ϕ_j is well defined on V_j . As a result, $\{V_j, \phi_j\}_{j=1}^{C_{\mathcal{M}}}$ is an atlas of \mathcal{M} . The relation among \tilde{V}_j , \bar{V}_j , V_j and V_k 's is illustrated in Fig. 8.

Define

$$\rho_j(\mathbf{x}) = \sum_{k \in I_j} \eta_k(\mathbf{x}). \quad (100)$$

Then $\{\rho_j\}_{j=1}^{C_{\mathcal{M}}}$ is a partition of unity of $\mathcal{M}(q)$ and $\{\rho_j|_{\mathcal{M}}\}_{j=1}^{C_{\mathcal{M}}}$ is a partition of unity of \mathcal{M} subordinate to $\{V_j\}_{j=1}^{C_{\mathcal{M}}}$.

We define the oracle encoder $\mathcal{E} : \mathcal{M}(q) \rightarrow \mathbb{R}^{C_{\mathcal{M}}(d+1)}$ as

$$\mathcal{E}(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}) & \dots & \mathbf{f}_{C_{\mathcal{M}}}(\mathbf{x}) \end{bmatrix}^{\top}, \quad \text{with} \quad \mathbf{f}_j(\mathbf{x}) = [\phi_j(\pi(\mathbf{x}))^{\top} \quad \rho_j(\mathbf{x})] \in \mathbb{R}^{d+1}, \quad (101)$$

and the corresponding decoder $\mathcal{D} : \mathbb{R}^{C_{\mathcal{M}}(d+1)} \rightarrow \mathcal{M}$ as

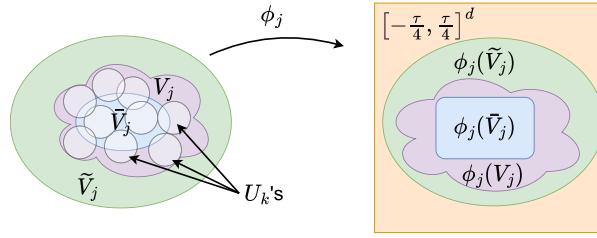


Fig. 8. Illustration of the relation among \tilde{V}_j , \tilde{V}_j , V_j and U_k 's. In the left figure, $\tilde{V}_j \subset \tilde{V}_j$ and $\partial\tilde{V}_j$ is away from $\partial\tilde{V}_j$ according to Lemma 10. For all U_k 's (white regions) that intersect \tilde{V}_j (blue region), they are inside \tilde{V}_j (green region). The set V_j (purple region) is the union of all such U_k 's that are not included in $V_{j'}$ for $j' < j$. After transformation ϕ_j , the relation among these sets is illustrated in the right figure.

$$\mathcal{D}(\mathbf{z}) = \sum_{j=1}^{C_M} \phi_j^{-1}((\mathbf{z}_j)_{1:d}) \times (\mathbf{z}_j)_{d+1},$$

where $(\mathbf{z}_j)_{1:d} = [(\mathbf{z}_j)_1 \ \dots \ (\mathbf{z}_j)_d]$. For any $\mathbf{x} \in \mathcal{M}(q)$, we can verify that

$$\begin{aligned} \mathcal{D} \circ \mathcal{E}(\mathbf{x}) &= \sum_{j=1}^{C_M} \phi_j^{-1} \circ \phi_j(\pi(\mathbf{x})) \times \rho_j(\mathbf{x}) \\ &= \sum_{\mathbf{x} \in \text{supp}(\rho_j)} \pi(\mathbf{x}) \times \rho_j(\mathbf{x}) \\ &= \pi(\mathbf{x}). \end{aligned}$$

• Constructing $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$.

The following lemma shows that there exist network $\tilde{\mathcal{E}}$ approximating \mathcal{E} and network $\tilde{\mathcal{D}}$ approximating \mathcal{D} so that $\tilde{\mathcal{G}} = \tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}$ approximate \mathcal{G} with high accuracy.

Lemma 11. Consider Setting 1. For any $0 < \epsilon < \tau/2$, there exists a network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{E}} = \mathcal{F}(D, C_M(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ giving rise to a network $\tilde{\mathcal{E}}$, and a network architecture $\mathcal{F}_{\text{NN}}^{\mathcal{D}} = \mathcal{F}(C_M(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ giving rise to a network $\tilde{\mathcal{D}}$, so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{G}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} \leq \epsilon$$

with $\tilde{\mathcal{G}} = \tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}$. The network architectures have

$$\begin{aligned} L_{\mathcal{E}} &= O(\log^2 \epsilon^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\epsilon^{-d}), \quad K_{\mathcal{E}} = O((D \log D)\epsilon^{-d} \log^2 \epsilon^{-1}), \\ \kappa_{\mathcal{E}} &= O(\epsilon^{-2}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}, \end{aligned}$$

and

$$\begin{aligned} L_{\mathcal{D}} &= O(\log^2 \epsilon^{-1} + \log D), \quad p_{\mathcal{D}} = O(D\epsilon^{-d}), \quad K_{\mathcal{D}} = O(D\epsilon^{-d} \log^2 \epsilon + D \log D), \\ \kappa_{\mathcal{D}} &= O(\epsilon^{-1}), \quad R_{\mathcal{D}} = B. \end{aligned}$$

Lemma 11 is proved in Appendix A.6.

Let $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ be defined in Lemma 11 so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{G}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} \leq \epsilon$$

For any $\mathbf{x} \in \mathcal{M}(q)$, we have

$$\|\tilde{\mathcal{G}}(\mathbf{x}) - \pi(\mathbf{x})\|_{\infty} = \|\tilde{\mathcal{G}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} \leq \epsilon. \quad \square$$

A.5. Proof of Lemma 10

Proof of Lemma 10. We construct $\{\tilde{V}_j, \phi_j\}_{j=1}^{C_M}$ and $\{\tilde{V}_j, \phi_j\}_{j=1}^{C_M}$ by covering \mathcal{M} using Euclidean balls. We first use Euclidean balls with radius $r_1 = \tau/8$ to cover \mathcal{M} . Since \mathcal{M} is compact, the number of balls is finite. Denote the number of balls by C_M and the centers by $\{\mathbf{c}_j\}_{j=1}^{C_M}$. We define

$$\tilde{V}_j = B_{r_1}(\mathbf{c}_j) \cap \mathcal{M}.$$

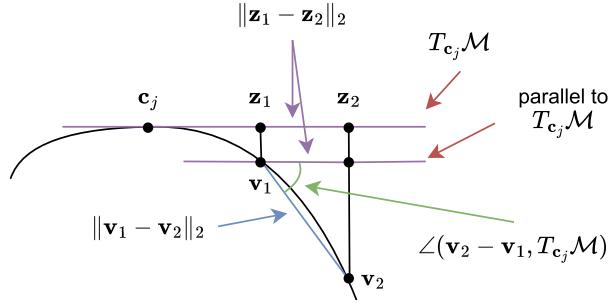


Fig. 9. Illustration of the last equality in (103).

Then $\{\bar{V}_j\}_{j=1}^{C_M}$ is a cover of \mathcal{M} . The following lemma shows that C_M is a constant depending on d and τ .

Lemma 12. Let \mathcal{M} be a d -dimensional compact Riemannian manifold embedded in \mathbb{R}^D . Assume \mathcal{M} has reach $\tau > 0$. Let $\{V_j\}_{j=1}^{C_M}$ be a minimum cover of \mathcal{M} with $V_j = \mathcal{M} \cap B_r(c_j)$ for a set of centers $\{c_j\}_{j=1}^{C_M}$. For any $r < \tau/2$, we have

$$C_M \leq \frac{|\mathcal{M}|}{\cos^d(\arcsin \frac{r}{2\tau}) |B_r^d|}, \quad (102)$$

where $|\mathcal{M}|$ denotes the volume of \mathcal{M} , and $|B_r^d|$ denotes the volume of the d -dimensional Euclidean ball with radius r .

Lemma 12 is proved in Appendix A.7. By Lemma 12, C_M is proportional to $|\mathcal{M}|/r^d$.

Let \bar{P}_j be the orthogonal projection from \bar{V}_j to the tangent plane of \mathcal{M} at c_j . By Chen et al. [12, Lemma 4.2], \bar{V}_j is diffeomorphic to a subset of \mathbb{R}^d and \bar{P}_j is a diffeomorphism.

With the same set of centers, we use Euclidean balls with radius $r_2 = \tau/4$ to cover \mathcal{M} . Define

$$\tilde{V}_j = B_{r_2}(c_j) \cap \mathcal{M}$$

and \tilde{P}_j be the orthogonal projection from \tilde{V}_j to the tangent plane of \mathcal{M} at c_j . As \tilde{V}_j is bounded and C_M is finite, there exists a constant Λ depending on τ so that $\|\tilde{P}_j(v)\|_\infty \leq \Lambda$. Setting \tilde{P}_j so that $\tilde{P}_j(c_j) = \mathbf{0}$, we have $\Lambda \leq \tau/4$. Again by Chen et al. [12, Lemma 4.2], \tilde{P}_j is a diffeomorphism between \tilde{V}_j and \mathbb{R}^d . We set $\phi_j = \tilde{P}_j$. Since $\bar{V}_i \subset \tilde{V}_i$, ϕ_i is well defined on \bar{V}_i . We have

$$\inf_{\bar{v} \in \bar{V}_i, \tilde{v} \in \partial \tilde{V}_i} d_{\mathcal{M}}(\bar{v}, \tilde{v}) \geq \inf_{\bar{v} \in \bar{V}_i, \tilde{v} \in \partial \tilde{V}_i} \|\bar{v} - \tilde{v}\|_2 \geq r_2 - r_1 = \tau/8.$$

Since ϕ_j 's are diffeomorphisms, there exist constants C_f and C_g so that item (iii) and (iv) hold.

We next prove the Lipschitz property of ϕ_j and ϕ_j^{-1} . For ϕ_j and any $v_1, v_2 \in \tilde{V}_j$, we have

$$\|\phi_j(v_1) - \phi_j(v_2)\|_\infty = \|\tilde{P}_j(v_1 - v_2)\|_\infty \leq \|\tilde{P}_j(v_1 - v_2)\|_2 \leq \|v_1 - v_2\|_2 \leq d_{\mathcal{M}}(v_1, v_2).$$

We then focus on ϕ_j^{-1} . Denote the tangent space of \mathcal{M} at $v \in \mathcal{M}$ by $T_v \mathcal{M}$, and the principal angle between two tangent spaces $T_{v_1} \mathcal{M}, T_{v_2} \mathcal{M}$ by $\angle(T_{v_1} \mathcal{M}, T_{v_2} \mathcal{M})$. We will use the following lemma

Lemma 13 (Corollary 3 in [8]). Let \mathcal{M} be a d -dimensional manifold embedded in \mathbb{R}^D . Denote the reach of \mathcal{M} by τ . We have

$$\sin \frac{\angle(T_{v_1} \mathcal{M}, T_{v_2} \mathcal{M})}{2} \leq \frac{\|v_1 - v_2\|_2}{2\tau}$$

for any $v_1, v_2 \in \mathcal{M}$.

We have

$$\|\phi_j^{-1}(z_1) - \phi_j^{-1}(z_2)\|_\infty \leq \|\phi_j^{-1}(z_1) - \phi_j^{-1}(z_2)\|_2 = \|v_1 - v_2\|_2 = \frac{\|z_1 - z_2\|_2}{|\cos(\angle(v_1 - v_2, T_{c_j} \mathcal{M}))|}, \quad (103)$$

where the last equality is due to ϕ_j being an orthogonal projection and some geometric derivation, see Fig. 9 for an illustration.

The denominator in (103) can be lower bounded as

$$\begin{aligned} |\cos(\angle(v_1 - v_2, T_{c_j} \mathcal{M}))| &= \sqrt{1 - \sin^2(\angle(v_1 - v_2, T_{c_j} \mathcal{M}))} \\ &\geq \sqrt{1 - \max_{v \in \tilde{V}_j} \sin^2(\angle(T_v \mathcal{M}, T_{c_j} \mathcal{M}))} \end{aligned}$$

$$\begin{aligned}
&\geq \sqrt{1 - \max_{\mathbf{v} \in \tilde{V}_j} 4 \sin^2 \left(\frac{\angle(T_{\mathbf{v}} \mathcal{M}, T_{\mathbf{c}_j} \mathcal{M})}{2} \right)} \\
&\geq \sqrt{1 - \max_{\mathbf{v} \in \tilde{V}_j} 4 \left(\frac{\|\mathbf{v} - \mathbf{c}_j\|_2}{2\tau} \right)^2} \\
&= \sqrt{1 - 4 \left(\frac{\tau}{4 \cdot 2\tau} \right)^2} \\
&= \frac{\sqrt{15}}{4} \\
&\geq 1/2,
\end{aligned} \tag{104}$$

where in the third inequality Lemma 13 is used.

Substituting (104) into (103) gives rise to

$$\|\phi_j^{-1}(\mathbf{z}_1) - \phi_j^{-1}(\mathbf{z}_2)\|_\infty \leq 2\|\mathbf{z}_1 - \mathbf{z}_2\|_2. \quad \square$$

A.6. Proof of Lemma 11

Proof of Lemma 11. We will construct networks $\tilde{\mathcal{E}}$ and $\tilde{\mathcal{D}}$ to approximate \mathcal{E} and \mathcal{D} , respectively.

• Construction of $\tilde{\mathcal{E}}$.

The encoder \mathcal{E} is a collection of \mathbf{f}_j 's, which consist of ϕ_j, π and ρ_j . We show that these functions can be approximated well by networks. By Lemma 10(iii), ϕ_j is Lipschitz continuous with Lipschitz constant 1. According to Lemma 6, for any $0 < \varepsilon < 1$, there exists a network architecture $\mathcal{F}(D, d; L_1, p_1, K_1, \kappa_1, R_1)$ that gives rise to a network $\tilde{\phi}_j$ satisfying

$$\sup_{\mathbf{x} \in V_j} \|\tilde{\phi}_j(\mathbf{x}) - \phi_j \circ \pi(\mathbf{x})\|_\infty \leq \varepsilon_1. \tag{105}$$

Such an architecture has

$$L_1 = O(\log^2 \varepsilon_1^{-1} + \log D), \quad p_1 = O(D\varepsilon_1^{-d}), \quad K_1 = O(D\varepsilon_1^{-d} \log^2 \varepsilon_1 + D \log D), \quad \kappa_1 = O(\varepsilon_1^{-2}), \quad R_1 = \tau/4.$$

The following lemma shows that ρ_j can be approximated by a network with arbitrary accuracy (see a proof in Appendix A.8):

Lemma 14. Consider Setting 1. For any $0 < \varepsilon < 1$ and $j = 1, \dots, C_{\mathcal{M}}$, there exists a network architecture $\mathcal{F}(D, 1; L, p, K, \kappa, 1)$ giving rise to a network $\tilde{\rho}_j$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} |\tilde{\rho}_j(\mathbf{x}) - \rho_j(\mathbf{x})| \leq \varepsilon.$$

Such a network architecture has

$$L = O(\log^2 \varepsilon^{-1} + \log D), \quad p = O(D\varepsilon^{-1}), \quad K = O((D \log D)\varepsilon^{-1} \log^2 \varepsilon^{-1}), \quad \kappa = O(\varepsilon^{-2}).$$

By Lemma 14, there exists a network $\tilde{\rho}_j \in \mathcal{F}(D, 1; L_2, p_2, K_2, \kappa_2, R_2)$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} |\tilde{\rho}_j(\mathbf{x}) - \rho_j(\mathbf{x})| \leq \varepsilon_1. \tag{106}$$

Such a network architecture has

$$L_2 = O(\log^2 \varepsilon_1^{-1} + \log D), \quad p_2 = O(D\varepsilon_1^{-1}), \quad K_2 = O((D \log D)\varepsilon_1^{-1} \log^2 \varepsilon_2^{-1}), \quad \kappa_2 = O(\varepsilon_1^{-2}), \quad R_2 = 1.$$

Define

$$\tilde{\mathbf{f}}_j(\mathbf{x}) = [(\tilde{\phi}_j(\mathbf{x}))^\top \quad \tilde{\rho}_j(\mathbf{x})]^\top \quad \text{and} \quad \tilde{\mathcal{E}}(\mathbf{x}) = [\tilde{\mathbf{f}}_1(\mathbf{x}) \quad \dots \quad \tilde{\mathbf{f}}_{C_{\mathcal{M}}}(\mathbf{x})]^\top. \tag{107}$$

We have $\tilde{\mathcal{E}} \in \mathcal{F}^{\mathcal{E}}(D, C_{\mathcal{M}}(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ with

$$\begin{aligned}
L_{\mathcal{E}} &= O(\log^2 \varepsilon_1^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\varepsilon_1^{-1}), \quad K_{\mathcal{E}} = O((D \log D)\varepsilon_1^{-d} \log^2 \varepsilon_1^{-1}), \\
\kappa_{\mathcal{E}} &= O(\varepsilon_1^{-2}), \quad R_{\mathcal{E}} = \max\{\tau/4, 1\}.
\end{aligned}$$

The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

• Construction of $\tilde{\mathcal{D}}$.

By Lemma 10(iv), ϕ_j^{-1} is Lipschitz continuous with Lipschitz constant 2. Although ϕ_j^{-1} is only defined on $\phi_j(\tilde{V}_j) \subset [-\tau/4, \tau/4]^d$, we can extend it to $[-\tau/4, \tau/4]^d$ by Lemma 7. Such an extension preserves the property Lemma 10(iv).

For any $0 < \varepsilon_2 < 1$, by Lemma 8, there exists a network $\bar{g}_j \in \mathcal{F}(d, D; L_3, p_3, K_3, \kappa_3, R_3)$ so that

$$\sup_{\mathbf{z} \in [-\tau/4, \tau/4]^d} \|\bar{g}_j(\mathbf{z}) - \phi_j^{-1}(\mathbf{z})\|_\infty \leq \varepsilon_2.$$

Such a network architecture has

$$L_3 = O(\log^2 \varepsilon_2^{-1}) + \log D, \quad p_3 = O(D\varepsilon_2^{-d}), \quad K_3 = O(D\varepsilon_2^{-d} \log^2 \varepsilon_2 + D \log D), \quad \kappa_3 = \varepsilon_2^{-1}, \quad R_3 = B.$$

Note that the input of \bar{g}_j is in \mathbb{R}^d while the input to the decoder is in $\mathbb{R}^{C_M(d+1)}$. We will append \bar{g}_j by a linear transformation that extract the corresponding elements (i.e., the first d element in the output of $\tilde{\mathbf{f}}_j$) from the input of the decoder. Define the matrix $A_j \in \mathbb{R}^{d \times (C_M(d+1))}$ with

$$(A_j)_{i,k} = \begin{cases} 1 & \text{if } k = (j-1)(d+1) + i, \\ 0 & \text{otherwise.} \end{cases}$$

We define the network

$$\tilde{g}_j(\mathbf{z}) = \bar{g}_j(A_j \mathbf{z})$$

for any $\mathbf{z} \in \mathbb{R}^{C_M(d+1)}$.

The following lemma shows that the multiplication operator \times can be well approximated by networks:

Lemma 15 (Proposition 3 in [62]). *For any $B > 0$ and $0 < \varepsilon < 1$, there exists a network $\tilde{\times}$ so that for any $|x_1| \leq B$ and $|x_2| \leq B$, we have*

$$|\tilde{\times}(x_1, x_2) - x_1 \times x_2| < \varepsilon, \quad \tilde{\times}(x_1, 0) = \tilde{\times}(0, x_2) = 0.$$

Such a network has $O(\log \varepsilon^{-1})$ layers and parameters. The width is bounded by 6 and all parameters are bounded by B^2 .

Based on Lemma 15, it is easy to derive the following lemma:

Lemma 16. *For any $B > 0$ and $0 < \varepsilon < 1$, there exists a network $\tilde{\times}$ so that for any $\mathbf{x} \in \mathbb{R}^D$ with $\|\mathbf{x}\|_\infty \leq B$ and $|y| \leq B$, we have*

$$\|\tilde{\times}(\mathbf{x}, y) - y\mathbf{x}\|_\infty < \varepsilon$$

and $(\tilde{\times}(\mathbf{x}, y))_i = 0$ if either $x_i = 0$ or $y = 0$, where x_i denotes the i -th element of \mathbf{x} . Such a network has $O(\log \varepsilon^{-1})$ layers, $O(D \log \varepsilon^{-1})$ parameters. The width is bounded by $6D$ and all parameters are bounded by B^2 .

Lemma 16 can be proved by stacking D networks defined in Lemma 15. The proof is omitted here.

Let $\tilde{\times}$ be the network defined in Lemma 16 with accuracy ε_2 . We construct the decoder $\tilde{\mathcal{D}}$ as

$$\tilde{\mathcal{D}}(\mathbf{z}) = \sum_{j=1}^{C_M} \tilde{\times}(\tilde{g}_j(\mathbf{z}), \bar{A}_j \mathbf{z}),$$

where $\bar{A}_j \in \mathbb{R}^{1 \times C_M(d+1)}$ is a weight matrix defined by

$$(\bar{A}_j)_{1,k} = \begin{cases} 1 & \text{if } k = j(d+1), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\Omega_{\mathbf{z}} = ([-\tau/4, \tau/4] \times ([0, 1]))^{C_M}$. Setting $\varepsilon_2 = \frac{\varepsilon_3}{2C_M}$, we deduce that

$$\begin{aligned} \sup_{\mathbf{z} \in \Omega_{\mathbf{z}}} \|\tilde{\mathcal{D}}(\mathbf{z}) - \mathcal{D}(\mathbf{z})\|_\infty &\leq \sup_{\mathbf{z} \in \Omega_{\mathbf{z}}} \sum_{j=1}^{C_M} \left\| \tilde{\times}(\tilde{g}_j(\mathbf{z}), \bar{A}_j \mathbf{z}) - \phi_j^{-1}((\mathbf{z}_j)_{1:d}) \times (\mathbf{z}_j)_{d+1} \right\|_\infty \\ &= \sup_{\mathbf{z} \in \Omega_{\mathbf{z}}} \sum_{j=1}^{C_M} \left\| \tilde{\times}(\tilde{g}_j(\mathbf{z}), \bar{A}_j \mathbf{z}) - \phi_j^{-1}(A_j \mathbf{z}) \times (\bar{A}_j \mathbf{z}) \right\|_\infty \\ &\leq \sup_{\mathbf{z} \in \Omega_{\mathbf{z}}} \sum_{j=1}^{C_M} \left(\left\| \tilde{\times}(\tilde{g}_j(\mathbf{z}), \bar{A}_j \mathbf{z}) - \tilde{g}_j(\mathbf{z}) \times (\bar{A}_j \mathbf{z}) \right\|_\infty \right. \\ &\quad \left. + \left\| \tilde{g}_j(\mathbf{z}) \times (\bar{A}_j \mathbf{z}) - \phi_j^{-1}(A_j \mathbf{z}) \times (\bar{A}_j \mathbf{z}) \right\|_\infty \right) \\ &\leq \sum_{j=1}^{C_M} (\varepsilon_2 + \varepsilon_2) \\ &= 2C_M \varepsilon_2 \end{aligned}$$

$$= \varepsilon_3. \quad (108)$$

• **Error estimation of $\tilde{\mathcal{G}}$.** We have

$$\begin{aligned} \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{G}}(\mathbf{x}) - G(\mathbf{x})\|_{\infty} &= \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{D}} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \mathcal{D} \circ \tilde{\mathcal{E}}(\mathbf{x})\|_{\infty} + \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\mathcal{D} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} \\ &\leq \varepsilon_3 + \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\mathcal{D} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty}. \end{aligned} \quad (109)$$

We next derive an upper bound of the second term in (109). Recall the definition of \mathbf{f}_j and $\tilde{\mathbf{f}}_j$ in (101) and (107), respectively. Plugin the expression of \mathcal{D} into the second term in (109), we have

$$\begin{aligned} \sup_{\mathbf{x} \in \mathcal{M}(q)} \|\mathcal{D} \circ \tilde{\mathcal{E}}(\mathbf{x}) - \mathcal{D} \circ \mathcal{E}(\mathbf{x})\|_{\infty} &= \sup_{\mathbf{x} \in \mathcal{M}(q)} \left\| \sum_{j=1}^{C_M} \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \times (\tilde{\mathbf{f}}_j)_{d+1} - \sum_{j=1}^{C_M} \phi_j^{-1}((\mathbf{f}_j)_{1:d}) \times (\mathbf{f}_j)_{d+1} \right\|_{\infty} \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{j=1}^{C_M} \left\| \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \times (\tilde{\mathbf{f}}_j)_{d+1} - \phi_j^{-1}((\mathbf{f}_j)_{1:d}) \times (\mathbf{f}_j)_{d+1} \right\|_{\infty} \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{j=1}^{C_M} \left(\left\| \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \times (\tilde{\mathbf{f}}_j)_{d+1} - \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \times (\mathbf{f}_j)_{d+1} \right\|_{\infty} \right. \\ &\quad \left. + \left\| \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \times (\mathbf{f}_j)_{d+1} - \phi_j^{-1}((\mathbf{f}_j)_{1:d}) \times (\mathbf{f}_j)_{d+1} \right\|_{\infty} \right) \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{j=1}^{C_M} \left(\left\| \phi_j^{-1}((\tilde{\mathbf{f}}_j)_{1:d}) \right\|_{\infty} \left\| (\tilde{\mathbf{f}}_j)_{d+1} - (\mathbf{f}_j)_{d+1} \right\|_{\infty} \right. \\ &\quad \left. + 2\|(\mathbf{f}_j)_{d+1}\|_{\infty} \|(\tilde{\mathbf{f}}_j)_{1:d} - (\mathbf{f}_j)_{1:d}\|_2 \right) \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{j=1}^{C_M} \left(\frac{\tau}{4} \|(\tilde{\mathbf{f}}_j)_{d+1} - (\mathbf{f}_j)_{d+1}\|_{\infty} + 2\|(\tilde{\mathbf{f}}_j)_{1:d} - (\mathbf{f}_j)_{1:d}\|_2 \right) \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{j=1}^{C_M} \left(\frac{\tau}{4} \|\tilde{\rho}_j(\mathbf{x}) - \rho_j(\mathbf{x})\|_{\infty} + 2\|\tilde{\phi}_j(\mathbf{x}) - \phi_j \circ \pi(\mathbf{x})\|_2 \right) \\ &\leq \sum_{j=1}^{C_M} \left(\frac{\tau}{4} \varepsilon_1 + 2\sqrt{d} \varepsilon_1 \right) \\ &\leq C_M \left(\frac{\tau}{4} + 2\sqrt{d} \right) \varepsilon_1. \end{aligned} \quad (110)$$

In the above, we used ϕ_j^{-1} is Lipschitz continuous in the third inequality, ϕ_j^{-1} is bounded by $\frac{\tau}{4}$ and $|\rho_j| \leq 1$ in the fourth inequality, the definition of $\tilde{\mathbf{f}}_j$, \mathbf{f}_j ((107) and (101), respectively) in the fifth inequality, and (105) and (106) in the sixth inequality.

Substituting (110) into (109) and setting $\varepsilon_1 = \frac{\varepsilon}{2C_M(\frac{\tau}{4} + 2\sqrt{d})}$ and $\varepsilon_3 = \varepsilon/2$ give rise to

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\mathcal{G}}(\mathbf{x}) - G(\mathbf{x})\|_{\infty} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

• **Network architectures.** For $\tilde{\mathcal{E}}$, we have $\tilde{\mathcal{E}} \in \mathcal{F}^{\tilde{\mathcal{E}}}(D, C_M(d+1); L_{\mathcal{E}}, p_{\mathcal{E}}, K_{\mathcal{E}}, \kappa_{\mathcal{E}}, R_{\mathcal{E}})$ with

$$L_{\mathcal{E}} = O(\log^2 \varepsilon^{-1} + \log D), \quad p_{\mathcal{E}} = O(D\varepsilon^{-d}), \quad K_{\mathcal{E}} = O((D \log D)\varepsilon^{-d} \log^2 \varepsilon^{-1}),$$

$$\kappa_{\mathcal{E}} = O(\varepsilon^{-2}), \quad R_{\mathcal{E}} = \tau/4.$$

The constant hidden in O depends on d, τ, q, B, C_M and the volume of \mathcal{M} .

For $\tilde{\mathcal{D}}$, it consists of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{g}}_j$:

- $\tilde{\mathbf{x}}$: It has depth of $O(\log \varepsilon^{-1})$, width bounded by $6D$ and $O(D \log \varepsilon^{-1})$ parameters. All parameters are bounded by B^2 .
- $\tilde{\mathbf{g}}_j$: It has depth $O(\log^2 \varepsilon^{-1}) + \log D$, width of $O(D\varepsilon^{-d})$ and $O(D\varepsilon^{-d} \log^2 \varepsilon + D \log D)$. All weight parameters are bounded by ε^{-2} .

Therefore, we have $\tilde{\mathcal{D}} \in \mathcal{F}^{\tilde{\mathcal{D}}}(C_M(d+1), D; L_{\mathcal{D}}, p_{\mathcal{D}}, K_{\mathcal{D}}, \kappa_{\mathcal{D}}, R_{\mathcal{D}})$ with

$$L_{\mathcal{D}} = O(\log^2 \varepsilon^{-1} + \log D), \quad p_{\mathcal{D}} = O(D\varepsilon^{-d}), \quad K_{\mathcal{D}} = O(D\varepsilon^{-d} \log^2 \varepsilon + D \log D),$$

$$\kappa_{\mathcal{D}} = O(\varepsilon^{-1}), \quad R_{\mathcal{D}} = B.$$

The constant hidden in O depends on $d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} . \square

A.7. Proof of Lemma 12

Proof of Lemma 12. Denote the packing number of \mathcal{M} using sets in the form of $\mathcal{M} \cap B_r(\mathbf{c})$ for $\mathbf{c} \in \mathcal{M}$ by $P_{\mathcal{M}}(r)$. By Niyogi et al. [42, Lemma 5.3], we have

$$P_{\mathcal{M}}(r) \leq \frac{|\mathcal{M}|}{\cos^d(\arcsin \frac{r}{2\tau}) |B_r^d|}. \quad (111)$$

According to Niyogi et al. [42, Lemma 5.2], we get

$$C_{\mathcal{M}} \leq P_{\mathcal{M}}(r/2) \leq \frac{|\mathcal{M}|}{\cos^d(\arcsin \frac{r}{4\tau}) |B_r^d|} = O(r^{-d}). \quad (112)$$

A.8. Proof of Lemma 14

Proof of Lemma 14. Define $\eta(\mathbf{x}) = [\eta_1(\mathbf{x}) \quad \dots \quad \eta_c(\mathbf{x})]^\top$. By Cloninger and Klock [15, Lemma 15], for any $0 < \varepsilon < 1$, there exists a network architecture $\mathcal{F}(D, c; L_1, p_1, K_1, \kappa_1, 1)$ giving rise to a network $\tilde{\eta}$ so that

$$\sup_{\mathbf{x} \in \mathcal{M}(q)} \|\tilde{\eta}(\mathbf{x}) - \eta(\mathbf{x})\|_1 \leq \varepsilon.$$

Such a network has

$$L_1 = O(\log^2 \varepsilon^{-1} + \log D), \quad p_1 = O(D\varepsilon^{-1}), \quad K_1 = O((D \log D)\varepsilon^{-1} \log^2 \varepsilon^{-1}), \quad \kappa_1 = O(\varepsilon^{-2}).$$

The constant hidden in O depends on $c, d, \tau, q, B, C_{\mathcal{M}}$ and the volume of \mathcal{M} .

Note that ρ_j is the sum of the elements in the output of η that have index in I_j . We construct $\tilde{\rho}_j$ by appending $\tilde{\eta}$ by a layer summing up the corresponding elements. Let $\mathbf{w} \in \mathbb{R}^c$ so that $\mathbf{w}_k = 1$ if $k \in I_j$, and $\mathbf{w}_k = 0$ otherwise. The definition of \mathbf{w} is an indicator for the k 's in I_j . For different j 's, we have different I_j 's and thus different \mathbf{w} 's. We construct $\tilde{\rho}_j$ as

$$\tilde{\rho}_j(\mathbf{x}) = \mathbf{w} \cdot \text{ReLU}(\tilde{\eta}(\mathbf{x})).$$

We have $\tilde{\rho}_j \in \mathcal{F}(1, L, p, K, \kappa, 1)$ for $\mathcal{F}(1, L, p, K, \kappa, 1)$ defined in Lemma 14 and

$$\begin{aligned} \sup_{\mathbf{x} \in \mathcal{M}(q)} |\tilde{\rho}_j - \rho_j| &= \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{k \in I_j} |\tilde{\eta}_k - \eta_k| \\ &\leq \sup_{\mathbf{x} \in \mathcal{M}(q)} \sum_{k=1}^c |\tilde{\eta}_k - \eta_k| \\ &\leq \varepsilon. \quad \square \end{aligned}$$

References

- [1] E. Aamari, C. Levraud, Nonasymptotic rates for manifold, tangent space and curvature estimation, *Ann. Stat.* 47 (2019) 177–204.
- [2] W.K. Allard, G. Chen, M. Maggioni, Multi-scale geometric methods for data sets II: geometric multi-resolution analysis, *Appl. Comput. Harmon. Anal.* 32 (2012) 435–462.
- [3] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* 15 (2003) 1373–1396.
- [4] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, *Adv. Neural Inf. Process. Syst.* 19 (2006).
- [5] Y. Bengio, et al., Learning deep architectures for AI, *Found. Trends Mach. Learn.* 2 (2009) 1–127.
- [6] J. Biehler, M.W. Gee, W.A. Wall, Towards efficient uncertainty quantification in complex and large-scale biomechanical problems based on a Bayesian multi-fidelity scheme, *Biomech. Model. Mechanobiol.* 14 (2015) 489–513.
- [7] J.-D. Boissonnat, A. Ghosh, Manifold reconstruction using tangential Delaunay complexes, in: *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, 2010.
- [8] J.-D. Boissonnat, A. Lieutier, M. Wintraecken, The reach, metric distortion, geodesic convexity and the variation of tangent spaces, *J. Appl. Comput. Topol.* 3 (2019) 29–58.
- [9] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, *Biol. Cybern.* 59 (1988) 291–294.
- [10] G. Canas, T. Poggio, L. Rosasco, Learning manifolds with K-means and K-flats, *Adv. Neural Inf. Process. Syst.* 25 (2012).
- [11] M. Chen, H. Jiang, W. Liao, T. Zhao, Efficient approximation of deep ReLU networks for functions on low dimensional manifolds, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [12] M. Chen, H. Jiang, W. Liao, T. Zhao, Nonparametric regression on low-dimensional manifolds using deep ReLU networks: function approximation and statistical recovery, *arXiv preprint arXiv:1908.01842*, 2019.
- [13] M. Chen, H. Liu, W. Liao, T. Zhao, Doubly robust off-policy learning on low-dimensional manifolds by deep neural networks, *arXiv preprint arXiv:2011.01797*, 2020.
- [14] C.K. Chui, H.N. Mhaskar, Deep nets for local manifold learning, *Front. Appl. Math. Stat.* 4 (2018) 12.
- [15] A. Cloninger, T. Klock, A deep network construction that adapts to intrinsic dimensionality beyond the domain, *Neural Netw.* 141 (2021) 404–419.

- [16] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, *Proc. Natl. Acad. Sci.* 102 (2005) 7426–7431.
- [17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009.
- [18] H. Federer, Curvature measures, *Trans. Am. Math. Soc.* 93 (1959) 418–491.
- [19] C. Fefferman, S. Mitter, H. Narayanan, Testing the manifold hypothesis, *J. Am. Math. Soc.* 29 (2016) 983–1049.
- [20] C.R. Genovese, M. Perone-Pacifico, I. Verdinelli, L. Wasserman, Manifold estimation and singular deconvolution under Hausdorff loss, *Ann. Stat.* 40 (2012) 941–963.
- [21] C.R. Genovese, M. Perone-Pacifico, I. Verdinelli, L. Wasserman, Minimax manifold estimation, *J. Mach. Learn. Res.* 13 (2012).
- [22] L. Györfi, M. Kohler, A. Krzyzak, H. Walk, et al., *A Distribution-Free Theory of Nonparametric Regression*, vol. 1, Springer, 2002.
- [23] J. He, R. Tsai, R. Ward, Side effects of learning from low-dimensional data embedded in a Euclidean space, *Res. Math. Sci.* 10 (13) (2023).
- [24] G.E. Hinton, R. Zemel, Autoencoders, minimum description length and Helmholtz free energy, *Adv. Neural Inf. Process. Syst.* 6 (1993).
- [25] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (1989) 359–366.
- [26] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint arXiv:1312.6114, 2013.
- [27] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, *Found. Trends Mach. Learn.* 12 (2019) 307–392.
- [28] M. Kirschbraun, Über die zusammenziehende und Lipschitzsche transformationen, *Fundam. Math.* 22 (1934) 77–108.
- [29] P.-S. Koutsourelakis, Accurate uncertainty quantification using inaccurate computational models, *SIAM J. Sci. Comput.* 31 (2009) 3274–3300.
- [30] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.* 37 (1991) 233–243.
- [31] J.M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, vol. 176, Springer Science & Business Media, 2006.
- [32] W. Liao, M. Maggioni, Adaptive geometric multiscale approximations for intrinsically low-dimensional data, *J. Mach. Learn. Res.* 20 (98) (2019) 1–63.
- [33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: European Conference on Computer Vision, Springer, 2014.
- [34] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, D.-R. Liou, Autoencoder for words, *Neurocomputing* 139 (2014) 84–96.
- [35] H. Liu, M. Chen, S. Er, W. Liao, T. Zhang, T. Zhao, Benefits of overparameterized convolutional residual networks: function approximation under smoothness constraint, arXiv preprint arXiv:2206.04569, 2022.
- [36] H. Liu, M. Chen, T. Zhao, W. Liao, Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks, in: International Conference on Machine Learning, PMLR, 2021.
- [37] H. Liu, H. Yang, M. Chen, T. Zhao, W. Liao, Deep nonparametric estimation of operators between infinite dimensional spaces, arXiv preprint arXiv:2201.00217, 2022.
- [38] M. Liu, Z. Cai, J. Chen, Adaptive two-layer ReLU neural network: I. Best least-squares approximation, *Comput. Math. Appl.* 113 (2022) 34–44.
- [39] M. Maggioni, S. Minsker, N. Strawn, Multiscale dictionary learning: non-asymptotic bounds and robustness, *J. Mach. Learn. Res.* 17 (2016) 43–93.
- [40] H. Montanelli, H. Yang, Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem, *Neural Netw.* 129 (2020) 1–6.
- [41] R. Nakada, M. Imaizumi, Adaptive approximation and generalization of deep neural network with intrinsic dimensionality, *J. Mach. Learn. Res.* 21 (2020) 1–38.
- [42] P. Niyogi, S. Smale, S. Weinberger, Finding the homology of submanifolds with high confidence from random samples, *Discrete Comput. Geom.* 39 (2008) 419–441.
- [43] L. Parussini, D. Venturi, P. Perdikaris, G.E. Karniadakis, Multi-fidelity Gaussian process regression for prediction of random fields, *J. Comput. Phys.* 336 (2017) 36–50.
- [44] P. Pope, C. Zhu, A. Abdulkader, M. Goldblum, T. Goldstein, The intrinsic dimension of images and its impact on learning, in: International Conference on Learning Representations, 2020.
- [45] N. Puchkin, V.G. Spokoiny, Structure-adaptive manifold estimation, *J. Mach. Learn. Res.* 23 (1) (2022) 40.
- [46] M. Ranzato, C. Poulton, S. Chopra, Y. Cun, Efficient learning of sparse representations with an energy-based model, *Adv. Neural Inf. Process. Syst.* 19 (2006).
- [47] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: International Conference on Machine Learning, PMLR, 2014.
- [48] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [49] J. Schmidt-Hieber, Deep ReLU network approximation of functions on a manifold, arXiv preprint arXiv:1908.00695, 2019.
- [50] S. Schonsheck, J. Chen, R. Lai, Chart auto-encoders for manifold structured data, arXiv preprint arXiv:1912.10094, 2019.
- [51] S.C. Schonsheck, S. Mahan, T. Klock, A. Cloninger, R. Lai, Semi-supervised manifold learning with complexity decoupled chart autoencoders, arXiv preprint arXiv:2208.10570, 2022.
- [52] U. Shaham, A. Cloninger, R.R. Coifman, Provable approximation properties for deep neural networks, *Appl. Comput. Harmon. Anal.* 44 (2018) 537–557.
- [53] Z. Shen, H. Yang, S. Zhang, Deep network approximation characterized by number of neurons, arXiv preprint arXiv:1906.05497, 2019.
- [54] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Publish or Perish, vol. 4, Incorporated, 1975.
- [55] R. Tang, Y. Yang, On empirical bayes variational autoencoder: an excess risk bound, in: Conference on Learning Theory, PMLR, 2021.
- [56] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (2000) 2319–2323.
- [57] C. Thäle, 50 years sets with positive reach—a survey, *Surv. Math. Appl.* 3 (2008) 123–165.
- [58] L.W. Tu, *Manifolds, An Introduction to Manifolds*, Springer, 2011, pp. 47–83.
- [59] A.W. Vaart, J.A. Wellner, *Weak Convergence and Empirical Processes: with Applications to Statistics*, Springer Science & Business Media, 1996.
- [60] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* 9 (2008).
- [61] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, 2008.
- [62] D. Yarotsky, Error bounds for approximations with deep ReLU networks, *Neural Netw.* 94 (2017) 103–114.
- [63] Z. Zhang, J. Wang, MLLE: modified locally linear embedding using multiple weights, *Adv. Neural Inf. Process. Syst.* 19 (2006).