Applied and Computational Harmonic Analysis 74 (2025) 101717

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

journal homepage: www.elsevier.com/locate/acha

Full Length Article

Generalization error guaranteed auto-encoder-based nonlinear model reduction for operator learning [☆]

Hao Liu [a], Biraj Dahal [b], Rongjie Lai [c], Wenjing Liao [b,*]

[a] *Department of Mathematics at Hong Kong Baptist University, Hong Kong Special Administrative Region of China*
[b] *School of Mathematics at Georgia Institute of Technology, United States of America*
[c] *Department of Mathematics at Purdue University, United States of America*

ARTICLE INFO

Communicated by Massimo Fornasier

MSC:
68T07

Keywords:
Deep learning theory
Generalization error
Model reduction
Auto-encoder
Operator learning

ABSTRACT

Many physical processes in science and engineering are naturally represented by operators between infinite-dimensional function spaces. The problem of operator learning, in this context, seeks to extract these physical processes from empirical data, which is challenging due to the infinite or high dimensionality of data. An integral component in addressing this challenge is model reduction, which reduces both the data dimensionality and problem size. In this paper, we utilize low-dimensional nonlinear structures in model reduction by investigating Auto-Encoder-based Neural Network (AENet). AENet first learns the latent variables of the input data and then learns the transformation from these latent variables to corresponding output data. Our numerical experiments validate the ability of AENet to accurately learn the solution operator of nonlinear partial differential equations. Furthermore, we establish a mathematical and statistical estimation theory that analyzes the generalization error of AENet. Our theoretical framework shows that the sample complexity of training AENet is intricately tied to the intrinsic dimension of the modeled process, while also demonstrating the robustness of AENet to noise.

1. Introduction

In the last two decades, deep learning has made remarkable successes in various fields such as computer vision [34,18], natural language processing [19], healthcare [47], and robotics [20], among others. More recently, deep neural networks have been extended to a wide range of applications in scientific computing. This expansion includes numerical partial differential equations (PDEs) [60,25,21,54,68], computational inverse problems [49,26], dynamics prediction [40], and model reduction [50,65,37,16,17], to name a few. These developments demonstrate the versatility and potential of deep learning in scientific machine learning.

In a wide array of scientific and engineering applications, numerous objects of interest are represented as functions or vector fields. For instance, many physical processes are modeled by operators that act between these function spaces. Differential equations are typical tools used to model such physical processes. With the advancement of machine learning, there has been a surge in data-driven approaches to understand physical processes. These methods enable the characterization and simulation of physical processes based on training data. In recent years, significant advances have been made in the realm of operator learning, which focuses on learning

[☆] This research is partially supported by National Natural Science Foundation of China 12201530, HKRGC ECS 22302123, NSF DMS-2401297, NSF DMS–2012652, NSF DMS-2145167 and DOE SC0024348.

[*] Corresponding author.
E-mail addresses: haoliu@hkbu.edu.hk (H. Liu), bdahal@gatech.edu (B. Dahal), lairj@purdue.edu (R. Lai), wliao60@gatech.edu (W. Liao).

https://doi.org/10.1016/j.acha.2024.101717
Received 7 February 2024; Received in revised form 8 July 2024; Accepted 7 October 2024

Available online 30 October 2024

unknown operators within functional spaces. Representative works include DeepONets [45] based on the universal approximation theory in Chen and Chen [10], Neural Operators [32,1,39], BelNet [69], etc. Mathematical theories on the approximation and generalization errors of operator learning can be found in Lanthaler et al. [35] for DeepONets, in Kovachki et al. [31] for FNO, and in Bhattacharya et al. [6], Liu et al. [43] for operator learning with dimension reduction techniques.

One of the major challenges of operator learning arises from the infinite or high dimensionality of the problem/data. Recent approximation theories on neural networks for operator learning [36] demonstrate that, operator learning methods, including DeepONet [45], NOMAD [58] and Fourier Neural Operator (FNO) [39], suffer from the curse of dimensionality without additional assumptions on low-dimensional structures. Specifically, Lanthaler and Stuart [36] proves that, there exists a r-times Fréchet differentiable functional, such that in order to achieve an ϵ approximation error for this functional, the network size of DeepONet, NOMAD (NOnlinear MAnifold Decoder) and FNO is lower bounded by $\exp(C\epsilon^{-1/(cr)})$ where C, c are constants depending on the problem. Such results demonstrate that, a huge network is needed to universally approximate r-times Fréchet differentiable functionals on an infinite dimensional space. It is impossible to reduce the network size unless additional structures about the operator (or input and output) are exploited.

Fortunately, the vast majority of real-world problems exhibit low-dimensional structures. For example, functions generated from translations or rotations only depend on few parameters [63,55,11]; Whale vocal signals can be parameterized by polynomial phase coefficients [66]; In molecular dynamics, the dynamical evolution is often governed by a small number of slow modes [13,14]. Thus it is natural to consider model reduction to reduce the data dimension and the problem size.

In the literature, linear reduction methods have shown considerable success when applied to models existing within low-dimensional linear spaces. Examples include the reduced-basis technique [53,56], proper orthogonal decomposition [7,23], Galerkin projection [23], and many others. A survey about model reduction can be found in Benner et al. [4,5]. More recently, linear model reduction methods have been combined with deep learning in various ways. Hesthaven and Ubbiali [22], Wang et al. [64] consider very low-dimensional inputs and employ Principal Component Analysis (PCA) [24] for the output space. Bhattacharya et al. [6] use PCA for both the input and output spaces. The active subspace method is used in O'Leary-Roseberry et al. [51] for dimension reduction. In these works, dimension reduction is achieved by existing linear model reduction methods, and operator learning is carried on the latent variables by a neural network. Theoretically, the network approximation error and stochastic error of PCA are analyzed in Bhattacharya et al. [6]. A generalization error analysis on operator learning with linear dimension reduction techniques is given in Liu et al. [43]. This paper shows that fixed linear encoders given by Fourier basis or Legendre polynomials give rise to a slow rate of convergence of the generalization error as n increases, and data-driven PCA encoders are suitable for input and output functions concentrated near low-dimensional linear subspaces.

However, many physical processes in practical applications are inherently nonlinear, such as fluid motion, nonlinear optical processes, and shallow water wave propagation. As a result, the functions of interest frequently reside on low-dimensional manifolds rather than within low-dimensional subspaces. Addressing these nonlinear structures is vital in model reduction. Recent studies have shown that deep neural networks are capable of representing a broad spectrum of nonlinear functions [67,44,61] and adapting to the low-dimensional structures of data [8,9,41,42,48]. Auto-Encoders, in particular, have gained widespread use in identifying low-dimensional latent variables within data [33,28]. Approximation and Statistical guarantees of Auto-Encoders for data near a low-dimensional manifold are established in Schonsheck et al. [57], Tang and Yang [62], Liu et al. [42].

In literature, Auto-Encoder-based neural networks have been proposed for model reduction in various ways [50,65,37,16,17,15,58,30,27]. Seidman et al. [58] assumes that the output functions in operator learning are concentrated near a low-dimensional manifold, and proposes NOnlinear MAnifold Decoder (NOMAD) for the solution submanifold. Numerical experiments in Seidman et al. [58] demonstrate that nonlinear decoders significantly outperform linear decoders, when the output functions are indeed on a low-dimensional manifold. In Kontolati et al. [30], Auto-Encoders are used to extract the latent features for the inputs and outputs respectively, and DeepONet is applied on latent features for operator learning. Numerical experiments in Kontolati et al. [30] demonstrate improved predictive accuracy when DeepONet is applied on latent features.

Despite the experimental success witnessed in Auto-Encoder-based neural networks for nonlinear model reduction, there is currently no established mathematical and statistical theory that can justify the heightened accuracy and reduced sample complexity achieved by these networks. Our paper aims to investigate this line of research through a comprehensive generalization error analysis of Auto-Encoder-based Neural Networks (AENet) within the context of nonlinear model reduction in operator learning. We present theoretical analysis that demonstrates the sample complexity of AENet depends on the intrinsic dimension of the model, rather than the dimension of its ambient space. This analysis provides a theoretical foundation for understanding how Auto-Encoder-based neural networks effectively exploit low-dimensional nonlinear structures in the realm of operator learning, offering a novel perspective on this subject.

1.1. Summary of our main results

This paper explores the use of Auto-Encoder-based neural network (AENet) for operator learning in function spaces, leveraging the Auto-Encoder-based nonlinear model reduction technique. Our goal is to achieve numerical success of nonlinear model reduction in comparison with linear model reduction methods. More importantly, as a novel part of this paper, we will establish a generalization error analysis in this context.

We explore AENet to handle the operator learning problems when the inputs are concentrated on a low-dimensional nonlinear manifold. Our algorithm has two stages. The first stage is to build an Auto-Encoder to learn the latent variable for the input. The second stage is to learn a transformation from the input latent variable to the output. The architecture of AENet is shown in Fig. 1(a). Furthermore, we provide a framework to analyze the generalization error and sample complexity of AENet.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets of functions in two Hilbert spaces and $\Psi : \mathcal{X} \to \mathcal{Y}$ be an unknown Lipschitz operator. Consider i.i.d. samples $\{u_i\}_{i=1}^{2n} \subset \mathcal{X}$ and the noisy outputs

$$\hat{v}_i = v_i + \epsilon_i, \text{ with } v = \Psi(u),$$

where the i.i.d. noise $\{\epsilon_i\}_{i=1}^{2n}$ is independent of the u_i's. Here $2n$ data are used for the ease of notation, as we will split the data into two subsets to train AENet, n data for each subset. The functions $u \in \mathcal{X}, v \in \mathcal{Y}$ are discretized as $S_{\mathcal{X}}(u) \in \mathbb{R}^{D_1}, S_{\mathcal{Y}}(v) \in \mathbb{R}^{D_2}$, where $S_{\mathcal{X}}$ and $S_{\mathcal{Y}}$ are discretization operators for functions in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Given the discretized data $\{S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\hat{v}_i)\}_{i=1}^{2n}$, we aim to learn the operator Ψ.

When the input $u \in \mathcal{X}$ is concentrated on a low-dimensional nonlinear set parameterized by d latent variables, we study AENet which learn the input latent variable and the operator in two stages.

Stage I: We use $\{S_{\mathcal{X}}(u_i)\}_{i=1}^{n}$ to train an Auto-Encoder $(E_{\mathcal{X}}^n, D_{\mathcal{X}}^n)$ with

$$E_{\mathcal{X}}^n : \mathbb{R}^{D_1} \to \mathbb{R}^d \text{ and } D_{\mathcal{X}}^n : \mathbb{R}^d \to \mathbb{R}^{D_1}$$

for the input. This Auto-Encoder gives rise to the input latent variable $E_{\mathcal{X}}^n \circ S_{\mathcal{X}}(u) \in \mathbb{R}^d$.

Stage II: We use $\{S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\hat{v}_i)\}_{i=n+1}^{2n}$ to learn a transformation from the input latent variable to the output:

$$\Gamma_{\mathrm{NN}}^n \in \underset{\Gamma'_{\mathrm{NN}} \in \mathcal{F}_{\mathrm{NN}}^{\Gamma}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=n+1}^{2n} \|\Gamma'_{\mathrm{NN}} \circ E_{\mathcal{X}}^n \circ S_{\mathcal{X}}(u_i) - S_{\mathcal{Y}}(\hat{v}_i)\|_{S_{\mathcal{Y}}}^2, \tag{1}$$

where $\|\cdot\|_{S_{\mathcal{Y}}}$ is the discretized counterpart of the function norm in $\mathcal{Y}$.

Combining Stage I and Stage II gives rise to the operator estimate in the discretized space

$$\Phi_{\mathrm{NN}}^n = \Gamma_{\mathrm{NN}}^n \circ E_{\mathcal{X}}^n : \ \mathbb{R}^{D_1} \to \mathbb{R}^{D_2},$$

which transforms the discretized input function $S_{\mathcal{X}}(u)$ to the discretized output function $S_{\mathcal{Y}}(v)$.

Numerical experiments are provided in Section 5 to learn the solutions of nonlinear PDEs from various initial conditions. We consider the transport equation for transportation models, the Burgers' equation with viscosity 10^{-3} in fluid mechanics and the Korteweg–De Vries (KdV) equation modeling waves on shallow water surfaces. Our experiments demonstrate that AENet significantly outperforms linear model reduction methods [6]. AENet is effective in handling nonlinear structures in the input, and are robust to noise.

This paper provides a solid mathematical and statistical estimation theory on the generalization error of AENet. Our Theorem 2 shows that, the squared generalization error decays in a power law as the sample size n increases, and the rate of decay depends on the intrinsic dimension d. Specifically, Theorem 2 proves the following upper bound on the squared generalization error:

$$\mathbb{E}_{\mathrm{Data}} \mathbb{E}_u \|\Phi_{\mathrm{NN}}^n \circ S_{\mathcal{X}}(u) - S_{\mathcal{Y}} \circ \Psi(u)\|_{S_{\mathcal{Y}}}^2 \le C(1+\sigma^2) n^{-\frac{1}{2+d}} \log^3 n, \tag{2}$$

where C is a constant depending on the model parameters, and σ^2 represents the variance of noise. The contributions of Theorem 2 are summarized below:

Leverage the dependence on intrinsic parameters: This theory justifies the benefits of model reduction by AENet. The rate of convergence for the generalization error depends on the intrinsic dimension d, even though the unknown operator is between two infinite-dimensional function spaces. To our best knowledge, this is the first statistical estimation theory on the generalization error of nonlinear model reduction by deep neural networks.

Robustness to noise: The constant $C(1+\sigma^2)$ in (2) is proportional to the variance of noise. Our result demonstrates that AENet is robust to noise. Moreover, AENet has a de-noising effect as the sample size increases, since squared generalization error decreases to 0 as n increases to ∞.

Dependence on the interpolation error: In some applications, test functions are discretized on a different grid as training functions. We can interpolate the test function on the training grid, and evaluate the output. In Remark 3, we show that, in this case the squared generalization error has an additional term about the interpolation error.

1.2. Organization

This paper is organized as follows: We provide preliminary definitions and discuss function discretization in Section 2. We then introduce the operator learning problem, explore nonlinear models and describe our AENet architecture in Section 3. Our main results, including the approximation theory and generalization error guarantees of AENet, are presented in Section 4. Numerical experiments are detailed in Section 5 and the proof of our main results is given in Section 6. Proofs of lemmas are postponed to Appendix B. Finally, we conclude our paper with discussions in Section 7.

2. Preliminaries and discretization of functions

In this section, we delineate the notations utilized throughout this paper. Additionally, we define key concepts such as Lipschitz operators, the Minkowski dimension and ReLU networks. Furthermore, we provide details about the function spaces of interest and the discretization operators employed in discretizing functions.

2.1. Notation

We use bold letters to denote vectors, and capital letters to denote matrices. For any vector $\mathbf{x} \in \mathbb{R}^d$, we denote its Euclidean norm by $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$, its ℓ^∞ norm by $\|\mathbf{x}\|_\infty = \sup_i |x_i|$, and its ℓ^1 norm by $\|\mathbf{x}\|_1 = \sum_i |x_i|$. We use $\|\cdot\|_0$ to denote the number of nonzero elements of its argument. We use $B_2^d(\mathbf{x}, \delta)$ to represent the open Euclidean ball in $\mathbb{R}^d$ centered at $\mathbf{x}$ with radius δ. Similarly, $B_\infty^d(\mathbf{x}, \delta)$ denotes the L^∞ ball in $\mathbb{R}^d$ centered at $\mathbf{x}$ with radius δ. We use $\#E$ to denote the cardinality of the set E and $|E|$ to denote the volume of E. For a function $u : \Omega \to \mathbb{R}$, its L^p norm is $\|u\|_{L^p(\Omega)} := \left(\int_\Omega |u(\mathbf{x})|^p d\mathbf{x}\right)^{1/p}$ and its L^∞ norm is $\|u\|_{L^\infty(\Omega)} := \sup_{\mathbf{x}\in\Omega} |u(\mathbf{x})|$. For a vector-valued function $\mathbf{f}$ defined on Ω, we denote $\|\mathbf{f}\|_{\infty,\infty} = \sup_{\mathbf{x}\in\Omega} \|\mathbf{f}(\mathbf{x})\|_\infty$. Throughout the paper, we use letters with a tilde to denote their discretized counterpart, letters with a subscript NN to denote networks, letters with a superscript n to denote empirical estimations.

2.2. Preliminaries

Definition 1 *(Lipschitz operators).* An operator $\Theta : \mathcal{A} \to \mathcal{B}$ is Lipschitz if

$$L_\Theta := \sup_{u_1 \neq u_2 \in \mathcal{A}} \frac{\|\Theta(u_1) - \Theta(u_2)\|_{\mathcal{B}}}{\|u_1 - u_2\|_{\mathcal{A}}} < \infty,$$

where L_Θ is called the Lipschitz constant of Θ.

Definition 2 *(Minkowski dimension).* Let $E \subset \mathbb{R}^D$. For any $\varepsilon > 0$, $\mathcal{N}(\varepsilon, E, \|\cdot\|_\infty)$ denotes the fewest number of ε-balls that cover E in terms of $\|\cdot\|_\infty$. The (upper) Minkowski dimension of E is defined as

$$d_M E := \limsup_{\varepsilon\to 0^+} \frac{\log \mathcal{N}(\varepsilon, E, \|\cdot\|_\infty)}{\log(1/\varepsilon)}.$$

The Minkowski dimension is also called the box-counting dimension. It describes how the box covering number $\mathcal{N}(\varepsilon, E, \|\cdot\|_\infty)$ scales with respect to the box side length ε. If $\mathcal{N}(\varepsilon, E, \|\cdot\|_\infty) \approx C\varepsilon^{-d}$, then $d_M E = d$.

Deep neural networks: We study the ReLU activated feedforward neural network (FNN):

$$f_{\rm NN}(\mathbf{x}) = W_L \cdot \mathrm{ReLU}\big(W_{L-1} \cdots \mathrm{ReLU}(W_1\mathbf{x} + \mathbf{b}_1) + \cdots + \mathbf{b}_{L-1}\big) + \mathbf{b}_L, \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^{c_0}$ for some positive integer c_0, W_l's are weight matrices with $W_l \in \mathbb{R}^{c_l\times c_{l-1}}$ and c_l denoting the output dimension of the l-th layer, $\mathbf{b}_l \in \mathbb{R}^{c_l}$ are biases and $\mathrm{ReLU}(a) = \max\{a, 0\}$ denotes the rectified linear unit (ReLU) which is applied elementwise. We consider the following class of FNNs

$$\begin{aligned}\mathcal{F}_{\rm NN}(d_1, d_2, L, p, K, \kappa, M) = \Big\{ & f_{\rm NN} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2} | f_{\rm NN}(\mathbf{x}) \text{ is in the form of (3) with } L \text{ layers,} \\ & \text{width bounded by } p, \|f_{\rm NN}\|_{L^\infty} \le M,\ \|W_l\|_{\infty,\infty} \le \kappa, \|\mathbf{b}_l\|_\infty \le \kappa,\ \textstyle\sum_{l=1}^L \|W_l\|_0 + \|\mathbf{b}_l\|_0 \le K \Big\},\end{aligned} \tag{4}$$

where $\|W\|_{\infty,\infty} = \max_{i,j} |W_{i,j}|$, $\|\mathbf{b}\|_\infty = \max_i |b_i|$ for any matrix W and vector $\mathbf{b}$.

2.3. Function spaces and discretization

We consider compact domains $\Omega_{\mathcal{X}} \subset \mathbb{R}^{d_{\Omega_{\mathcal{X}}}}$ and $\Omega_{\mathcal{Y}} \subset \mathbb{R}^{d_{\Omega_{\mathcal{Y}}}}$, and Hilbert spaces $L^2(\Omega_{\mathcal{X}})$ and $L^2(\Omega_{\mathcal{Y}})$. The space $L^2(\Omega_{\mathcal{X}}) := \{u : \Omega_{\mathcal{X}} \to \mathbb{R} : \int_{\Omega_{\mathcal{X}}} |u(\mathbf{x})|^2 d\mathbf{x} < \infty\}$ is equipped with the inner product $\langle u_1, u_2\rangle := \int_{\Omega_{\mathcal{X}}} u_1(\mathbf{x})u_2(\mathbf{x})d\mathbf{x}$, $\forall u_1, u_2 \in L^2(\Omega_{\mathcal{X}})$. The norms of $L^2(\Omega_{\mathcal{X}})$ and $L^2(\Omega_{\mathcal{Y}})$ are denoted by $\|\cdot\|_{\mathcal{X}} = \|\cdot\|_{L^2(\Omega_{\mathcal{X}})}$ and $\|\cdot\|_{\mathcal{Y}} = \|\cdot\|_{L^2(\Omega_{\mathcal{Y}})}$, respectively. Let $\mathcal{X} \subset L^2(\Omega_{\mathcal{X}})$ and $\mathcal{Y} \subset L^2(\Omega_{\mathcal{Y}})$. This paper considers differentiable input and output functions:

$$\mathcal{X} \subset C^1(\Omega_{\mathcal{X}}) := \{u \in L^2(\Omega_{\mathcal{X}}) :\ \nabla u \text{ is continuous}\}, \tag{5}$$

$$\mathcal{Y} \subset C^1(\Omega_{\mathcal{Y}}) := \{v \in L^2(\Omega_{\mathcal{Y}}) :\ \nabla v \text{ is continuous}\}, \tag{6}$$

and

$$\sup_{u\in\mathcal{X}} \sup_{\mathbf{x}\in\Omega_{\mathcal{X}}} \|\nabla u(\mathbf{x})\|_1 < \infty, \quad \sup_{v\in\mathcal{Y}} \sup_{\mathbf{y}\in\Omega_{\mathcal{Y}}} \|\nabla v(\mathbf{y})\|_1 < \infty. \tag{7}$$

In applications, functions need to be discretized. Let $\{\mathbf{x}_i\}_{i=1}^{D_1} \subset \Omega_{\mathcal{X}}$ and $\{\mathbf{y}_j\}_{j=1}^{D_2} \subset \Omega_{\mathcal{Y}}$ be the discretization grid on the $\Omega_{\mathcal{X}}$ and $\Omega_{\mathcal{Y}}$ domain respectively. The discretization operator on $\mathcal{X}$ and $\mathcal{Y}$ are

$$\begin{aligned} &\mathcal{S}_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^{D_1}, \text{ s.t. } \mathcal{S}_{\mathcal{X}}(u) = \{u(\mathbf{x}_i)\}_{i=1}^{D_1}, \\ &\mathcal{S}_{\mathcal{Y}} : \mathcal{Y} \to \mathbb{R}^{D_2}, \text{ s.t. } \mathcal{S}_{\mathcal{Y}}(v) = \{v(\mathbf{y}_j)\}_{j=1}^{D_2}. \end{aligned}$$

This discretization operator $\mathcal{S}_{\mathcal{X}}$ gives rise to an inner product and the induced norm on $\mathbb{R}^{D_1}$ such that

$$\langle \mathcal{S}_{\mathcal{X}}(u_1), \mathcal{S}_{\mathcal{X}}(u_2)\rangle_{\mathcal{S}_{\mathcal{X}}} = \sum_{i=1}^{D_1} \omega_i u_1(\mathbf{x}_i) u_2(\mathbf{x}_i), \tag{8}$$

where $\{\omega_i : \omega_i > 0\}_{i=1}^{D}$ is given by a proper quadrature rule for the integral $\langle u_1, u_2\rangle$. Popular quadrature rules in numerical analysis include the midpoint, trapezoidal, Simpson's rules, etc. [2]. The basic properties of the norms $\|\cdot\|_{\mathcal{S}_{\mathcal{X}}}$ and $\|\cdot\|_{\mathcal{S}_{\mathcal{Y}}}$ are given in Appendix C.

For regular function sets $\mathcal{X}$ and $\mathcal{Y}$ of practical interests, the convergence of Riemann integrals yields $\|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}}^2 \approx \|u\|_{L^2(\Omega_{\mathcal{X}})}^2$ for any $u \in \mathcal{X}$ and $\|\mathcal{S}_{\mathcal{Y}}(v)\|_{\mathcal{S}_{\mathcal{Y}}}^2 \approx \|v\|_{L^2(\Omega_{\mathcal{Y}})}^2$ for any $v \in \mathcal{Y}$, when the discretization grid is sufficiently fine. This motivates us to assume the following property:

Assumption 1. Suppose the function spaces $\mathcal{X}$ and $\mathcal{Y}$ are sufficiently regular such that: there exist discretization operators $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{S}_{\mathcal{Y}}$ satisfying the property:

$$0.5\|u\|_{\mathcal{X}} \le \|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}} \le 2\|u\|_{\mathcal{X}}, \quad 0.5\|v\|_{\mathcal{Y}} \le \|\mathcal{S}_{\mathcal{Y}}(v)\|_{\mathcal{S}_{\mathcal{Y}}} \le 2\|v\|_{\mathcal{Y}} \tag{9}$$

for all functions $u \in \mathcal{X}$ and $v \in \mathcal{Y}$.

Assumption 1 is a weak assumption which holds for large classes of regular functions as long as the discretization grid is sufficiently fine. For simplicity, we consider $\Omega_{\mathcal{X}} = [0,1]^{d_{\Omega_{\mathcal{X}}}}$ and $\mathcal{X} \subset C^1(\Omega_{\mathcal{X}})$. Suppose the grid points $\{\mathbf{x}_i\}_{i=1}^{D_1}$ are on a uniform grid of $[0,1]^{d_{\Omega_{\mathcal{X}}}}$ with spacing Δx and the quadrature rule in (8) is given by the Newton-Cotes formula where the integrand is approximated by splines. Piecewise constant, linear, quadratic approximations of the integrand give rise to the Midpoint, Trapezoid and Simpson rules, respectively. Taking the Midpoint rule as an example, we can express

$$\|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}} = \|I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u)\|_{L^2(\Omega_{\mathcal{X}})}, \text{ where } I_{\rm const} : \mathbb{R}^{D_1} \to L^2(\Omega_{\mathcal{X}})$$

is the piecewise constant interpolation operator, and $I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u)$ is the piecewise constant approximation of u. As a result,

$$\|u\|_{L^2(\Omega_{\mathcal{X}})} - \|I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u) - u\|_{L^2(\Omega_{\mathcal{X}})} \le \|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}} \le \|u\|_{L^2(\Omega_{\mathcal{X}})} + \|I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u) - u\|_{L^2(\Omega_{\mathcal{X}})}.$$

Assumption 1 holds as long as $\|I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u) - u\|_{L^2(\Omega_{\mathcal{X}})} \le \frac{1}{2}\|u\|_{L^2(\Omega_{\mathcal{X}})}$ uniformly for all functions $u \in \mathcal{X}$. By Calculus, piecewise constant approximation of u at a uniform grid with spacing Δx gives rise to the error

$$\|I_{\rm const} \circ \mathcal{S}_{\mathcal{X}}(u) - u\|_{L^\infty(\Omega_{\mathcal{X}})} \le \Delta x \sup_{\mathbf{x} \in \Omega_{\mathcal{X}}} \|\nabla u(\mathbf{x})\|_1$$

where ∇u denotes the gradient of u, and $\|\nabla u(\mathbf{x})\|_1$ is the ℓ^1 norm of the gradient vector $\nabla u(\mathbf{x})$.

If all functions in $\mathcal{X}$ satisfy mild conditions such that

$$\delta := \inf_{u \in \mathcal{X}} \left(\frac{\frac{1}{2}\|u\|_{L^2(\Omega_{\mathcal{X}})}}{\sup_{\mathbf{x} \in \Omega_{\mathcal{X}}} \|\nabla u(\mathbf{x})\|_1} \right) > 0, \tag{10}$$

then the discretization operator $\mathcal{S}_{\mathcal{X}}$ satisfies Assumption 1 for all the function $u \in \mathcal{X}$ as long as $\Delta x \le \delta$. Roughly speaking, the condition in (10) excludes functions whose function norm is too small, or whose derivative is too large. From the viewpoint of Fourier analysis, the condition in (10) excludes infinitely oscillatory functions.

Example 1. Let $A \ge a > 0$ and

$$\mathcal{X} = \left\{ \sum_{k=-N}^{N} a_k e^{2\pi i k x} : a \le |a_k| \le A \right\} \subset L^2([0,1]).$$

When the uniform sampling grid is sufficiently fine that

$$\Delta x \le \frac{a\sqrt{2N+1}}{2\pi A N(N+1)}, \tag{11}$$

then the discretization operator $\mathcal{S}_{\mathcal{X}}$ satisfies Assumption 1.

Example 1 is proved in Appendix A.1. In Example 1, the function set $\mathcal{X}$ includes Fourier series up to frequency N. Assumption 1 holds as long as the grid spacing is sufficiently small to resolve the resolution up to frequency N, as shown in (11). The larger N is, the more oscillatory the functions in $\mathcal{X}$ are, and therefore, Δx needs to be smaller.

When functions are discretized, an aliasing error occurs when the discretization grid is not fine enough to resolve the high frequency components of the function. The impact of aliasing on operator learning is studied in Bartolucci et al. [3]. In practice, aliasing can be avoided when the input and output functions are bandlimited and the discretization grid is fine enough, as shown in Example 1. Our Assumption 1 excludes the possibility of aliasing by requiring the discretization grid to be sufficiently fine so that all frequency components in the input and output can be resolved.

3. Nonlinear model reduction by AENet

In this section, we will present the problem setup and the AENet architecture.

3.1. Problem formulation

In this paper, we represent the unknown physical process by an operator $\Psi : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are subsets of two separable Hilbert spaces $L^2(\Omega_\mathcal{X})$ and $L^2(\Omega_\mathcal{Y})$ respectively. Our goal is to learn the operator Ψ from the given samples: $\{u_i, \hat{v}_i\}_{i=1}^{2n}$, where u_i is an input of Ψ and $\hat{v}_i$ is the noisy output. In practice, the functions u, v are discretized in the given data sets $\{\mathcal{S}_\mathcal{X}(u_i), \mathcal{S}_\mathcal{Y}(\hat{v}_i)\}_{i=1}^{2n}$.

Setting 1. Let $\Omega_\mathcal{X} \subset \mathbb{R}^{d_{\Omega_\mathcal{X}}}$ and $\Omega_\mathcal{Y} \subset \mathbb{R}^{d_{\Omega_\mathcal{Y}}}$ be compact domains, and $\mathcal{X} \subset C^1(\Omega_\mathcal{X}) \subset L^2(\Omega_\mathcal{X})$ and $\mathcal{Y} \subset C^1(\Omega_\mathcal{Y}) \subset L^2(\Omega_\mathcal{Y})$ such that (7) holds. Suppose the function sets $\mathcal{X}$ and $\mathcal{Y}$ and the discretization operators $\mathcal{S}_\mathcal{X}$ and $\mathcal{S}_\mathcal{Y}$ satisfy Assumption 1. The unknown operator $\Psi : \mathcal{X} \to \mathcal{Y}$ is Lipschitz with Lipschitz constant $L_\Psi > 0$, and γ is a probability measure on $\mathcal{X}$. Suppose $\{u_i\}_{i=1}^{2n}$ are i.i.d. samples from γ and the $\hat{v}_i$'s are generated according to model:

$$v_i = \Psi(u_i) \text{ and } \hat{v}_i = v_i + \epsilon_i, \tag{12}$$

where the ϵ_i's are i.i.d. samples from a probability measure on $\mathcal{Y}$, independently of the u_i's. The given data are

$$\mathcal{J} = \{\mathcal{S}_\mathcal{X}(u_i), \mathcal{S}_\mathcal{Y}(\hat{v}_i)\}_{i=1}^{2n}, \tag{13}$$

where $\mathcal{S}_\mathcal{Y}(\hat{v}_i) = \mathcal{S}_\mathcal{Y}(v_i) + \mathcal{S}_\mathcal{Y}(\epsilon_i)$.

For simplicity, we denote the discretized functions as $\tilde{u} = \mathcal{S}_\mathcal{X}(u)$ and $\tilde{v} = \mathcal{S}_\mathcal{Y}(v)$ for the rest of the paper. Additional assumptions on the measure γ and noise ε are stated in Assumption 2 and 3 in the following subsections, respectively.

3.2. Low-dimensional nonlinear models

Even though $L^2(\Omega_\mathcal{X})$ and $L^2(\Omega_\mathcal{Y})$ are infinite-dimensional function spaces, the functions of practical interests often exhibit low-dimensional structures. The simplest low-dimensional model is the linear subspace model. However, a large amount of functions in real-world applications exhibit nonlinear structures. For example, functions generated from translations or rotations have a nonlinear dependence on few parameters [63,55,11], which motivates us to consider functions with a low-dimensional nonlinear parameterization.

Assumption 2. In Setting 1, the probability measure γ is supported on a low-dimensional set $\mathcal{M} \subset \mathcal{X}$ such that: There exist Lipschitz maps

$$\mathbf{f} : \mathcal{M} \to [-1,1]^d, \text{ and } \mathbf{g} : [-1,1]^d \to \mathcal{M}$$

such that $u = \mathbf{g} \circ \mathbf{f}(u)$ for any $u \in \mathcal{M}$. The Lipchitz constants $\mathbf{f}$ and $\mathbf{g}$ are $L_\mathbf{f} > 0$ and $L_\mathbf{g} > 0$ respectively, such that

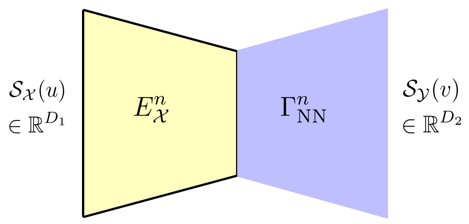
$$\|\mathbf{f}(u_1) - \mathbf{f}(u_2)\|_2 \le L_\mathbf{f} \|u_1 - u_2\|_\mathcal{X}, \quad \|\mathbf{g}(\mathbf{z}_1) - \mathbf{g}(\mathbf{z}_2)\|_\mathcal{X} \le L_\mathbf{g} \|\mathbf{z}_1 - \mathbf{z}_2\|_2$$

for any $u_1, u_2 \in \mathcal{M}$, and $\mathbf{z}_1, \mathbf{z}_2 \in [-1,1]^d$. Additionally, there exists $c_1 > 0$ such that

$$\mathcal{M} \subset \{u : \ \|u\|_{L^\infty(\Omega_\mathcal{X})} \le c_1 \|u\|_\mathcal{X}\}, \tag{14}$$

and $R_\mathcal{Y} := \sup_{u \in \mathcal{M}} \{\|\Psi(u)\|_{L^\infty(\Omega_\mathcal{Y})}\} < \infty$.

Assumption 2 says that, even though the input u is in the infinite-dimensional space, it can be parameterized by a d-dimensional latent variable. The intrinsic dimension of the inputs is d. Assumption 2 includes linear and nonlinear models since $\mathbf{f}$ and $\mathbf{g}$ can be linear and nonlinear maps. The condition in (14) is a mild assumption excluding the case that the large values of u concentrate at a set with a small Lebesgue measure. Assumption 2 implies that $R_\mathcal{X} := \sup_{u \in \mathcal{M}} \|u\|_{L^\infty(\Omega_\mathcal{X})} < \infty$. Assumption 2 also implies a low-dimensional parameterization of $\mathcal{S}_\mathcal{X}(u)$. We denote the range of $\mathcal{M}$ under $\mathcal{S}_\mathcal{X}$ by



(a) AENet architecture

$$\begin{tikzcd}
u \in \mathcal{M} \subset \mathcal{X} \arrow[rr, "\Psi"] \arrow[d, "\mathcal{S}_{\mathcal{X}}"'] & & v = \Psi(u) \in \mathcal{Y} \arrow[d, "\mathcal{S}_{\mathcal{Y}}"] \\
\widetilde{u} \in \widetilde{\mathcal{M}} \subset \mathbb{R}^{D_1} \arrow[r, "\widetilde{\mathbf{f}}"] & \mathbb{R}^d \arrow[r, "\Gamma"] & \widetilde{v} = \mathcal{S}_{\mathcal{Y}}(v) \in \mathbb{R}^{D_2}
\end{tikzcd}$$

(b) A transformation flow chart

Fig. 1. An illustration of the AENet architecture and the transformation flow chart. The oracle transformation Φ has a dimension reduction component $\widetilde{\mathbf{f}}$ and a forward transformation component Γ. These two components are marked in red. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\widetilde{\mathcal{M}} := \mathcal{S}_{\mathcal{X}}(\mathcal{M}) = \{\mathcal{S}_{\mathcal{X}}(u) : \ u \in \mathcal{M}\} \subset \mathbb{R}^{D_1}.$$

Lemma 1. *In Setting 1 and under Assumptions 1 and 2, every point in $\widetilde{\mathcal{M}}$ exhibits the low-dimensional parameterization: $\widetilde{\mathbf{f}} : \ \widetilde{\mathcal{M}} \subset \mathbb{R}^{D_1} \to [-1,1]^d$, such that $\widetilde{\mathbf{f}}(\mathcal{S}_{\mathcal{X}}(u)) := \mathbf{f}(u)$ and $\widetilde{\mathbf{g}} : \ [-1,1]^d \to \widetilde{\mathcal{M}} \subset \mathbb{R}^{D_1}$, such that $\widetilde{\mathbf{g}}(\theta) := \mathcal{S}_{\mathcal{X}} \circ \mathbf{g}(\theta)$ which guarantees*

$$\widetilde{\mathbf{g}} \circ \widetilde{\mathbf{f}}(\mathcal{S}_{\mathcal{X}}(u)) = \mathcal{S}_{\mathcal{X}}(u), \ \forall u \in \mathcal{M}. \tag{15}$$

$\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{g}}$ are Lipschitz with Lipchitz constants bounded by $2L_{\mathbf{f}}$ and $2L_{\mathbf{g}}$ respectively.

Lemma 1 is proved in Appendix B.1. The low-dimensional parameterizations in Assumption 2 and Lemma 1 motivate us to perform nonlinear dimension reductions of $\widetilde{\mathcal{M}}$, shown in Fig. 1(b). Another advantage is that $\widetilde{\mathcal{M}}$ is a d-dimensional manifold in $\mathbb{R}^{D_1}$. The following lemma shows that $\widetilde{\mathcal{M}}$ has Minkowski dimension no more than d (proof in Appendix B.2).

Lemma 2. *In Setting 1 and under Assumptions 1 and 2, the Minkowski dimension of $\widetilde{\mathcal{M}}$ is no more than d:*

$$d_M \widetilde{\mathcal{M}} \le d.$$

3.3. AENet

Since functions are always discretized in numerical simulations, in order to learn the operator Ψ, it is sufficient to learn the transformation Φ on the discretized objects between $\widetilde{u} := \mathcal{S}_{\mathcal{X}}(u) \in \mathbb{R}^{D_1}$ and $\widetilde{v} := \mathcal{S}_{\mathcal{Y}}(v) \in \mathbb{R}^{D_2}$. Specifically, $\Phi : \mathbb{R}^{D_1} \to \mathbb{R}^{D_2}$ is the oracle transformation such that $\Phi \circ \mathcal{S}_{\mathcal{X}}(u) = \mathcal{S}_{\mathcal{Y}} \circ \Psi(u), \ \forall u \in \mathcal{X}$. In some applications, the training and test data are sampled on different grids. We will discuss interpolation and the discretization-invariant evaluation for the test data at the end of Section 4.

In this paper, we study AENet which learns the input latent variable by an Auto-Encoder, and then learns the transformation from this input latent variable to the output. The architecture of AENet is demonstrated in Fig. 1 (a). AENet aims to approximate the oracle transformation

$$\Phi = \mathcal{S}_{\mathcal{Y}} \circ \Psi \circ \mathbf{g} \circ \widetilde{\mathbf{f}} : \widetilde{\mathcal{M}} \subset \mathbb{R}^{D_1} \to \mathbb{R}^{D_2}$$

such that

$$\Phi(\mathcal{S}_{\mathcal{X}}(u)) = \mathcal{S}_{\mathcal{Y}}(v).$$

The oracle transformation Φ has two components shown in Fig. 1(b):

- *A dimension reduction component:* $\widetilde{\mathbf{f}} : \mathbb{R}^{D_1} \to \mathbb{R}^d$;
- *A forward transformation component:*

$$\Gamma := \mathcal{S}_{\mathcal{Y}} \circ \Psi \circ \mathbf{g} : \mathbb{R}^d \to \mathbb{R}^{D_2}. \tag{16}$$

We propose to learn AENet in two stages. Given the training data $\mathcal{J} = \{\widetilde{u}_i, \mathcal{S}_{\mathcal{Y}}(\widehat{v}_i)\}_{i=1}^{2n}$ in (13) with $\widetilde{u}_i = \mathcal{S}_{\mathcal{X}}(u_i)$, we split the data into two subsets $\mathcal{J}_1 = \{\widetilde{u}_i, \mathcal{S}_{\mathcal{Y}}(\widehat{v}_i)\}_{i=1}^{n}$ and $\mathcal{J}_2 = \{\widetilde{u}_i, \mathcal{S}_{\mathcal{Y}}(\widehat{v}_i)\}_{i=n+1}^{2n}$ (data can be split unevenly as well), where $\mathcal{J}_1$ is used to build the Auto-Encoder for the input space and $\mathcal{J}_2$ is used to learn the transformation Γ from the input latent variable to the output.

Stage I: Based on the inputs $\{\widetilde{u}_i\}_{i=1}^n$ in $\mathcal{J}_1$, we learn an Auto-Encoder for the input space. The encoder $E^n_{\mathcal{X}} : \widetilde{\mathcal{M}} \to \mathbb{R}^d$ and the corresponding decoder $D^n_{\mathcal{X}} : \mathbb{R}^d \to \mathbb{R}^{D_1}$ are given by the minimizers of the empirical mean squared loss

$$(D^n_{\mathcal{X}}, E^n_{\mathcal{X}}) = \underset{D_{\mathcal{X}} \in \mathcal{F}^{D_{\mathcal{X}}}_{\mathrm{NN}}, E_{\mathcal{X}} \in \mathcal{F}^{E_{\mathcal{X}}}_{\mathrm{NN}}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \|\widetilde{u}_i - D_{\mathcal{X}} \circ E_{\mathcal{X}}(\widetilde{u}_i)\|^2_{\mathcal{S}_{\mathcal{X}}}, \tag{17}$$

with proper network architectures of $\mathcal{F}_{\rm NN}^{E_\mathcal{X}}$ and $\mathcal{F}_{\rm NN}^{D_\mathcal{X}}$. The Auto-Encoder in Stage I yields the input latent variable $E_\mathcal{X}^n(u) \in \mathbb{R}^d$. Stage I of AENet is represented by the yellow part of Fig. 1 (a).

Stage II: We next learn a transformation $\Gamma_{\rm NN}$ between the input latent variable $E_\mathcal{X}^n(\widetilde{u})$ and the output $\mathcal{S}_\mathcal{Y}(v)$ using the data in $\mathcal{J}_2$ by solving the optimization problem in (1) to obtain $\Gamma_{\rm NN}^n$. Stage II of AENet is indicated by the blue part of Fig. 1 (a).

Finally, the unknown transformation Φ is estimated by

$$\Phi_{\rm NN}^n = \Gamma_{\rm NN}^n \circ E_\mathcal{X}^n.$$

The performance of AENet can be measured by the squared generalization error

$$\mathbb{E}_\mathcal{J} \mathbb{E}_{u\sim\gamma} \|\Phi_{\rm NN}^n \circ \mathcal{S}_\mathcal{X}(u) - \mathcal{S}_\mathcal{Y}\circ\Psi(u)\|_{\mathcal{S}_\mathcal{Y}}^2,$$

where $\mathbb{E}_{u\sim\gamma}$ is the expectation over the test sample $u \sim \gamma$, and $\mathbb{E}_\mathcal{J}$ is the expectation over the joint distribution of training samples.

4. Main results on approximation and generalization errors

We will state our main theoretical results on approximation and generalization errors in this section and defer detailed proof in Section 6. Our results show that AENet can efficiently learn the input latent variables and the operator with the sample complexity depending on the intrinsic dimension of the model.

4.1. Approximation theory of AENet

Our first result is on the approximation theory of AENet. We show that, the oracle transformation Φ, its reduction component $\widetilde{\mathbf{f}}$ and transformation component Γ can be well approximated by neural networks with proper architectures.

Theorem 1. *In Setting 1, suppose Assumptions 1 and 2 hold.*

(i) For any $\varepsilon_1 \in (0, 1/4), \varepsilon_2 \in (0, 1)$, choose the network architectures $\mathcal{F}_{\rm NN}^{E_\mathcal{X}} = \mathcal{F}_{\rm NN}(D_1, d, L_1, p_1, K_1, \kappa_1, M_1)$ and $\mathcal{F}_{\rm NN}^{D_\mathcal{X}} = \mathcal{F}_{\rm NN}(d, D_1, L_2, p_2, K_2, \kappa_2, M_2)$ with parameters

$$L_1 = O(\log \varepsilon_1^{-1}),\ p_1 = O(\varepsilon_1^{-d}),\ K_1 = O(\varepsilon_1^{-d}),\ \kappa_1 = O(\varepsilon_1^{-7}),\ M_1 = 1,$$
$$L_2 = O(\log \varepsilon_2^{-1}),\ p_2 = O(\varepsilon_2^{-d}),\ K_2 = O(\varepsilon_2^{-d}\log \varepsilon_2^{-1}),\ \kappa_2 = O(\varepsilon_2^{-1}),\ M_2 = R_\mathcal{X}.$$

There exists $\widetilde{\mathbf{f}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{E_\mathcal{X}}$ and $\widetilde{\mathbf{g}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{D_\mathcal{X}}$ satisfying

$$\sup_{\widetilde{u}\in\widetilde{\mathcal{M}}} \|\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \le \varepsilon_1,$$
$$\sup_{\mathbf{z}\in[-1,1]^d} \|\widetilde{\mathbf{g}}_{\rm NN}(\mathbf{z}) - \widetilde{\mathbf{g}}(\mathbf{z})\|_\infty \le \varepsilon_2,$$
$$\sup_{\widetilde{u}\in\widetilde{\mathcal{M}}} \|\widetilde{\mathbf{g}}_{\rm NN}\circ\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \widetilde{\mathbf{g}}\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \le \varepsilon_2 + 2\sqrt{d}L_\mathbf{g}\varepsilon_1.$$

The constant hidden in $O(\cdot)$ depends on $d, L_\mathbf{f}, L_\mathbf{g}, R_\mathcal{X}$ and is polynomial in D_1.

(ii) For any $\varepsilon_3 \in (0, 1)$, there exists a network $\Gamma_{\rm NN} \in \mathcal{F}_{\rm NN}(d, D_2, L_3, p_3, K_3, \kappa_3, M_3)$ with

$$L_3 = O(\log \varepsilon_3^{-1}),\ p_3 = O(\varepsilon_3^{-d}),\ K_3 = O(\varepsilon_3^{-d}\log \varepsilon_3^{-1}),\ \kappa_3 = O(\varepsilon_3^{-1}),\ M_3 = R_\mathcal{Y}$$

satisfying

$$\sup_{\mathbf{z}\in[-1,1]^d} \|\Gamma_{\rm NN}(\mathbf{z}) - \Gamma(\mathbf{z})\|_\infty \le \varepsilon_3.$$

The constant hidden in $O(\cdot)$ depends on $d, L_\Psi, L_\mathbf{g}$ and is linear in D_2.

Theorem 1 provides a construction of three neural networks to approximate $\widetilde{\mathbf{f}}, \widetilde{\mathbf{g}}, \Gamma$ with accuracy $\varepsilon_1, \varepsilon_2, \varepsilon_3$ respectively. A proper choice of ε_1 and ε_3 yields the following approximation result on the oracle transformation Φ:

Corollary 1. *Under the assumptions in Theorem 1, for any $\varepsilon \in (0, 1)$, set $\varepsilon_1 = \frac{\varepsilon}{4\sqrt{d}L_\Psi L_\mathbf{g}}, \varepsilon_3 = \frac{\varepsilon}{2}$ in Theorem 1 and denote the network $\Phi_{\rm NN} = \Gamma_{\rm NN}\circ\widetilde{\mathbf{f}}_{\rm NN}$. Then we have*

$$\sup_{\widetilde{u}\in\widetilde{\mathcal{M}}} \|\Phi_{\rm NN}(\widetilde{u}) - \Phi(\widetilde{u})\|_\infty \le \varepsilon.$$

Theorem 1 and Corollary 1 are proved in Section 6.1 and 6.2 respectively. Corollary 1 provides an explicit construction of neural networks to approximate the oracle transformation Φ. It demonstrates that AENet with properly chosen parameters has the representation power for the oracle transformation with an arbitrary accuracy ε.

Remark 1. In Corollary 1, the input is a discretization by $\mathcal{S}_\mathcal{X}$, and $\Phi_{\rm NN}$ is constructed for inputs discretized by $\mathcal{S}_\mathcal{X}$. By interpolating functions on $\Omega_\mathcal{X}$, we can apply this network to functions discretized on a different grid. Suppose a new input u is discretized by another discretization operator $\mathcal{S}'_\mathcal{X}(u)\in\mathbb{R}^{D'_1}$, where the grid points of $\mathcal{S}_\mathcal{X}$ and $\mathcal{S}'_\mathcal{X}$ are different. Then $\mathcal{S}'_\mathcal{X}(u)$ cannot be directly passed to $\Phi_{\rm NN}$. Let $P_{{\rm intp},\mathcal{X}}$ be an interpolation operator from the grid to the whole domain $\Omega_\mathcal{X}$. For the input $\mathcal{S}'_\mathcal{X}(u)$, we can interpolate it by $P_{{\rm intp},\mathcal{X}}$ and then discretize it by $\mathcal{S}_\mathcal{X}$ to obtain $\mathcal{S}_\mathcal{X}(P_{{\rm intp},\mathcal{X}}(\mathcal{S}'_\mathcal{X}(u)))\in\mathbb{R}^{D_1}$. Based on this setting and under the same condition of Corollary 1, we have (see details in Appendix A.2)

$$\sup_{u\in\mathcal{M}}\|\Phi_{\rm NN}\circ\mathcal{S}_\mathcal{X}(P_{{\rm intp},\mathcal{X}}(\mathcal{S}'_\mathcal{X}(u)))-\Phi(\mathcal{S}_\mathcal{X}(u))\|_\infty\le\varepsilon+\sup_{u\in\mathcal{M}}\|\Phi_{\rm NN}\circ\mathcal{S}_\mathcal{X}(P_{{\rm intp},\mathcal{X}}(\mathcal{S}'_\mathcal{X}(u)))-\Phi_{\rm NN}\circ\mathcal{S}_\mathcal{X}(u)\|_\infty, \tag{18}$$

where the first term captures the network approximation error, the second term arises from the interpolation error on $\Omega_\mathcal{X}$.

Remark 2. In Petersen and Voigtlaender [52], it is shown that all upper and lower bounds concerning approximation rates of fully-connected neural networks for functions translate to essentially the same bounds concerning approximation rates of convolutional neural networks. By utilizing Petersen and Voigtlaender [52, Theorem 4.1], one can translate our approximation results in Theorem 1 and Corollary 1 to convolutional neural networks.

4.2. Generalization error of AENet

Our second main result is on the generalization error of AENet with i.i.d. sub-Gaussian noise.

Assumption 3. Let $\mathcal{S}_\mathcal{Y}$ be the discretization operator in $\mathcal{Y}$ under Setting 1. The noise distribution μ in Setting 1 satisfies: For any sample $\epsilon\sim\mu$ (a random function defined on $\Omega_\mathcal{Y}$), $\{(\mathcal{S}_\mathcal{Y}(\epsilon))_k\}_{k=1}^{D_2}$ with $(\mathcal{S}_\mathcal{Y}(\epsilon))_k=\epsilon(\mathbf{y}_k)$ are i.i.d. sub-Gaussian with $\mathbb{E}\left[(\mathcal{S}_\mathcal{Y}(\epsilon))_k\right]=0$ and variance proxy σ^2 for $k=1,\dots,D_2$.

Theorem 2. *Consider Setting 1 and suppose Assumptions 1, 2 and 3 hold. In Stage I, set the network architectures $\mathcal{F}_{\rm NN}^{E_\mathcal{X}}=\mathcal{F}_{\rm NN}(D_1,d,L_1,p_1,K_1,\kappa_1,M_1)$ and $\mathcal{F}_{\rm NN}^{D_\mathcal{X}}=\mathcal{F}_{\rm NN}(d,D_1,L_2,p_2,K_2,\kappa_2,M_2)$ with parameters*

$$L_1=O(\log n),\ p_1=O(n^{\frac{d}{2+d}}),\ K_1=O(n^{\frac{d}{2+d}}),\ \kappa_1=O(n^{\frac{7}{2+d}}),\ M_1=1, \tag{19}$$

$$L_2=O(\log n),\ p_2=O(n^{\frac{d}{2+d}}),\ K_2=O(n^{\frac{d}{2+d}}\log n),\ \kappa_2=O(n^{\frac{1}{2+d}}),\ M_2=R_\mathcal{X}. \tag{20}$$

In Stage II, set the network architecture $\mathcal{F}_{\rm NN}^\Gamma=\mathcal{F}_{\rm NN}(d,D_2,L_3,p_3,K_3,\kappa_3,M_3)$ with

$$L_3=O(\log n),\ p_3=O(n^{\frac{d}{2(2+d)}}),\ K_3=O(n^{\frac{d}{2(2+d)}}\log n),\ \kappa_3=O(n^{\frac{7}{2(2+d)}}),\ M_3=R_\mathcal{Y}. \tag{21}$$

Let $E_\mathcal{X}^n, D_\mathcal{X}^n$ be the empirical minimizer of (17) in Stage I, and $\Gamma_{\rm NN}^n$ be the empirical minimizer of (1) in Stage II. For $n>4^{2+d}$, the squared generalization error of $\Phi_{\rm NN}^n=\Gamma_{\rm NN}^n\circ E_\mathcal{X}^n$ satisfies

$$\mathbb{E}_\mathcal{J}\mathbb{E}_{u\sim\gamma}\|\Phi_{\rm NN}^n\circ\mathcal{S}_\mathcal{X}(u)-\mathcal{S}_\mathcal{Y}\circ\Psi(u)\|_{\mathcal{S}_\mathcal{Y}}^2\le C(1+\sigma^2)n^{-\frac{1}{2+d}}\log^3 n, \tag{22}$$

for some $C>0$. The constant hidden in $O(\cdot)$ and C depend on $d, L_\mathbf{f},\ L_\mathbf{g},\ L_\Psi,\ R_\mathcal{X},\ R_\mathcal{Y},\ |\Omega_\mathcal{X}|,\ |\Omega_\mathcal{Y}|$ and is polynomial in D_1 and is linear in D_2.

Theorem 2 is proved in Section 6.3. Its contributions are summarized in the introduction.

Remark 3. In Theorem 2, $\Phi_{\rm NN}^n$ is trained on the inputs discretized by $\mathcal{S}_\mathcal{X}$. The result of Theorem 2 can be applied to a new input discretized on a different grid, as considered in Remark 1. Under the condition of Theorem 2, we have (see details in Appendix A.3)

$$\begin{aligned}&\mathbb{E}_\mathcal{J}\mathbb{E}_{u\sim\gamma}\|\Phi_{\rm NN}^n\circ\mathcal{S}_\mathcal{X}(P_{{\rm intp},\mathcal{X}}(\mathcal{S}'_\mathcal{X}(u)))-\mathcal{S}_\mathcal{Y}\circ\Psi(u)\|_{\mathcal{S}_\mathcal{Y}}^2\\ \le& C(1+\sigma^2)n^{-\frac{1}{2+d}}\log^3 n+\mathbb{E}_\mathcal{J}\mathbb{E}_{u\sim\gamma}\|\Phi_{\rm NN}^n\circ\mathcal{S}_\mathcal{X}(P_{{\rm intp},\mathcal{X}}(\mathcal{S}'_\mathcal{X}(u)))-\Phi_{\rm NN}^n\circ\mathcal{S}_\mathcal{X}(u)\|_{\mathcal{S}_\mathcal{Y}}^2,\end{aligned} \tag{23}$$

where the first term captures the network estimation error and the second term arises from the interpolation error on $\Omega_\mathcal{X}$.

Remark 4. The setting in this paper has some similarity with that considered in Liu et al. [43, Theorem 15] in the sense that the input functions are assumed to exhibit a low-dimensional nonlinear structure. However, there are fundamental differences in the encoding procedure and the utilization of latent features, as well as in proof techniques.
Encoding Procedure: In Liu et al. [43], the encoder and decoder are either given or to be estimated, which are assumed to be Lipschitz and the Lipschitz constants of the encoder and decoder are upper bounded by a constant. In this paper, we use a nonlinear Auto-Ecoder for encoding and decoding. Due to the complicated structure of the nonlinear encoding and decoding networks, their

Lipschitz constants are not guaranteed to be upper-bounded by a constant. As a result, the theory in this paper can not be implied from that of Liu et al. [43], and highly nontrivial techniques are developed in this paper to allow encoding and decoding by Auto-Encoder.
Utilization of Latent Features: In model reduction, one is not only interested in the prediction of the output given an input, but also interested in the low-dimensional latent feature. [43] focuses on operator learning to predict the output given an input, and a single feedforward network in [43] does not necessarily provide meaningful low-dimensional latent features which represent the geometry of underlying low-dimensional structure. In comparison, this paper focuses on learning meaningful low-dimensional latent features, as well as the output predicted from the low-dimensional latent feature, which is beyond the scope of [43].
Proof Technique: Since this paper aims to learn meaningful low-dimensional latent features in the setting of operator learning, the proof is more challenging and complicated than that of [43, Theorem 15]. In this paper, a careful analysis is performed to guarantee that the Auto-Encoder is "well behaved". By "well-behaved", we mean that the Auto-Encoder has sufficient representation power and well-controlled variance. With these properties in place, the operator can be effectively learned.

5. Numerical experiments

We next present several numerical experiments to demonstrate the efficacy of AENet. We consider the solution operators of the linear transport equation, the nonlinear viscous Burgers' equation, and the Korteweg-de Vries (KdV) equation.

For all examples, we use a 512 dimensional equally spaced grid for the spatial domain. The networks are all fully connected feed-forward ReLU networks as in our theory. We trained every neural network for 500 epochs with the Adam optimization algorithm using the MSE loss, a learning rate of 10^{-3}, and a batch size of 64. For training of the neural networks, all data (input and outputs function values) were scaled down to fit into the range $[-1,1]$.

All training was done with 2000 training samples and 500 test samples, except for the training involved in Fig. 4(e), Fig. 8(e), and Fig. 11(e) where the training sample varies.

For all examples, the input data has a low dimensional nonlinear structure. Indeed, the input data matrix (by stacking the initial conditions) for all examples we consider has slowly decaying singular values (see Fig. 2(a), 6(a) and 9(a)), which indicates the shortcomings of using a linear encoder and the necessity of using a nonlinear encoder as in AENet. Additional plots showing the nonlinearity of the data can be found in Fig. 2(b), 6(b), and 9(b).

We compare AENet with two methods involving dimension reduction and neural networks. PCANet refers to the method in Bhattacharya et al. [6], which consists of a PCA encoder for the input, a PCA decoder for the output, and a neural network in between. We also consider DeepONet [45] implemented with the DeepXDE package [46], a popular method for operator learning that also involves a dimension reduction component (i.e. the branch net). In DeepONet, we take the output dimension of the branch/trunk net as the reduced dimension.

For all examples, we implement the Auto-Encoder for AENet with layer widths 500, 500, 500, d_{ae}, 500, 500, 500, and we implement the operator neural network for AENet and PCANet with layer widths 500, 500, 500. We use 40 dimensional PCA for the output in PCANet. We use a simple unstacked DeepONet. The total computation time for each model is shown in Table 1.

5.1. Transport equation

We consider the linear transport equation given by

$$u_t = -u_x, \quad x \in [0,1], t \in [0,0.3] \tag{24}$$

with zero Dirichlet boundary condition and initial condition $u(x,0) = g(x), x \in [0,1]$. We seek to approximate the operator Ψ that takes $g(x) = u(x,0)$ as input and outputs $u(x,0.3)$ from the solution of (24) at $t = 0.3$. We consider the weak version of this PDE, allowing us to consider an g that is not differentiable everywhere. Note that the analytic solution to this equation is $u(x,t) = g(x-t)$.

Let $\sigma(x) = \max(x,0)$, and fix $\epsilon = 0.05$. For any α and t, define the "hat" function

$$H_{\alpha,t}(x) = \frac{2\alpha}{\epsilon}\left(\sigma(x-t) - 2\sigma\left(x-t-\frac{\epsilon}{2}\right) + \sigma(x-t-\epsilon)\right).$$

Let $a \in [1,4]$ and $h \in [0,1]$. We define the "two-hat" function

$$g_{a,h}(x) = H_{a,0.1}(x) + H_{2.5,0.2+0.1h}(x). \tag{25}$$

Our sampling measure $\gamma_{transport}$ is defined on $\mathcal{M} := \{g_{a,h} : a \in [1,4], h \in [0,1]\}$ by sampling a, h uniformly and then constructing $g_{a,h}$.

Fig. 2(a) plots the singular values in descending order of the input data sampled from $\gamma_{transport}$. The slow decay of the singular values indicates that a non-linear encoder would be a better choice than a linear encoder for this problem. Fig. 2(b) further shows the non-linearity of the data, when we project the data to the top 2 principal components. Fig. 2(c) and 2(d) further show the projections of this data set to the 1st-6th principal components. This data set is nonlinearly parametrized by 2 intrinsic parameters, but the top 2 principal components are not sufficient to represent the data, as shown in Fig. 2(b). Fig. 2(c) shows that the top 3 linear principal components yield a better representation of the data, since the coloring by a and h is well recognized.

We then use the nonlinear Auto-Encoder for a nonlinear dimension reduction of the data. Fig. 3 shows the latent features given by the Auto-Encoder with reduced dimension 2. The intrinsic parameters a and h are well represented in the latent space.

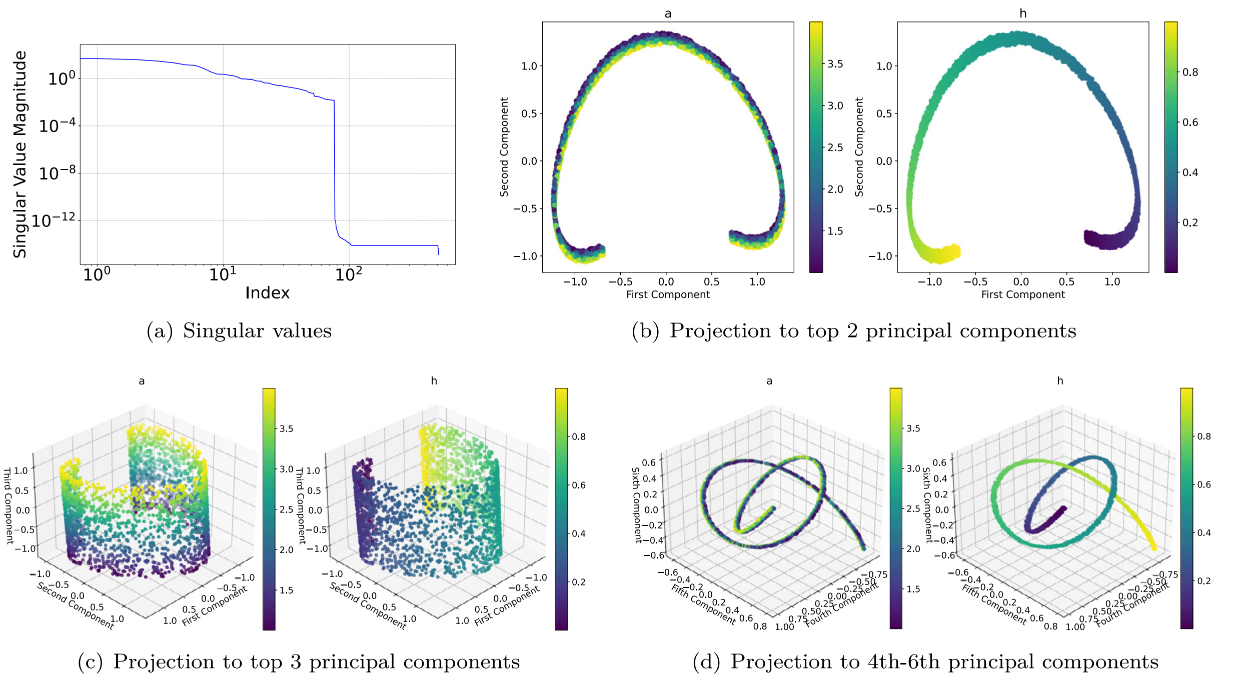


Fig. 2. Nonlinearity of the initial conditions for the **transport** equation. (a) shows the singular values of the data matrix. (b) shows the projection of data to the top 2 principal components and (c) shows the projection to top 3 principal components. (d) shows the projection to the 4th-6th principal components. In (b), (c) and (d), the projections are colored according to the a parameter in the left subplots and according to the h parameter in the right subplots.

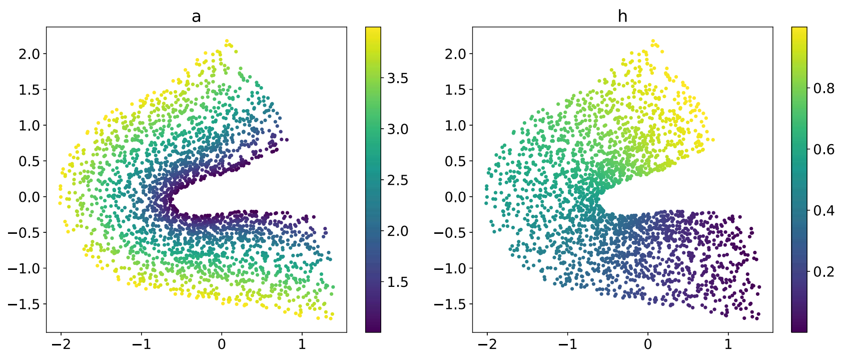


Fig. 3. Latent features of the initial conditions $g_{a,h}$ (Transport) in (25) given by the Auto-Encoder. The left plot is colored according to a and the right plot is colored according to h.

Fig. 4(a) shows a sample $u \sim \gamma_{transport}$, as well as $\Psi(u)$. Before learning the operator Ψ, we compare the projection error of Auto-Encoder and PCA on a test sample from γ in Fig. 4(b). In Fig. 4(b), Auto-Encoder is trained three times with different initilizations, and the average squared test error is shown with standard deviation error bar. Auto-Encoder yields a significantly smaller projection error than PCA for the same reduced dimension. Fig. 4(c) shows the relative test error of AENet, PCANet, and DeepONet (after learning Ψ) as functions of the reduced dimension. We further show the comparison of relative test error (as a percent) in Table 1(a). AENet outperforms PCANet when the reduced dimension is the intrinsic dimension 2, and they are comparable when the reduced dimension is bigger than 2. Finally, Fig. 4(d) shows an example of the predicted solution at $t = 1$ for AENet and PCANet with input reduced dimension 2. See Fig. 20 in Appendix D.1 for more comparisons of the predicted solution from AENet and PCANet.

To validate our theory in Theorem 2, we show a log-log plot of the absolute squared test error versus training sample size in Fig. 4(e) for AENet. The curve is almost linear, depicting the theorized power law in the sample size n. To show robustness to noise, we plot the squared test error versus the variance of Gaussian noise added to the output data in Fig. 4(f), depicting the theorized relationship. In Fig. 4(f), the latent dimension of AENet is taken as 2. In Fig. 4(e) and 4(f), we perform three experimental runs, and show the mean with standard deviation error bar.

5.2. Burgers' equation

We consider the viscous Burgers' equation with periodic boundary conditions given by (for fixed viscosity $\nu = 10^{-3}$)

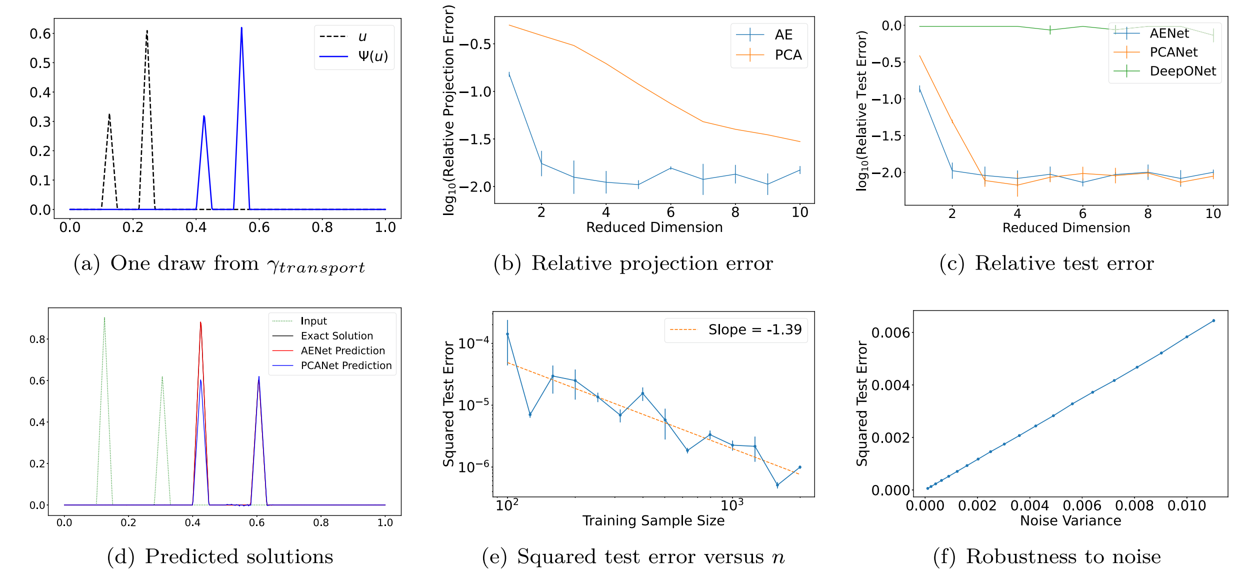


Fig. 4. Results of the transport equation.

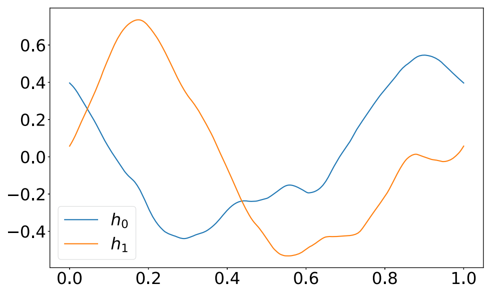


Fig. 5. w_0 and w_1 used for the Burgers' example.

$$u_t = \nu u_{xx} - uu_x, \quad x \in [0,1), t \in (0,1] \tag{26}$$

with a periodic boundary condition and the initial condition $u(x,0) = g(x), x \in [0,1)$. We seek to approximate the operator Ψ which takes $g(x) = u(x,0)$ as input and outputs $u(x,1)$ from the solution of (26) at $t = 1$.

Let w_0 and w_1 be two functions sampled from the probability measure $N\left(0, 7^4(-\frac{d^2}{dx^2} + 7^2 I)^{-2.5}\right)$ on $[0,1)$, which is considered in Bhattacharya et al. [6]. Fig. 5 shows a plot of the b_0 and b_1 used for the results in this section. For any $a \in [-0.9, 0.9]$ and $h \in [0,1]$, define

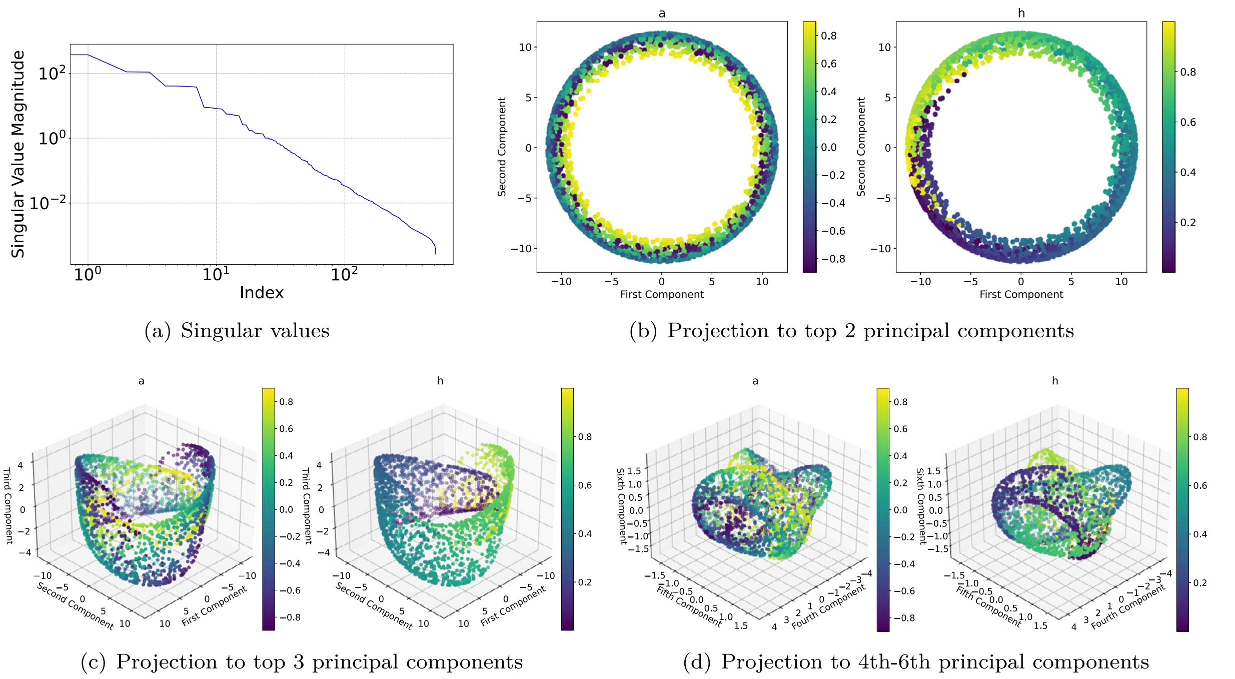
$$g_{a,h}(x) = aw_0(x-h) + \sqrt{1-a^2}w_1(x-h). \tag{27}$$

Our sampling measure γ_{burg} is defined on $\mathcal{M} := \{g_{a,h} : a \in [-0.9, 0.9], h \in [0,1]\}$ by sampling a and h uniformly and then constructing $g_{a,h}$ restricted to $x \in [0,1)$.

Fig. 6(a) plots the singular values in descending order of the input data sampled from γ_{burg}. The slow decay of the singular values indicates that a non-linear encoder would be a better choice than a linear encoder for this problem. Fig. 6(b) shows the non-linearity of the data, when we project the data into the top 2 principal components. Fig. 6(c) and 6(d) further show the projections of this data set to the 1st-6th principal components. This data set is nonlinearly parametrized by 2 intrinsic parameters, but the top 2 principal components are not sufficient to represent the data, as shown in Fig. 6(b).

On the other hand, we can use Auto-Encoder for nonlinear dimension reduction. Fig. 7 shows the projection of the training data by the encoder of a trained Auto-Encoder with reduced dimension 2. The latent parameters reveal the geometry of an annulus, i.e. the Cartesian product of an interval and a circle. This matches the distribution of parameters in $\mathcal{M}$, because the first parameter a varies on a closed interval, and the second parameter b represents translation on the periodic domain which represents a circle.

Fig. 8 contains various plots comparing AENet, PCANet, and DeepONet for the Burgers' equation, analogous to the role Fig. 4 plays for the transport equation. In Fig. 8(c) AENet outperforms PCANet for all reduced dimensions. Fig. 8(d) compares AENet with reduced dimension 2 to PCANet with reduced dimension 2 for domain and 40 for range. See Fig. 21 in Appendix D.1 for more comparisons of


(a) Singular values　(b) Projection to top 2 principal components

(c) Projection to top 3 principal components　(d) Projection to 4th-6th principal components

Fig. 6. Nonlinearity of the initial conditions for the **Burgers'** equation. (a) shows the singular values of the data matrix. (b) shows the projection of data to the top 2 principal components and (c) shows the projection to top 3 principal components. (d) shows the projection to the 4th-6th principal components. In (b), (c) and (d), the projections are colored according to the a parameter in the left subplots and according to the h parameter in the right subplots.

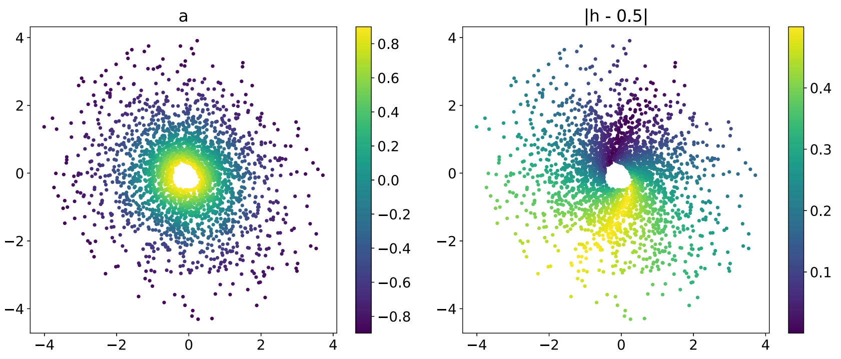


Fig. 7. Latent features of the initial conditions $g_{a,h}$ (Burgers') in (27) given by the Auto-Encoder. The left plot is colored according to a and the right plot is colored according to $|h - 0.5|$.

the predicted solution from AENet and PCANet. Fig. 8(e) and 8(f) display the absolute squared test error. Figs. 8(e) and 8(f) are also generated using AENet with reduced dimension 2. The comparison of relative test error (as a percent) is further shown in Table 1(b).

For the projection error, Auto-Encoder does a much better job than PCA on all three examples. However, for learning the operator, how much information is needed and which information is important depends on the complexity of the PDE. For simple PDEs, such as the transport equation by solving which the initial condition is only shifted, only rough information of the initial condition is sufficient. Even though PCANet has a larger projection error than AENet, it can produce a small operator error as AENet, as demonstrated in Fig. 4(b) and (c). However, solutions of the Burgers' equation have more complicated dynamics. For the Burgers' equation, the operator needs more comprehensive information of the initial condition. In this case, AENet can better extract the intrinsic representation of the initial condition, leading to a smaller operator error.

5.3. Korteweg–De Vries (KdV) equation

We consider one dimensional KdV equation given by

$$u_t = -u_{xxx} - uu_x, \quad x \in [0,6], t \in (0,0.01] \tag{28}$$

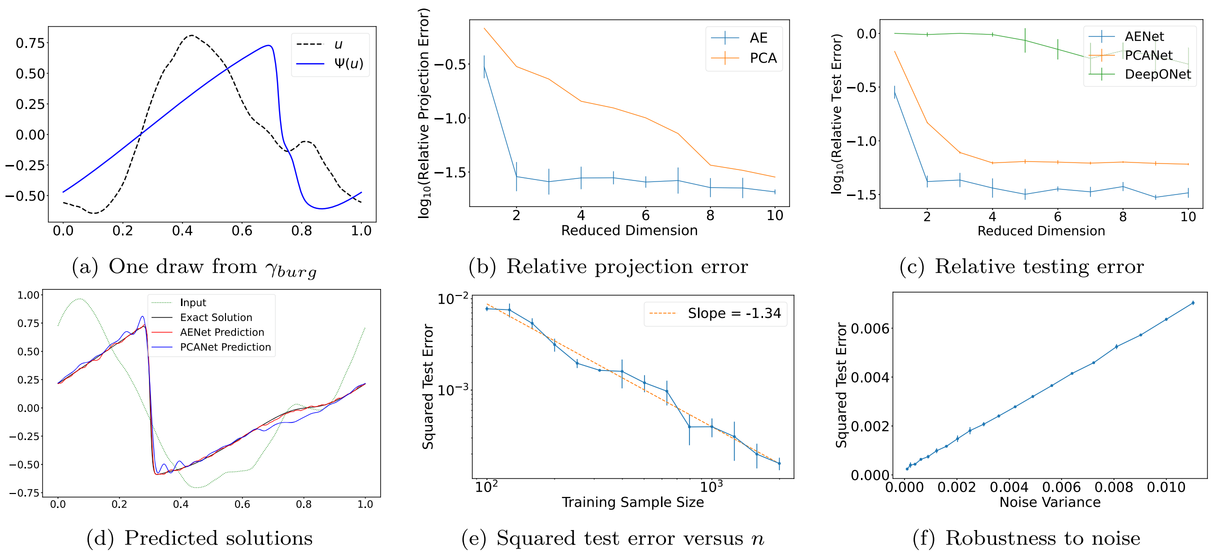


Fig. 8. Results of the Burgers' equation.

with initial condition $u(x,0) = f(x), x \in [0,6]$. We seek to approximate the operator which takes $u(x,0)$ as input and outputs $u(x,0.01)$ from the solution of (28).

For any $a \in [6,18]$ and $h \in [0,3]$, consider the function

$$g_{a,h}(x) = \frac{a^2}{2}\text{sech}\left(\frac{a}{2}(x-1))\right)^2 + \frac{6^2}{2}\ \text{sech}\left(\frac{6}{2}(x-2-h))\right)^2. \tag{29}$$

Our sampling measure γ_{kdv} is defined on

$$\mathcal{M} := \{g_{a,h} : a \in [6,18], h \in [0,3]\} \tag{30}$$

by sampling a and h uniformly and then constructing $g_{a,h}$ restricted to $x \in [0,6]$.

Fig. 9(a) plots the singular values in descending order of the input data sampled from γ_{kdv}. The slow decay of the singular values indicates that a non-linear encoder would be a better choice than a linear encoder for this problem. Fig. 9(b) further shows the non-linearity of the data, when we project the data into the top 2 principal components. Fig. 9(c) and 9(d) show the projections of this data set to the 1st-6th principal components. This data set is nonlinearly parametrized by 2 intrinsic parameters, but the top 2 principal components are not sufficient to represent the data, as shown in Fig. 9(b). Fig. 9(c) shows that the top 3 linear principal components yield a better representation of the data, since the coloring by a and h is well recognized.

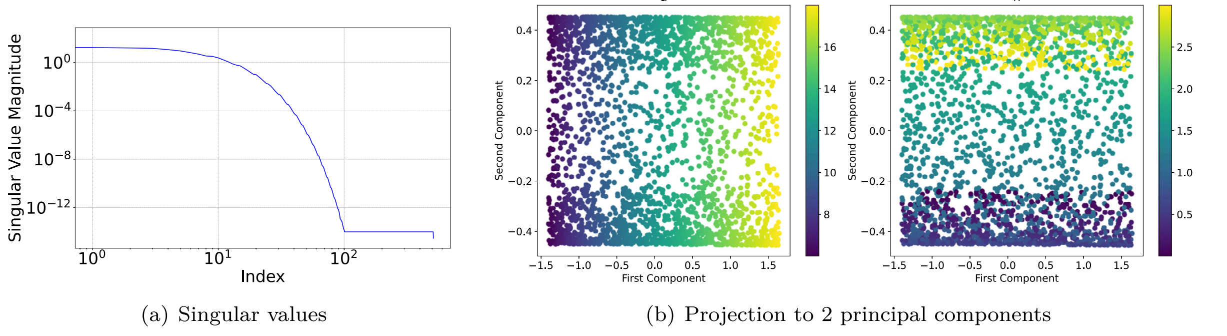
Fig. 10 shows latent parameters of the training data given by the Auto-Encoder with reduced dimension 2. The intrinsic parameters a and h are well represented in the latent space.

Fig. 11 contains various plots comparing AENet, PCANet, and DeepONet for the KdV equation, analogous to the role Fig. 4 plays for the transport equation. Fig. 11(d) compares AENet with reduced dimension 2 to PCANet with reduced dimension 2 for domain and 40 for range. See Fig. 22 in Appendix D.1 for more comparisons of the predicted solution from AENet and PCANet. Fig. 11(e) and 11(f) display the absolute squared test error with respect to n (in log-log plot) and noise variance respectively. Figs. 11(e) and 11(f) are generated using AENet with reduced dimension 2. We further compare the relative test error (as a percent) of AENet, PCANet, and DeepONet in Table 1(c).

5.4. Comparison of relative test error

We compare AENet, PCANet, and DeepONet with various reduced dimensions for the input on the three PDEs mentioned above (using a reduced dimension of 40 for the output of PCANet). We repeat the experiments 3 times and report the mean relative test error along with the standard deviation among the runs in Table 1(a)-(c). DeepONet becomes successful when the reduced dimension is 100 or more.

AENet reaches a stable value for the projection and test error at $k = 2$ reduced dimension. This is because the number of latent variables in the dataset is 2 by construction. We can estimate the intrinsic dimension using state of the art algorithms: Maximum Likelihood Estimation (MLE) [38] and Two Nearest Neighbors (TwoNN) [12]. The estimated dimension of the transport, Burgers, and KdV equation are reported in Table 2. All values in the table are approximately 2, which agrees with the reduced dimension where AENet stabilizes.


(a) Singular values (b) Projection to 2 principal components

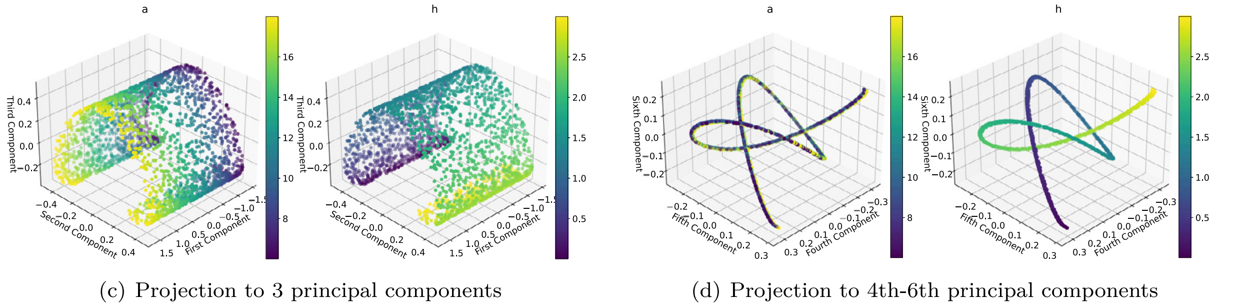

(c) Projection to 3 principal components (d) Projection to 4th-6th principal components

Fig. 9. Nonlinearity of the initial conditions for the **KdV** equation. (a) shows the singular values of the data matrix. (b) shows the projection of data to the top 2 principal components and (c) shows the projection to top 3 principal components. (d) shows the projection to the 4th-6th principal components. In (b), (c) and (d), the projections are colored according to the a parameter in the left subplots and according to the h parameter in the right subplots.

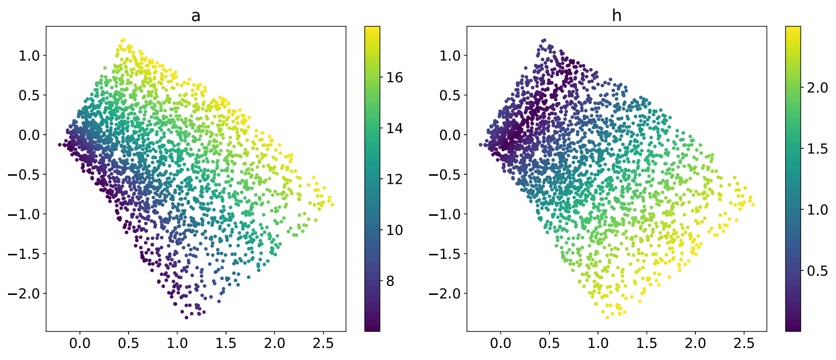


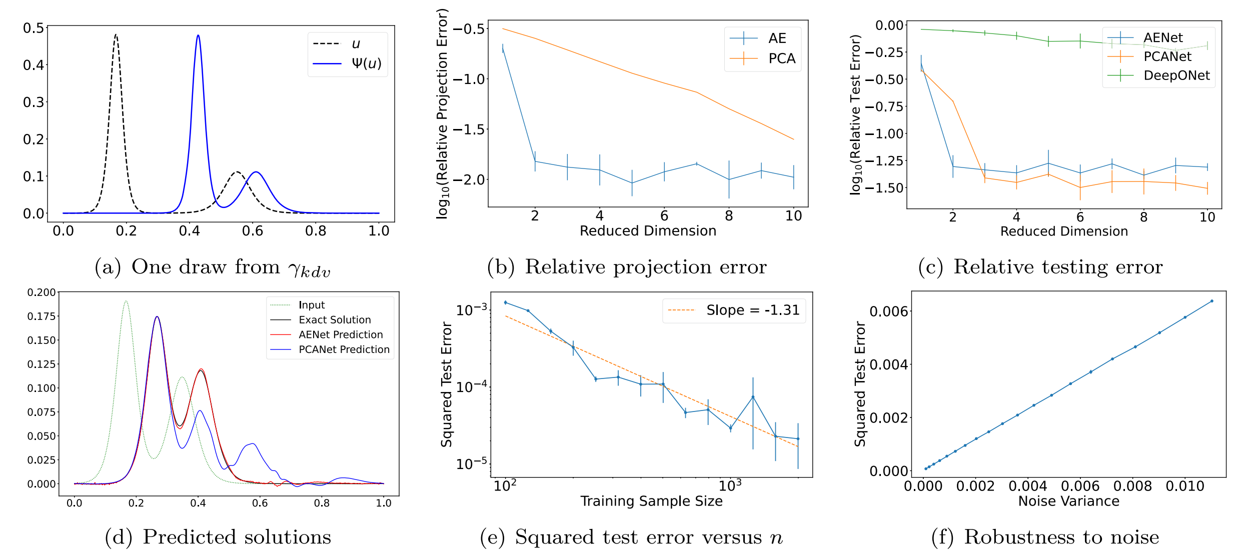
Fig. 10. Latent features of the initial conditions $g_{a,h}$ (KdV) in (29) given by the Auto-Encoder. The left plot is colored according to a and the right plot is colored according to h.

5.5. *Test data on a different grid as the training data*

Finally, we show the robustness of our method when the test data are sampled on a different grid from that of the training data, as shown in Remark 1 and 3. Our training data are sampled on a uniform grid with 256 grid points. When the test data are sampled on a different grid, we interpolate the test data to the same grid as the training data and then evaluate the operator. Fig. 12 shows the squared test error for operator prediction when the test data are sampled on a different grid size for the transport equation in Subsection 5.1, the Burgers' equation in Subsection 5.2 and the KdV equation in Subsection 5.3. We used cubic interpolation for all equations. The operator prediction on test samples by this simple interpolation technique is almost discretization invariant as long as the test samples have a sufficient resolution. For the transport equation, the squared test error is almost the same when the grid size is more than 10^2. For the Burgers' equation and the KdV equation, the squared test error is almost the same when the grid size is more than 10^1 and 10^2, respectively.

5.6. *Ablation study*

The specific architecture for our numerical experiments is described at the beginning of Section 5, but there are many alternative architectures (in terms of network size) that could be used. In this section, we compare the experimental performance of AENet on the examples of the preceding subsections while varying the network width and depth.



(a) One draw from γ_{kdv} (b) Relative projection error (c) Relative testing error

(d) Predicted solutions (e) Squared test error versus n (f) Robustness to noise

Fig. 11. Results of the KdV equation.

Table 1
Comparison table of the relative test error **(as a percent)** when the reduced dimension (in the first row) for the input varies. Each column corresponds to a reduced dimension for the input, except for the last column. The first number is the mean, and in parentheses is the standard deviation among 3 trials. The last column shows time taken to train the model for reduced dimension 2.

(a) Transport equation

Method	1	2	4	6	8	10	20	40	100	Time ($k = 2$)
AENet	12.1 (0.2)	1.1 (0.5)	1.3 (0.7)	1.0 (0.2)	0.9 (0.1)	1.5 (0.4)	0.8 (0.2)	0.9 (0.2)	1.0 (0.1)	284.9 s
PCANet	38.6 (0.0)	5.1 (0.2)	0.9 (0.3)	1.0 (0.2)	1.1 (0.1)	0.9 (0.0)	1.1 (0.2)	0.8 (0.3)	0.7 (0.0)	101.7 s
DeepONet	96.4 (0.0)	96.4 (0.0)	96.4 (0.0)	96.4 (0.0)	96.4 (0.0)	66.0 (6.2)	33.7 (26.0)	18.5 (8.3)	5.0 (1.6)	61.3 s

(b) Burgers' equation

Method	1	2	4	6	8	10	20	40	100	Time ($k = 2$)
AENet	28.6 (4.2)	4.2 (0.6)	3.7 (0.8)	3.6 (0.2)	3.8 (0.4)	3.3 (0.4)	3.1 (0.5)	3.1 (0.2)	3.2 (0.2)	266.0 s
PCANet	68.0 (0.1)	14.7 (0.2)	6.2 (0.2)	6.3 (0.3)	6.3 (0.1)	6.1 (0.1)	6.1 (0.1)	6.5 (0.2)	6.6 (0.3)	101.7 s
DeepONet	100.0 (0.0)	97.5 (4.3)	97.5 (4.3)	72.8 (16.7)	70.1 (12.5)	55.2 (21.0)	31.2 (6.2)	18.2 (2.6)	10.7 (0.9)	61.3 s

(c) KdV equation

Method	1	2	4	6	8	10	20	40	100	Time ($k = 2$)
AENet	51.9 (0.6)	5.7 (0.9)	4.7 (0.5)	5.2 (1.2)	3.9 (0.1)	4.4 (0.7)	4.7 (1.0)	4.8 (0.0)	5.0 (0.3)	282.5 s
PCANet	38.7 (0.0)	19.9 (0.4)	4.0 (0.5)	3.9 (0.4)	4.8 (1.4)	3.4 (0.2)	3.7 (0.2)	4.2 (0.7)	3.8 (0.2)	101.2 s
DeepONet	91.1 (0.1)	91.1 (0.1)	68.9 (3.1)	68.0 (2.2)	62.6 (3.4)	56.0 (2.9)	42.5 (5.2)	20.9 (0.7)	9.1 (0.8)	61.7 s

Table 2
Estimated intrinsic dimension of data used in our experiments by Maximum Likelihood Estimation (MLE) [38] and Two Nearest Neighbors (TwoNN) [12].

	MLE	TwoNN
Transport	2.06	1.99
Burgers	2.00	2.02
KdV	2.08	1.95

Specifically, we construct the encoder, decoder, and latent map "component" networks of AENet using ReLU networks with a fixed latent dimension of 2 and varying width and depth. The number of hidden layers of each component network is 2, 3, or 4 hidden layers, and the width is 100, 200, 300, 400, 500, 600, or 700. We use the same experimental setup as Section 5, and we perform three trials with different random seeds. We report the Auto-Encoder projection error for each architecture and problem in Table 3, and the operator error in Table 4 (in terms of the average relative L^2 error over the trials).

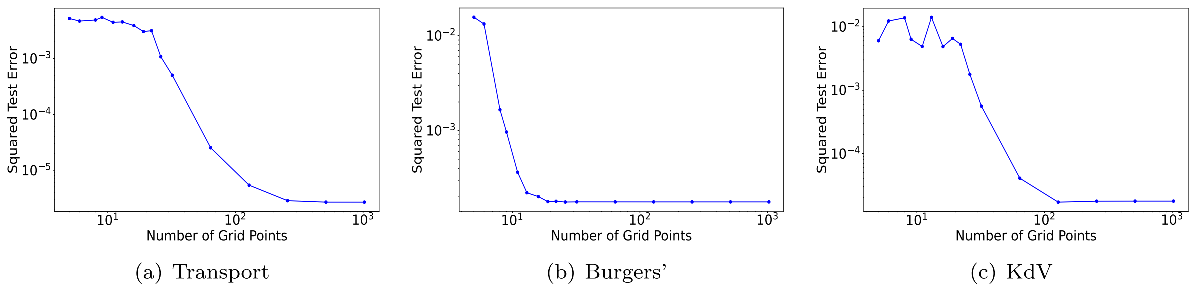


Fig. 12. Squared test error of AENet versus grid size of the test data for the transport equation in Subsection 5.1, the Burgers' equation in Subsection 5.2 and the KdV equation in Subsection 5.3. To evaluate the operator for a test sample, we interpolate the test sample to the same grid as the training data.

Table 3
Relative L^2 Projection Error (as a %) of AENet with various architectures (averaged over 3 trials).

Width	Transport Depth			Burgers' Depth			KdV Depth		
	2	3	4	2	3	4	2	3	4
100	1.49	2.05	1.56	5.06	4.18	5.04	1.50	1.48	1.12
200	2.10	1.95	1.86	2.33	3.67	4.48	1.26	1.23	1.20
300	1.29	1.08	1.72	2.49	4.02	3.33	1.15	1.15	1.67
400	2.04	1.62	1.73	3.77	3.79	3.53	1.98	1.47	1.98
500	1.44	1.86	1.79	3.77	2.98	2.74	1.91	1.55	2.01
600	2.10	1.14	1.71	3.69	3.62	3.85	1.49	1.76	1.85
700	1.66	1.27	1.96	2.98	4.60	4.83	1.95	1.79	1.64

Table 4
Relative L^2 Operator Error (as a %) of AENet with various architectures (averaged over 3 trials).

Width	Transport Depth			Burgers' Depth			KdV Depth		
	2	3	4	2	3	4	2	3	4
100	1.30	1.22	1.09	10.56	7.47	7.43	7.74	7.69	6.79
200	1.50	1.27	1.18	6.64	5.23	5.82	6.58	4.12	5.87
300	1.25	0.90	1.33	5.64	4.65	4.33	6.87	4.40	4.17
400	1.32	1.23	1.11	5.77	4.80	4.31	7.88	4.20	4.48
500	0.92	1.16	1.30	5.87	4.02	4.06	5.13	5.18	5.27
600	1.03	1.22	0.92	5.57	4.87	4.63	6.03	5.69	5.77
700	1.43	0.92	0.95	5.26	5.38	5.83	6.20	6.20	4.56

In addition, Fig. 13 shows the relative projection and operator errors as a function of the number of trainable parameters for all problems, with the starred point corresponding to depth 3 and width 500 networks as used in the preceding subsections (with each trial appearing as a different point). This shows that AENet performs similarly among various architectures, once a sufficiently large number of parameters is picked (as seen by the KdV operator error).

5.7. Comparison to convolutional neural networks

While our theory applies to feedforward neural networks, it is also common to use convolutional neural networks (CNNs) as the building block for Auto-Encoders for scientific computing [37]. In this subsection, we experimentally compare the performance of a modified AENet using CNNs. We employ convolution layers for the encoder in the Auto-Encoder, and we use transposed convolution layers for the decoder in the Auto-Encoder and the output network.

Appendix D.2 provides a definition of convolutional Auto-Encoder. To be brief, we employ a 3 layer convolution network (CNN) with kernel size 8, maximum number of channels 64 which uses max pooling. For the decoder and output networks, we employ a 3 layer transpose convolution network (TCNN) with kernel size 8 and maximum number of channels 64. The encoder CNN has input dimension D and output dimension k, while the decoder and output TCNNs have input dimension k and output dimension D.

The same experimental setup is used as described at the start of this Section 5 (e.g. $D = 512$). The number of trainable parameters in the convolutional AENet is 1,517,514 compared to 1,580,225 for the feedforward AENet which is made up of feedforward networks with depth 3 and width 400. We perform three trials of each experiment, and we report the mean with standard deviation error bars for relevant plots in the rest of this section.

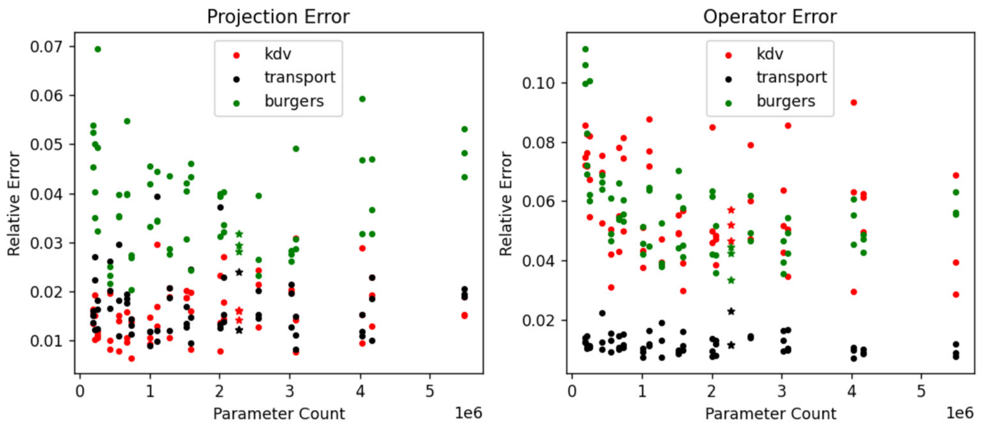


Fig. 13. Performance of AENet with architectures with varying parameter counts (⋆ is the chosen architecture with depth 3 and width 500).

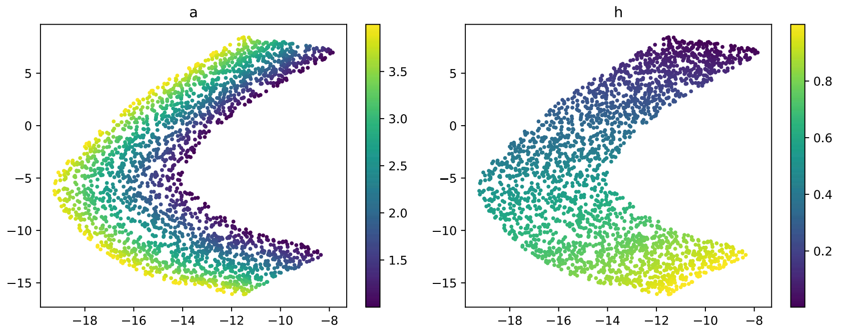
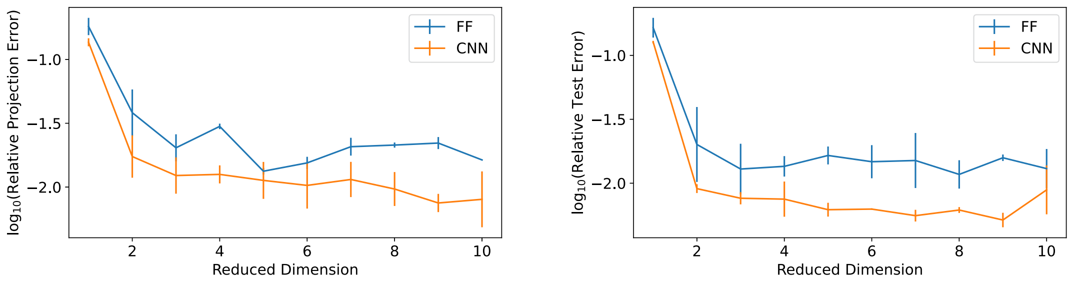


Fig. 14. Latent features of the initial conditions $g_{a,h}$ (Transport) in (25) given by the convolutional Auto-Encoder. The left plot is colored according to a and the right plot is colored according to h.



(a) Relative projection error (b) Relative test error

Fig. 15. Projection and test errors of AENet for transport equation versus reduced dimension.

We consider the same transport equation and data as in Subsection 5.1. Fig. 14 shows the learned latent features given by the convolutional Auto-Encoder with reduced dimension 2, which appears similar to the result for feedforward Auto-Encoder shown in Fig. 3. We further compare the projection and test errors for varying reduced dimensions in Fig. 15.

Next, we consider the same Burgers' equation and data as in Subsection 5.2. Fig. 16 shows the learned latent features given by the convolutional Auto-Encoder with reduced dimension 2, which appears similar to the result for feedforward Auto-Encoder shown in Fig. 7. We further compare the projection and test errors for varying reduced dimensions in Fig. 17.

Finally, we consider the same KdV equation and data as in Subsection 5.3. Fig. 18 shows the learned latent features given by the convolutional Auto-Encoder with reduced dimension 2, which is similar to the result for feedforward Auto-Encoder shown in Fig. 10. We further compare the projection and test errors for varying reduced dimensions in Fig. 19.

In general, the performances of convolutional Auto-Encoder and feedforward Auto-Encoder are similar, while it seems that convolutional Auto-Encoder does a better job in learning the latent feature. We leave a more thorough theoretical study of convolutional architectures for AENet as an interesting future work.

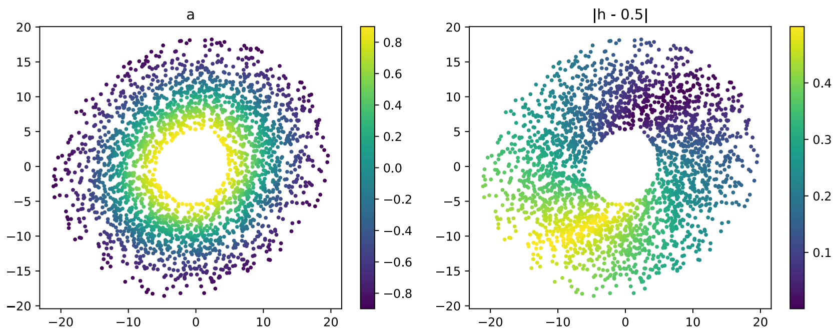


Fig. 16. Latent features of the initial conditions $g_{a,h}$ (Burgers') in (27) given by the convolutional Auto-Encoder. The left plot is colored according to a and the right plot is colored according to $|h - 0.5|$.

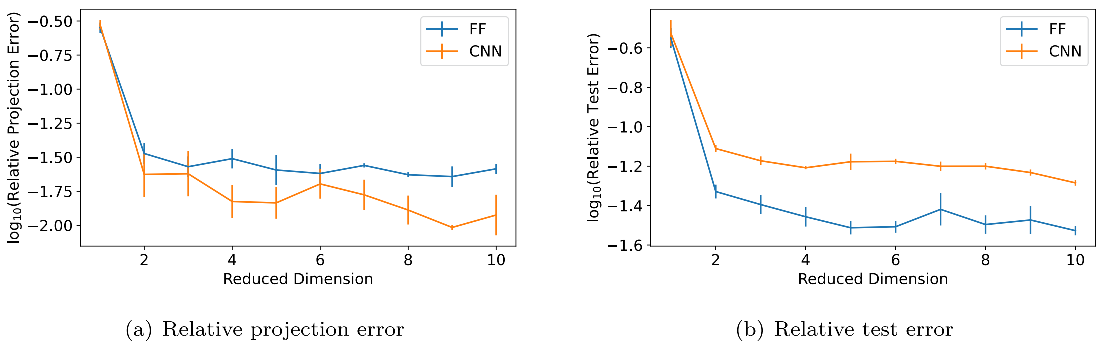


Fig. 17. Projection and test errors of AENet for Burgers' equation versus reduced dimension.

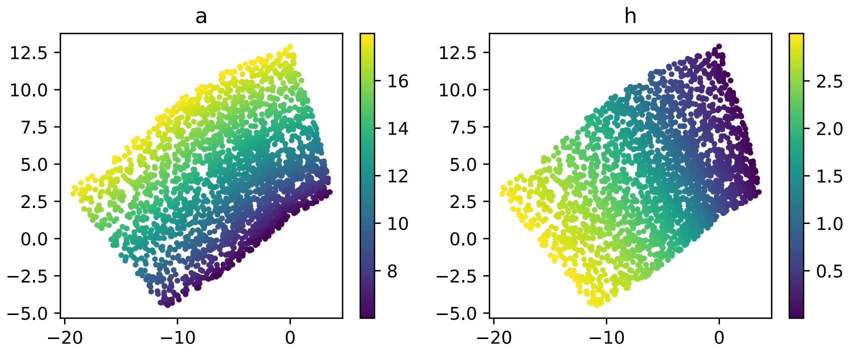


Fig. 18. Latent features of the initial conditions $g_{a,h}$ (KdV) in (29) given by the Auto-Encoder. The left plot is colored according to a and the right plot is colored according to h.

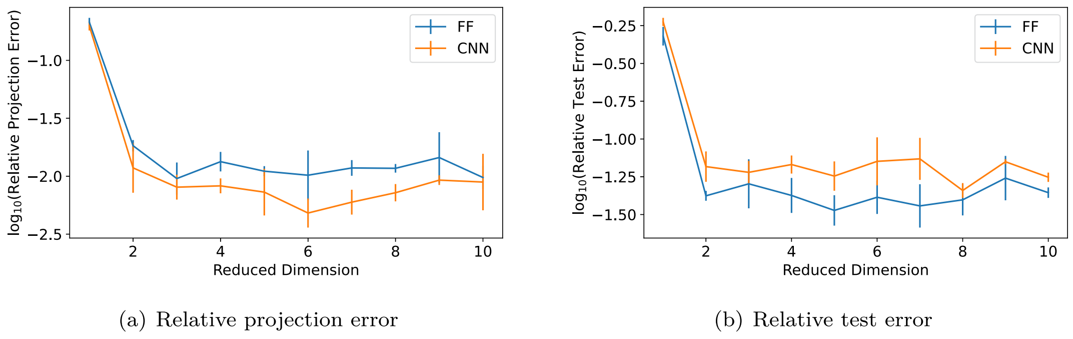


Fig. 19. Projection and test errors of AENet for KdV equation versus reduced dimension.

5.7.1. Discussion

In all examples, AENet with convolutional architecture has similar performance to feedforward AENet. Some example predicted outputs for the three problems are displayed in Appendix D.2. Further experimentation to select the best convolution-based architecture for each component between encoder, decoder, and output is left as a future work.

6. Proof of main results

In this section, we present the proof of our main results: Theorem 1, Corollary 1 and Theorem 2. Proofs of lemmas are given in Appendix B.

6.1. Proof of the approximation theory in Theorem 1

Proof of Theorem 1. We will prove the approximation theory for the Auto-Encoder of the input $\widetilde{u}$ and the transformation Γ in order.
• **Approximation theory of Auto-Encoder for the input $\widetilde{u}$:** We first prove an approximation theory for the Auto-Encoder of $\widetilde{u} = \mathcal{S}_{\mathcal{X}}(u)$. We will show that $\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{g}}$ can be well approximated by neural networks. Note that $\widetilde{\mathbf{f}}$ is only defined on $\widetilde{\mathcal{M}}$. The following lemma shows that it can be extended to the cubical domain $[-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}$ while keeping the same Lipschitz constant.

Lemma 3 *(Kirszbraun theorem [29]). If $E \subset \mathbb{R}^D$, then any Lipschitz function $\mathbf{f} : E \to \mathbb{R}^d$ can be extended to the whole $\mathbb{R}^D$ keeping the Lipschitz constant of the original function.*

In the rest of this paper, without other specification, we use $\widetilde{\mathbf{f}}$ to denote the extended function.

For the network construction to approximate $\widetilde{\mathbf{f}}$, we will use the following neural network approximation result on a set in $\mathbb{R}^D$ with Minkowski dimension $d < D$, which is a variant of Nakada and Imaizumi [48, Theorem 5] (see a proof in Appendix B.3):

Lemma 4. *Let D, d be positive integers with $d < D$, $M > 0$ and $\Xi \subset [0,1]^D$. Suppose $d \geq d_M \Xi$. For any $\varepsilon > 0$, consider a network class $\mathcal{F}_{\mathrm{NN}}(D, 1, L, p, K, \kappa, M)$ with*

$$L = O(\log \varepsilon^{-1}),\ p = O(\varepsilon^{-d}),\ K = O(\varepsilon^{-d}),\ \kappa = O(\varepsilon^{-3-4(1+\lceil \log_2 M \rceil}).$$

Then for $\varepsilon \in (0, 1/4)$ and any Lipschitz function $f^* : [0,1]^D \to [-M, M]$ with function value and Lipschitz constant bounded by M, there exists a network $f_{\mathrm{NN}} \in \mathcal{F}_{\mathrm{NN}}(D, 1, L, p, K, \kappa, M)$ satisfying

$$\|f_{\mathrm{NN}} - f^*\|_{L^\infty(\Xi)} \leq \varepsilon.$$

The constant hidden in $O(\cdot)$ depends on d, M and is only polynomial in D.

To use Lemma 4 to derive an approximation result of $\widetilde{\mathbf{f}}$, we need an upper bound on $\sup_{\widetilde{u} \in [-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}} \|\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty$. Note that $\widetilde{\mathbf{f}}$ is the extended function from $\widetilde{\mathcal{M}}$ to $[-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}$. Even though $\|\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty$ is bounded by 1 for $\widetilde{u} \in \widetilde{\mathcal{M}}$, $\|\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty$ may exceed 1 for $\widetilde{u} \in [-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}$. Since $\widetilde{\mathbf{f}}(\widetilde{u}) \in [-1,1]^d$ for any $\widetilde{u} \in \widetilde{\mathcal{M}}$ and we are building the approximation theory of $\widetilde{\mathbf{f}}$ on $\widetilde{\mathcal{M}}$, we can clip the value of $\widetilde{\mathbf{f}}(\widetilde{u})$ for $\widetilde{u} \in [-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}$ to $[-1,1]^d$. Specifically, we introduce the clipping operator

$$\mathrm{CL}(\widetilde{\mathbf{f}}) = \min\{\max\{\widetilde{\mathbf{f}}, -1\}, 1\}, \tag{31}$$

where min, max are applied element-wisely. The operator $\mathrm{CL}(\widetilde{\mathbf{f}})$ clips the outputs of $\widetilde{\mathbf{f}}$ to $[-1,1]^d$. It is easy to show that the functions $\mathrm{CL}(\widetilde{\mathbf{f}})$ are Lipschitz with the same Lipschitz constant as $\widetilde{\mathbf{f}}$.

Now, we are ready to conduct approximation analysis on $\widetilde{\mathbf{f}} : \widetilde{\mathcal{M}} \to [-1,1]^d$. Denote $\widetilde{\mathbf{f}} = [\widetilde{f}_1, ..., \widetilde{f}_d]^\top$. For $k = 1, ..., d$, by Lemma 1, each $\mathrm{CL}(\widetilde{f}_k)$ is a function from $[-R_{\mathcal{X}}, R_{\mathcal{X}}]^{D_1}$ to $[-1,1]$ and with Lipschitz constant $2L_{\mathbf{f}}$. According to Lemma 2, we have $d_M \mathcal{S}_{\mathcal{X}}(\mathcal{M}) \leq d$. For any $\varepsilon_1 > 0$, by Lemma 4 with a proper scaling and shifting, there exists a network architecture $\mathcal{F}_{\mathrm{NN}}(D_1, 1, L_4, p_4, K_4, \kappa_4, M_4)$ with

$$L_4 = O(\log \varepsilon^{-1}),\ p_4 = O(\varepsilon_1^{-d}),\ K_4 = O(\varepsilon_1^{-d}),\ \kappa_4 = O(\varepsilon_1^{-7}),\ M_4 = 1$$

so that for each $\widetilde{f}_k$, there exists $\widetilde{f}_{\mathrm{NN},k} \in \mathcal{F}_{\mathrm{NN}}(D_1, 1, L_4, p_4, K_4, \kappa_4, M_4)$ satisfying

$$\|\widetilde{f}_{\mathrm{NN},k} - \mathrm{CL}(\widetilde{f}_k)\|_{L^\infty(\widetilde{\mathcal{M}})} \leq \varepsilon.$$

The constant hidden in $O(\cdot)$ depends on $d, L_{\mathbf{f}}, R_{\mathcal{X}}$ and polynomially in D_1.

Define the network $\widetilde{\mathbf{f}}_{\mathrm{NN}} = [\widetilde{f}_{\mathrm{NN},1}, ..., \widetilde{f}_{\mathrm{NN},d}]^\top$ as the concatenation of $\widetilde{f}_{\mathrm{NN},k}$'s. We have

$$\sup_{\widetilde{u} \in \widetilde{\mathcal{M}}} \|\widetilde{\mathbf{f}}_{\mathrm{NN}}(\widetilde{u}) - \widetilde{\mathbf{f}}(\widetilde{u})\|_\infty = \sup_{\widetilde{u} \in \widetilde{\mathcal{M}}} \|\widetilde{\mathbf{f}}_{\mathrm{NN}}(\widetilde{u}) - \mathrm{CL} \circ \widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \leq \varepsilon_1$$

since $\mathrm{CL} \circ \widetilde{\mathbf{f}}(\widetilde{u}) = \widetilde{\mathbf{f}}(\widetilde{u}) \in [-1,1]^d$ for any $u \in \widetilde{\mathcal{M}}$.

Furthermore, we have $\widetilde{\mathbf{f}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{E_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(D_1, d, L_1, p_1, K_1, \kappa_1, M_1)$ with

$$L_1 = L_4 = O(\log \varepsilon^{-1}),\ p_1 = dp_4 = O(\varepsilon_1^{-d}),\ K_1 = dK_4 = O(\varepsilon_1^{-d}),\ \kappa_1 = \kappa_4 = O(\varepsilon_1^{-7}),\ M_1 = 1.$$

For the network approximation of $\widetilde{\mathbf{g}}$, we will use the following result:

Lemma 5 *(Theorem 1 of Yarotsky [67]). For any $\varepsilon \in (0,1)$, there is a network architecture $\mathcal{F}_{\rm NN}(D, 1, L, p, K, \kappa, M)$ with*

$$L = O(\log \varepsilon^{-1}),\ p = O(\varepsilon^{-D}),\ K = O(\varepsilon^{-D} \log \varepsilon^{-1}),\ \kappa = O(\varepsilon^{-1}),\ M = 1$$

so that for any Lipschitz function $f^* : [0,1]^D \to [-1,1]$ with Lipschitz constant bounded by 1, there exists a network $f_{\rm NN} \in \mathcal{F}_{\rm NN}(D, 1, L, p, K, \kappa, M)$ satisfying

$$\|f_{\rm NN} - f^*\|_{L^\infty([0,1]^D)} \leq \varepsilon.$$

The constants hidden in $O(\cdot)$ depend on D.

Denote $\widetilde{\mathbf{g}} = [\widetilde{g}_1, ..., \widetilde{g}_{D_1}]$. By Lemma 1, each $\widetilde{g}_k$ is Lipschitz with Lipschitz constant $2L_{\mathbf{g}}$. By Lemma 5 with a proper scaling and shifting, for any $\varepsilon_2 \in (0,1)$, there exists a network architecture $\mathcal{F}_{\rm NN}(d, 1, L_5, p_5, K_5, \kappa_5, M_5)$ with

$$L_5 = O(\log(1/\varepsilon_2)),\ p_5 = O(\varepsilon_2^{-d}),\ K_5 = O(\varepsilon_2^{-d} \log(1/\varepsilon_2)),\ \kappa_5 = O(\varepsilon_2^{-1}),\ M_5 = R_{\mathcal{X}}$$

so that for each $g_{1,k}$, there is a $\widetilde{g}_{{\rm NN},k} \in \mathcal{F}_{\rm NN}(d, 1, L_5, p_5, K_5, \kappa_5, M_5)$ satisfying

$$\|\widetilde{g}_{{\rm NN},k} - \widetilde{g}_k\|_{L^\infty([-1,1]^d)} \leq \varepsilon_2. \tag{32}$$

The constant hidden in $O(\cdot)$ depends on $d, L_{\mathbf{g}}$ and $R_{\mathcal{X}}$.

Define $\widetilde{\mathbf{g}} = [\widetilde{g}_{{\rm NN},1}, ..., \widetilde{g}_{{\rm NN},D_1}]^\top$ as the concatenation of $\widetilde{g}_k$'s. According to (32), we have

$$\sup_{\mathbf{z} \in [-1,1]^d} \|\widetilde{\mathbf{g}}_{\rm NN}(\mathbf{z}) - \widetilde{\mathbf{g}}(\mathbf{z})\|_\infty \leq \varepsilon_2,$$

and $\widetilde{\mathbf{g}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{D_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(d, D_1, L_2, p_2, K_2, \kappa_2, M_2)$ with

$$L_2 = L_5 = O(\log(1/\varepsilon_2)),\ p_2 = D_1 p_5 = O(\varepsilon_2^{-d}),$$
$$K_2 = D_1 K_5 = O(\varepsilon_2^{-d} \log(1/\varepsilon_2)),\ \kappa_2 = \kappa_5 = O(\varepsilon_2^{-1}),\ M_2 = R_{\mathcal{X}}.$$

The constant hidden in $O(\cdot)$ depends on $d, L_{\mathbf{g}}, R_{\mathcal{X}}$ and is linear in D_1.

As a result, we have the following for any $u \in \mathcal{M}$:

$$\begin{aligned}
&\|\widetilde{\mathbf{g}}_{\rm NN} \circ \widetilde{\mathbf{f}}_{\rm NN} \circ S_{\mathcal{X}}(u) - \widetilde{\mathbf{g}} \circ \widetilde{\mathbf{f}} \circ S_{\mathcal{X}}(u)\|_\infty \\
\leq &\|\widetilde{\mathbf{g}}_{\rm NN} \circ \widetilde{\mathbf{f}}_{\rm NN} \circ S_{\mathcal{X}}(u) - \widetilde{\mathbf{g}} \circ \widetilde{\mathbf{f}}_{\rm NN} \circ S_{\mathcal{X}}(u)\|_\infty + \|\widetilde{\mathbf{g}} \circ \widetilde{\mathbf{f}}_{\rm NN} \circ S_{\mathcal{X}}(u) - \widetilde{\mathbf{g}} \circ \widetilde{\mathbf{f}} \circ S_{\mathcal{X}}(u)\|_\infty \\
\leq &\varepsilon_2 + 2\sqrt{d} L_{\mathbf{g}} \varepsilon_1.
\end{aligned} \tag{33}$$

• **Approximation theory for the transformation Γ from the input latent variable to the output:** The network approximation of the transformation Γ can be proved using Lemma 5. First, we notice that the operator Γ is Lipschitz (see a proof in Appendix B.4).

Lemma 6. *The operator Γ defined in (16) is Lipschitz with Lipschitz constant $2L_\Psi L_{\mathbf{g}}$:*

$$\|\Gamma(\mathbf{z}_1) - \Gamma(\mathbf{z}_2)\|_{S_{\mathcal{Y}}} \leq 2L_\Psi L_{\mathbf{g}} \|\mathbf{z}_1 - \mathbf{z}_2\|_2 \tag{34}$$

for any $\mathbf{z}_1, \mathbf{z}_2 \in [-1,1]^d$.

Denote $\Gamma = [\Gamma_1, ..., \Gamma_{D_2}]^\top$. According to Lemma 6, each Γ_k is Lipschitz with Lipschitz constant $2L_\Psi L_{\mathbf{g}}$. By Lemma 5 with proper scaling and shifting, for any $\varepsilon_3 \in (0,1)$, there exists a network architecture $\mathcal{F}_{\rm NN}(d, 1, L_6, p_6, K_6, \kappa_6, M_6)$ with

$$L_6 = O(\log(1/\varepsilon_3)),\ p_6 = O(\varepsilon_3^{-d}),\ K_6 = O(\varepsilon_3^{-d} \log(1/\varepsilon_3)),\ \kappa_6 = O(\varepsilon_3^{-1}), M_6 = R_{\mathcal{Y}}$$

so that for each Γ_k, there is a $\Gamma_{{\rm NN},k} \in \mathcal{F}_{\rm NN}(d, 1, L_6, p_6, K_6, \kappa_6, M_6)$ satisfying

$$\|\Gamma_{{\rm NN},k} - \Gamma_k\|_{L^\infty((-1,1]^d)} \leq \varepsilon_3.$$

The constant hidden in $O(\cdot)$ depends on $d, L_{\mathbf{g}}, L_\Psi$ and $R_{\mathcal{Y}}$.

Define the network $\Gamma_{\rm NN} = [\Gamma_{{\rm NN},1}, ..., \Gamma_{{\rm NN},d}]^\top$ as the concatenation of the $\Gamma_{{\rm NN},k}$'s. Then we have

$$\sup_{\mathbf{z}\in[-1,1]^d} \|\Gamma_{\rm NN}(\mathbf{z}) - \Gamma(\mathbf{z})\|_\infty \le \varepsilon_3.$$

Furthermore, we have $\Gamma_{\rm NN} \in \mathcal{F}_{\rm NN}^{\Gamma} = \mathcal{F}_{\rm NN}(d, D_2, L_3, p_3, K_3, \kappa_3, M_3)$ with

$$L_3 = L_6 = O(\log(1/\varepsilon_3)),\ p_3 = dp_6 = O(\varepsilon_3^{-d}),$$
$$K_3 = dK_6 = O(\varepsilon_3^{-d}\log(1/\varepsilon_3)),\ \kappa_3 = \kappa_6 = O(\varepsilon_3^{-1}),\ M_3 = R_{\mathcal{Y}}.$$

The constant hidden in $O(\cdot)$ depends on $d, L_{\mathbf{g}}, L_\Psi, R_{\mathcal{Y}}$ and is linear in D_2. $\square$

6.2. *Proof of Corollary 1*

Proof of Corollary 1. For any $\widetilde{u} \in \widetilde{\mathcal{M}}$, we have

$$\begin{aligned}\|\Phi_{\rm NN}(\widetilde{u}) - \Phi(\widetilde{u})\|_\infty =& \|\Gamma_{\rm NN}\circ\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \Gamma\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \\ \le& \|\Gamma_{\rm NN}\circ\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \Gamma\circ\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u})\|_\infty + \|\Gamma\circ\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \Gamma\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \\ \le& \|\Gamma_{\rm NN} - \Gamma\|_\infty + 2L_\Psi L_{\mathbf{g}}\|\widetilde{\mathbf{f}}_{\rm NN} - \widetilde{\mathbf{f}}\|_\infty \le \varepsilon_3 + 2\sqrt{d}L_\Psi L_{\mathbf{g}}\varepsilon_1 = \varepsilon. \quad \square\end{aligned}$$

6.3. *Proof of the generalization theory in Theorem 2*

In this section, we first give an upper bound of the generalization error of Stage I in Section 6.3.1. The generalization error combining Stage I and II is analyzed in Section 6.3.2.

6.3.1. *An upper bound for the generalization error in Stage I*

In Stage I, the encoder $E_{\mathcal{X}}^n$ and decoder $D_{\mathcal{X}}^n$ are learned based on the first half data $\mathcal{J}_1$. We expect that $D_{\mathcal{X}}^n \circ E_{\mathcal{X}}^n(\widetilde{u})$ is close to $\widetilde{u}$ for any $\widetilde{u} \in \widetilde{\mathcal{M}}$. We study the generalization error

$$\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\|D_{\mathcal{X}}^n \circ E_{\mathcal{X}}^n(\widetilde{u}) - \widetilde{u}\|_\infty^2. \tag{35}$$

Let $\mathcal{F} = \mathcal{F}_{\rm NN}(d_1, d_2, L, p, K, \kappa, M)$ be a network class from $[-B, B]^{d_1}$ to $[-R, R]^{d_2}$ for some $B, R > 0$. We denote $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_{L^{\infty,\infty}})$ as the δ-covering number of $\mathcal{F}$, where the norm $\|\cdot\|_{L^{\infty,\infty}}$ is defined as $\|F_{\rm NN}\|_{L^{\infty,\infty}} = \sup_{\mathbf{x}\in[-B,B]^{d_1}} \|F_{\rm NN}(\mathbf{x})\|_\infty$ for any $F_{\rm NN} \in \mathcal{F}$.

An upper bound of the generalization error in (35) is given in the following lemma (see a proof in Appendix B.5).

Lemma 7. *Suppose Assumption 1 and 2 hold. For any $\varepsilon_1 \in (0, 1/4), \varepsilon_2 \in (0, 1)$, set the network architectures $\mathcal{F}_{\rm NN}^{E_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(D_1, d, L_1, p_1, K_1, \kappa_1, M_1)$ with*

$$L_1 = O(\log \varepsilon_1^{-1}),\ p_1 = O(\varepsilon_1^{-d}),\ K_1 = O(\varepsilon_1^{-d}),\ \kappa_1 = O(\varepsilon_1^{-7}),\ M_1 = 1,$$

and $\mathcal{F}_{\rm NN}^{D_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(d, D_1, L_2, p_2, K_2, \kappa_2, M_2)$ with

$$L_2 = O(\log \varepsilon_2^{-1}),\ p_2 = O(\varepsilon_2^{-d}),\ K_2 = O(\varepsilon_2^{-d}\log \varepsilon_2^{-1}),\ \kappa_2 = O(\varepsilon_2^{-1}),\ M_2 = R_{\mathcal{X}}.$$

Denote the network architecture

$$\mathcal{F}_{\rm NN}^{G} = \{G_{\rm NN} : \mathbb{R}^{D_1} \to \mathbb{R}^{D_1} | G_{\rm NN} = D_{\rm NN}\circ E_{\rm NN}\ \textit{for}\ D_{\rm NN} \in \mathcal{F}_{\rm NN}^{D_{\mathcal{X}}},\ E_{\rm NN} \in \mathcal{F}_{\rm NN}^{E_{\mathcal{X}}}\}.$$

Let $E_{\mathcal{X}}^n, D_{\mathcal{X}}^n$ be the learned autoencoder in Stage I given by (17). For any $\delta > 0$, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\left[\|D_{\mathcal{X}}^n \circ E_{\mathcal{X}}^n(\widetilde{u}) - \widetilde{u}\|_\infty^2\right] \le& 16dL_{\mathbf{g}}^2\varepsilon_1^2 + 4\varepsilon_2^2 \\ &+ \frac{48R_{\mathcal{X}}^2}{n}\log\mathcal{N}\left(\frac{\delta}{4R_{\mathcal{X}}}, \mathcal{F}_{\rm NN}^{G}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta.\end{aligned} \tag{36}$$

In Lemma 7, $\varepsilon_1, \varepsilon_2$ correspond to the approximation error of $\mathcal{F}_{\rm NN}^{E_{\mathcal{X}}}$, $\mathcal{F}_{\rm NN}^{D_{\mathcal{X}}}$ in Theorem 1, respectively. For any $\varepsilon \in (0, 1/4)$, by Theorem 1, we choose $\varepsilon_1 = \varepsilon_2 = \varepsilon$ and set the network architectures $\mathcal{F}_{\rm NN}^{E_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(D_1, d, L_1, p_1, K_1, \kappa_1, M_1)$ with

$$L_1 = O(\log \varepsilon^{-1}),\ p_1 = O(\varepsilon^{-d}),\ K_1 = O(\varepsilon^{-d}),\ \kappa_1 = O(\varepsilon^{-7}),\ M_1 = 1, \tag{37}$$

and $\mathcal{F}_{\rm NN}^{D_{\mathcal{X}}} = \mathcal{F}_{\rm NN}(d, D_1, L_2, p_2, K_2, \kappa_2, M_2)$ with

$$L_2 = O(\log(1/\varepsilon)),\ p_2 = O(\varepsilon^{-d}),\ K_2 = O(\varepsilon^{-d}\log(1/\varepsilon)),\ \kappa_2 = O(\varepsilon^{-1}),\ M_2 = R_{\mathcal{X}}. \tag{38}$$

There exist $\widetilde{\mathbf{f}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{E_\mathcal{X}}, \widetilde{\mathbf{g}}_{\rm NN} \in \mathcal{F}_{\rm NN}^{D_\mathcal{X}}$ satisfying

$$\sup_{\widetilde{u}\in\widetilde{\mathcal{M}}} \|\widetilde{\mathbf{f}}_{\rm NN}(\widetilde{u}) - \widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \le \varepsilon, \qquad \sup_{\mathbf{z}\in[-1,1]^d} \|\widetilde{\mathbf{g}}_{\rm NN}(\mathbf{z}) - \widetilde{\mathbf{g}}(\mathbf{z})\|_\infty \le \varepsilon.$$

The constant hidden in $O(\cdot)$ depends on $d, L_\mathbf{f}, L_\mathbf{g}, R_\mathcal{X}$ and is polynomial in D_1.

By Lemma 7, we have

$$\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\left[\|D_\mathcal{X}^n\circ E_\mathcal{X}^n(\widetilde{u}) - \widetilde{u}\|_\infty^2\right] \le (16dL_\mathbf{g}^2+4)\varepsilon^2 + \frac{48R_\mathcal{X}^2}{n}\log\mathcal{N}\left(\frac{\delta}{4R_\mathcal{X}}, \mathcal{F}_{\rm NN}^G, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \tag{39}$$

The following lemma gives an upper bound of the covering number of any given network architecture:

Lemma 8 *(Lemma 5.3 of Chen et al. [9]). Let $\mathcal{F}_{\rm NN}(d_1, d_2, L, p, K, \kappa, M)$ be a network architecture from $[-B,B]^{d_1}$ to $[-R,R]^{d_2}$ for some $B, R > 0$. We have*

$$\mathcal{N}(\delta, \mathcal{F}_{\rm NN}, \|\cdot\|_{L^{\infty,\infty}}) \le \left(\frac{2L^2(pB+2)\kappa^L p^{L+1}}{\delta}\right)^{d_2K}.$$

According to the definition of $\mathcal{F}_{\rm NN}^G$, we have $\mathcal{F}_{\rm NN}^G \subset \mathcal{F}_{\rm NN}(D_1, D_1, L_4, p_4, K_4, \kappa_4, M_4)$ with

$$L_4 = O(\log\varepsilon^{-1}),\ p_4 = O(\varepsilon^{-d}),\ K_4 = O(\varepsilon^{-d}\log\varepsilon^{-1}),\ \kappa_4 = O(\varepsilon^{-7}),\ M_4 = R_\mathcal{X}. \tag{40}$$

Substituting (40) into Lemma 8, we get

$$\log\mathcal{N}\left(\frac{\delta}{4R_\mathcal{X}}, \mathcal{F}_{\rm NN}^G, \|\cdot\|_{L^{\infty,\infty}}\right) \le C_1\varepsilon^{-d}\log^2\varepsilon^{-1}(\log\varepsilon^{-1} + \log\delta^{-1}), \tag{41}$$

for some C_1 depending on $d, L_\mathbf{f}, L_\mathbf{g}, R_\mathcal{X}, |\Omega_\mathcal{X}|$, and is polynomial in D_1.

Substituting (41) into (39) gives rise to

$$\begin{aligned}&\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\|D_\mathcal{X}^n\circ E_\mathcal{X}^n\circ S_\mathcal{X}(u) - S_\mathcal{X}(u)\|_\infty^2\\ \le&(16dL_\mathbf{g}^2+4)\varepsilon^2 + \frac{C_1 48R_\mathcal{X}^2}{n}\varepsilon^{-d}\log^2\frac{1}{\varepsilon}(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta} + D_1) + 6\delta.\end{aligned} \tag{42}$$

By balancing the terms in (42), we set $\varepsilon = n^{-\frac{1}{2+d}}, \delta = \frac{1}{n}$. The bound in (42) reduces to

$$\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\|D_\mathcal{X}^n\circ E_\mathcal{X}^n\circ S_\mathcal{X}(u) - S_\mathcal{X}(u)\|_\infty^2 \le C_2 n^{-\frac{2}{2+d}}\log^3 n$$

for some C_2 depending on $d, L_\mathbf{f}, L_\mathbf{g}, R_\mathcal{X}$ and is polynomial in D_1.

By Markov inequality, we further have the probability bound

$$\mathbb{P}(\|D_\mathcal{X}^n\circ E_\mathcal{X}^n\circ S_\mathcal{X}(u) - S_\mathcal{X}(u)\|_\infty^2 \ge t) \le \frac{\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\|D_\mathcal{X}^n\circ E_\mathcal{X}^n\circ S_\mathcal{X}(u) - S_\mathcal{X}(u)\|_\infty^2}{t} \le \frac{C_2 n^{-\frac{2}{2+d}}\log^3 n}{t} \tag{43}$$

for $t > 0$ and $u \sim \gamma$.

6.3.2. Proof of Theorem 2

Proof of Theorem 2. Recall that the dataset is evenly splitted into $\mathcal{J}_1$ and $\mathcal{J}_2$, which are used in Stage I and Stage II, respectively. We decompose the error as

$$\begin{aligned}&\mathbb{E}_\mathcal{J}\mathbb{E}_{u\sim\gamma}\left[\|\Phi_{\rm NN}^n\circ S_\mathcal{X}(u) - S_\mathcal{Y}\circ\Psi(u)\|_{S_\mathcal{Y}}^2\right]\\ =&\mathbb{E}_\mathcal{J}\mathbb{E}_{u\sim\gamma}\left[\|\Phi_{\rm NN}^n(\widetilde{u}) - \widetilde{v}\|_{S_\mathcal{Y}}^2\right]\\ =&\underbrace{\mathbb{E}_{\mathcal{J}_1}\left[2\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\|\Phi_{\rm NN}^n(\widetilde{u})_i - \widetilde{v}_i\|_{S_\mathcal{Y}}^2\middle|\mathcal{J}_1\right]\right]}_{\rm T_1}\end{aligned}$$

$$
\underbrace{+\mathbb{E}_{\mathcal{J}_1}\left[\mathbb{E}_{\mathcal{J}_2}\mathbb{E}_{u\sim\gamma}\left[\|\Phi^n_{\rm NN}(\widetilde{u})-\widetilde{v}\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]-2\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\|\Phi^n_{\rm NN}(\widetilde{u}_i)-\widetilde{v}_i\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\right]}_{\mathrm{T}_2}. \tag{44}
$$

The term T_1 captures the bias of network approximation. The term T_2 captures the variance. We will derive the upper bound for each term in the rest of the proof.

Bounding T_1**.** We deduce

$$
\begin{aligned}
\mathrm{T}_1 =&\mathbb{E}_{\mathcal{J}_1}\left[2\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\|\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}\circ\Psi(u_i)\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\right]\\
=&\mathbb{E}_{\mathcal{J}_1}\left[2\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\|\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}\circ\Psi(u_i)-\mathcal{S}_\mathcal{Y}(\epsilon_i)+\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\right]\\
=&\mathbb{E}_{\mathcal{J}_1}\left[\frac{2}{n}\mathbb{E}_{\mathcal{J}_2}\left[\sum_{i=n+1}^{2n}\|\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(\widehat{v}_i)+\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\right]\\
=&\mathbb{E}_{\mathcal{J}_1}\Bigg[\frac{2}{n}\mathbb{E}_{\mathcal{J}_2}\Bigg[\sum_{i=n+1}^{2n}\Big(\|\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(\widehat{v}_i)\|^2_{\mathcal{S}_\mathcal{Y}}+\|\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\\
&+2\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}\circ\Psi(u_i)-\mathcal{S}_\mathcal{Y}(\epsilon_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big)\Big|\mathcal{J}_1\Bigg]\Bigg]\\
=&\mathbb{E}_{\mathcal{J}_1}\Bigg[\frac{2}{n}\mathbb{E}_{\mathcal{J}_2}\Bigg[\sum_{i=n+1}^{2n}\Big(\|\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(\widehat{v}_i)\|^2_{\mathcal{S}_\mathcal{Y}}-\|\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\\
&+2\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big)\Big|\mathcal{J}_1\Bigg]\Bigg]\\
=&\mathbb{E}_{\mathcal{J}_1}\Bigg[2\mathbb{E}_{\mathcal{J}_2}\left[\inf_{\Gamma'_{\rm NN}\in\mathcal{F}^\Gamma_{\rm NN}}\frac{1}{n}\sum_{i=n+1}^{2n}\left(\|\Gamma'_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(\widehat{v}_i)\|^2_{\mathcal{S}_\mathcal{Y}}-\|\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\right)\Big|\mathcal{J}_1\right]\\
&+\mathbb{E}_{\mathcal{J}_2}\left[\frac{4}{n}\sum_{i=n+1}^{2n}\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\Bigg]\\
\leq&\mathbb{E}_{\mathcal{J}_1}\Bigg[2\inf_{\Gamma'_{\rm NN}\in\mathcal{F}^\Gamma_{\rm NN}}\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left(\|\Gamma'_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(\widehat{v}_i)\|^2_{\mathcal{S}_\mathcal{Y}}-\|\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\right)\Big|\mathcal{J}_1\right]\\
&+\mathbb{E}_{\mathcal{J}_2}\left[\frac{4}{n}\sum_{i=n+1}^{2n}\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\Bigg]\\
=&\mathbb{E}_{\mathcal{J}_1}\Bigg[2\inf_{\Gamma'_{\rm NN}\in\mathcal{F}^\Gamma_{\rm NN}}\mathbb{E}_{\mathcal{J}_2}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left(\|\Gamma'_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(v_i)-\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}-\|\mathcal{S}_\mathcal{Y}(\epsilon_i)\|^2_{\mathcal{S}_\mathcal{Y}}\right)\Big|\mathcal{J}_1\right]\\
&+\mathbb{E}_{\mathcal{J}_2}\left[\frac{4}{n}\sum_{i=n+1}^{2n}\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\Bigg]\\
=&\mathbb{E}_{\mathcal{J}_1}\Bigg[2\inf_{\Gamma'_{\rm NN}\in\mathcal{F}^\Gamma_{\rm NN}}\mathbb{E}_{\mathcal{J}_2}\Bigg[\frac{1}{n}\sum_{i=n+1}^{2n}\Big(\|\Gamma'_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}\circ\Psi(u_i)\|^2_{\mathcal{S}_\mathcal{Y}}\\
&-2\left\langle\Gamma_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i)-\mathcal{S}_\mathcal{Y}(v_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big)\Big|\mathcal{J}_1\Bigg]+\mathbb{E}_{\mathcal{J}_2}\left[\frac{4}{n}\sum_{i=n+1}^{2n}\left\langle\Gamma^n_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\Bigg]\\
=&\underbrace{\mathbb{E}_{\mathcal{J}_1}\left[2\inf_{\Gamma'_{\rm NN}\in\mathcal{F}^\Gamma_{\rm NN}}\mathbb{E}_{\rm u\sim\gamma}\left[\|\Gamma'_{\rm NN}\circ E^n_\mathcal{X}\circ\mathcal{S}_\mathcal{X}(u)-\mathcal{S}_\mathcal{Y}\circ\Psi(u)\|^2_{\mathcal{S}_\mathcal{Y}}\Big|\mathcal{J}_1\right]\right]}_{\mathrm{T}_{1a}}\\
&\underbrace{+\mathbb{E}_{\mathcal{J}}\left[\frac{4}{n}\sum_{i=n+1}^{2n}\left\langle\Phi^n_{\rm NN}\circ\mathcal{S}_\mathcal{X}(u_i),\mathcal{S}_\mathcal{Y}(\epsilon_i)\right\rangle_{\mathcal{S}_\mathcal{Y}}\right]}_{\mathrm{T}_{1b}}.
\end{aligned} \tag{45}
$$

Term $\mathrm{T}_{1\mathrm{a}}$ captures the approximation error and term $\mathrm{T}_{1\mathrm{b}}$ captures the stochastic error. We will analyze $\mathrm{T}_{1\mathrm{a}}$ and $\mathrm{T}_{1\mathrm{b}}$ separately.

Bounding $\mathrm{T}_{1\mathrm{a}}$. We first focus on $\mathrm{T}_{1\mathrm{a}}$ in (45) and derive an upper bound using approximation results of $\widetilde{\mathbf{f}},\widetilde{\mathbf{g}}$ and Γ. We define the ξ-neighborhood of a set as follows:

Definition 3. For a set $\Xi \in [-M,M]^D$ for some $M>0$, the ξ-neighborhood containing Ξ is defined as the set

$$T_\xi(\Xi)=\{\mathbf{z}|\inf_{\mathbf{z}_1\in\Xi}\|\mathbf{z}-\mathbf{z}_1\|_\infty\le\xi\}. \tag{46}$$

We next show that for any $\xi\ge 0$, the function $\widetilde{\mathbf{f}}$ defined in Lemma 1 extended to $\mathbb{R}^{D_1}$ according to Lemma 3 can be well approximated by a network on $T_\xi(\widetilde{\mathcal{M}})$.

The following lemma shows that we can approximate $\mathrm{CL}(\widetilde{\mathbf{f}})$ well on a ξ-neighborhood of $\widetilde{\mathcal{M}}$, where $\mathrm{CL}(\cdot)$ is the clipping operator defined in (31).

Lemma 9. *For any $\varepsilon\in(0,1/4)$ and $\xi\ge 0$, set network architectures $\mathcal{F}_{\mathrm{NN}}(D_1,d,L_1,p_1,K_1,\kappa_1,M_1)$ with*

$$L_1=O(\log\varepsilon^{-1}),\ p_1=O(\varepsilon_1^{-d}),\ K_1=O(\varepsilon_1^{-d}),\ \kappa_1=O(\varepsilon_1^{-7}),\ M_1=1.$$

For any Lipschitz function $\widetilde{\mathbf{f}}:\widetilde{\mathcal{M}}\to[-1,1]^d$ extended to domain $[-R_{\mathcal{X}},R_{\mathcal{X}}]^D$ according to Lemma 3, with Lipscthiz constant bounded by $L_{\mathbf{f}}$, there exists $\widetilde{\mathbf{f}}_{\mathrm{NN}}\in\mathcal{F}_{\mathrm{NN}}(D_1,d,L_1,p_1,K_1,\kappa_1,M_1)$ such that

$$\sup_{\widetilde{u}\in T_\xi(\widetilde{\mathcal{M}})}\|\widetilde{\mathbf{f}}_{\mathrm{NN}}(\widetilde{u})-\mathrm{CL}\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty\le CD(\varepsilon+\xi), \tag{47}$$

for some absolute constant C. The constant hidden in $O(\cdot)$ depends on $d,L_{\mathbf{f}},\xi,R_{\mathcal{X}}$ and is polynomial in D_1.

Lemma 9 is proved in Appendix B.6.

Let $\varepsilon_1,\xi\in(0,1/4)$. By Lemma 9, set the network architecture $\bar{\mathcal{F}}_{\mathrm{NN}}^{E_{\mathcal{X}}}=\mathcal{F}_{\mathrm{NN}}(D_1,d,L_5,p_5,K_5,\kappa_5,M_5)$ with

$$L_5=O(\log\varepsilon_1^{-1}),\ p_5=O(\varepsilon_1^{-d}),\ K_5=O(\varepsilon_1^{-d}),\ \kappa_5=O(\varepsilon_1^{-7}),\ M_5=1.$$

There exists $\bar{\mathbf{f}}_{\mathrm{NN}}\in\bar{\mathcal{F}}_{\mathrm{NN}}^{E_{\mathcal{X}}}$ satisfying

$$\sup_{\widetilde{u}\in T_\xi(\widetilde{\mathcal{M}})}\|\bar{\mathbf{f}}_{\mathrm{NN}}(\widetilde{u})-\mathrm{CL}\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty\le C_3(\varepsilon_1+\xi) \tag{48}$$

for some C_3 linear in D_1, where $\mathrm{CL}(\cdot)$ is the clipping operator defined in (31).

By Theorem 1, set the network architecture $\bar{\mathcal{F}}_{\mathrm{NN}}^{\Gamma}=\mathcal{F}_{\mathrm{NN}}(d,D_2,L_6,p_6,K_6,\kappa_6,M_6)$ with

$$L_6=O(\log\varepsilon_1^{-1}),\ p_6=O(\varepsilon_1^{-d}),\ K_6=O(\varepsilon_1^{-d}\log\varepsilon_1^{-1}),\ \kappa_6=O(\varepsilon_1^{-1}),\ M_6=R_{\mathcal{Y}}.$$

There exists $\bar{\Gamma}_{\mathrm{NN}}\in\bar{\mathcal{F}}_{\mathrm{NN}}^{\Gamma}$ satisfying

$$\sup_{\mathbf{z}\in[-1,1]^d}\|\bar{\Gamma}_{\mathrm{NN}}(\mathbf{z})-\Gamma(\mathbf{z})\|_\infty\le\varepsilon_1.$$

The constant hidden in $O(\cdot)$ depends on $d,L_\Psi,L_{\mathbf{g}}$ and is linear in D_2.

Define the network

$$\mathcal{F}_{\mathrm{NN}}^{\Gamma}=\Big\{\Gamma'_{\mathrm{NN}}:[-1,1]^d\to[-R_{\mathcal{Y}},R_{\mathcal{Y}}]^{D_2}\,|\,\Gamma'_{\mathrm{NN}}=\bar{\Gamma}'_{\mathrm{NN}}\circ\bar{\mathbf{f}}'_{\mathrm{NN}}\circ\widetilde{\mathbf{g}}'_{\mathrm{NN}}\\ \text{for }\bar{\Gamma}'_{\mathrm{NN}}\in\bar{\mathcal{F}}_{\mathrm{NN}}^{\Gamma},\ \bar{\mathbf{f}}'_{\mathrm{NN}}\in\bar{\mathcal{F}}_{\mathrm{NN}}^{E_{\mathcal{X}}},\ \widetilde{\mathbf{g}}'_{\mathrm{NN}}\in\mathcal{F}_{\mathrm{NN}}^{D_{\mathcal{X}}}\Big\}$$

where $\mathcal{F}_{\mathrm{NN}}^{D_{\mathcal{X}}}$ is given in (38).

We have $\mathcal{F}_{\mathrm{NN}}^{\Gamma}\in\mathcal{F}_{\mathrm{NN}}(d,D_2,L_7,p_7,K_7,\kappa_7,M_7)$ with

$$L_7=O\left(\log(\varepsilon^{-1}+\varepsilon_1^{-1})\right),\ p_7=O\left(\varepsilon^{-d}+\varepsilon_1^{-d}\right),\ K_7=O\left((\varepsilon^{-d}+\varepsilon_1^{-d})\log(\varepsilon^{-1}+\varepsilon_1^{-1})\right),$$
$$\kappa_7=O\left(\varepsilon^{-1}+\varepsilon_1^{-7}\right),\ M_7=R_{\mathcal{Y}}$$

and $\bar{\Gamma}_{\mathrm{NN}}\circ\bar{\mathbf{f}}_{\mathrm{NN}}\circ D_{\mathcal{X}}^n\in\mathcal{F}_{\mathrm{NN}}^{\Gamma}$. We thus have

$$\begin{aligned}\mathrm{T}_{1\mathrm{a}}=&\mathbb{E}_{\mathcal{J}_1}\left[2\inf_{\Gamma'_{\mathrm{NN}}\in\mathcal{F}_{\mathrm{NN}}^{\Gamma}}\mathbb{E}_{u\sim\gamma}\left[\|\Gamma'_{\mathrm{NN}}\circ E_{\mathcal{X}}^n\circ S_{\mathcal{X}}(u)-S_{\mathcal{Y}}\circ\Psi(u)\|_{S_{\mathcal{Y}}}^2\Big|\mathcal{J}_1\right]\right.\\ \le&\mathbb{E}_{\mathcal{J}_1}\left[2\mathbb{E}_{u\sim\gamma}\left[\|\bar{\Gamma}_{\mathrm{NN}}\circ\bar{\mathbf{f}}_{\mathrm{NN}}\circ D_{\mathcal{X}}^n\circ E_{\mathcal{X}}^n\circ S_{\mathcal{X}}(u)-S_{\mathcal{Y}}\circ\Psi(u)\|_{S_{\mathcal{Y}}}^2\Big|\mathcal{J}_1\right]\right]\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{\mathcal{J}}\left[2\mathbb{E}_{u\sim\gamma}\left[\|\bar{\Gamma}_{\rm NN}\circ\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \Gamma\circ\widetilde{\mathbf{f}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}}\right]\right]\\
&\le \mathbb{E}_{\mathcal{J}}\Big[2\mathbb{E}_{u\sim\gamma}\Big[3\|\bar{\Gamma}_{\rm NN}\circ\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \Gamma\circ\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}}\\
&\quad + 3\|\Gamma\circ\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \Gamma\circ({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}}\\
&\quad + 3\|\Gamma\circ({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \Gamma\circ\widetilde{\mathbf{f}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}}\Big]\Big]\\
&\le \mathbb{E}_{\mathcal{J}}\Big[2\mathbb{E}_{u\sim\gamma}\Big[3|\Omega_{\mathcal{Y}}|\varepsilon_1^2\\
&\quad + 12L_\Psi^2L_{\mathbf{g}}^2\|\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - ({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u)\|_2^2\\
&\quad + 12L_\Psi^2L_{\mathbf{g}}^2\|({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \widetilde{\mathbf{f}}\circ S_{\mathcal{X}}(u)\|_2^2\Big]\Big],
\end{aligned}
\tag{49}
$$

where we used Lemma 6 in the last inequality.

To derive an upper bound of (49), denote

$$
\begin{aligned}
{\rm I} &= 12L_\Psi^2L_{\mathbf{g}}^2\|\bar{\mathbf{f}}_{\rm NN}\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - ({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u)\|_2^2,\\
{\rm II} &= 12L_\Psi^2L_{\mathbf{g}}^2\|({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - \widetilde{\mathbf{f}}\circ S_{\mathcal{X}}(u)\|_2^2.
\end{aligned}
$$

We have

$$
\begin{aligned}
{\rm I} &\le 12L_\Psi^2L_{\mathbf{g}}^2(64dR_{\mathcal{X}}^2L_{\mathbf{f}}^2) = 768dL_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2R_{\mathcal{X}}^2,\\
{\rm II} &\le 48L_\Psi^2L_{\mathbf{g}}^2d.
\end{aligned}
$$

When $\|D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - S_{\mathcal{X}}(u)\|_\infty^2 \le \xi^2$, we have $D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) \in T_\xi(\widetilde{\mathcal{M}})$, and by (48),

$$
\begin{aligned}
{\rm I} &\le 24C_3^2L_\Psi^2L_{\mathbf{g}}^2(\varepsilon_1^2+\xi^2),\\
{\rm II} &\le 12L_\Psi^2L_{\mathbf{g}}^2\|({\rm CL}\circ\widetilde{\mathbf{f}})\circ D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - {\rm CL}\circ\widetilde{\mathbf{f}}\circ S_{\mathcal{X}}(u)\|_2^2\\
&\le 96L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2\|D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{X}}} \le 96L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2\xi^2.
\end{aligned}
$$

We thus have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}[{\rm I}+{\rm II}] \le& \mathbb{P}(\|D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - S_{\mathcal{X}}(u)\|_\infty^2 \le \xi^2)(24C_3^2L_\Psi^2L_{\mathbf{g}}^2 + 96L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2)(\varepsilon_1^2+\xi^2)\\
&+ \mathbb{P}(\|D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - S_{\mathcal{X}}(u)\|_\infty^2 \ge \xi^2)(768|\Omega_{\mathcal{X}}|L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2R_{\mathcal{X}}^2 + 48L_\Psi^2L_{\mathbf{g}}^2)\\
\le&(24C_3^2L_\Psi^2L_{\mathbf{g}}^2 + 96L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2)(\varepsilon_1^2+\xi^2)\\
&+ (768|\Omega_{\mathcal{X}}|L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2R_{\mathcal{X}}^2 + 48L_\Psi^2L_{\mathbf{g}}^2)\frac{\mathbb{E}\left[\|D^n_{\mathcal{X}}\circ E^n_{\mathcal{X}}\circ S_{\mathcal{X}}(u) - S_{\mathcal{X}}(u)\|_\infty^2\right]}{\xi^2}\\
\le&(24C_3^2L_\Psi^2L_{\mathbf{g}}^2 + 96L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2)(\varepsilon_1^2+\xi^2)\\
&+ (768|\Omega_{\mathcal{X}}|L_\Psi^2L_{\mathbf{g}}^2L_{\mathbf{f}}^2R_{\mathcal{X}}^2 + 48L_\Psi^2L_{\mathbf{g}}^2)\frac{C_2n^{-\frac{2}{2+d}}\log^3 n}{\xi^2},
\end{aligned}
\tag{50}
$$

where we used (43) in the last two inequalities.

Set $\varepsilon_1^2 = \xi^2 = n^{-\frac{1}{2+d}}$ with $n > 4^{2+d}$, (50) becomes

$$
\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}[{\rm I}+{\rm II}] \le C_4n^{-\frac{1}{2+d}}\log^3 n,
\tag{51}
$$

for some C_4 depending on $d, L_{\mathbf{f}}, L_{\mathbf{g}}, L_\Psi, R_{\mathcal{X}}, R_{\mathcal{Y}}, |\Omega_{\mathcal{X}}|$ and is polynomial in D_1 and is linear in D_2.

Substituting (51) and $\varepsilon_1^2 = n^{-\frac{1}{2+d}}$ into (49) gives rise to

$$
{\rm T}_{1a} \le C_5n^{-\frac{1}{2+d}}\log^3 n,
\tag{52}
$$

for some C_5 depending on $d, L_{\mathbf{f}}, L_{\mathbf{g}}, L_\Psi, R_{\mathcal{X}}, R_{\mathcal{Y}}, |\Omega_{\mathcal{X}}|, |\Omega_{\mathcal{Y}}|$ and is polynomial in D_1 and is linear in D_2.

The resulting network architecture is $\mathcal{F}^\Gamma_{\rm NN} \in \mathcal{F}_{\rm NN}(d, D_2, L_7, p_7, K_7, \kappa_7, M_7)$ with

$$
L_7 = O\left(\log(\varepsilon^{-1})\right),\ p_7 = O\left(\varepsilon^{-d}\right),\ K_7 = O\left(\varepsilon^{-d}\log\varepsilon^{-1}\right),\ \kappa_7 = O\left(\varepsilon^{-7/2}\right),\ M_7 = R_{\mathcal{Y}}
$$

for $\varepsilon = n^{-\frac{2}{2+d}}$.

Bounding T_{1b}. Define the network architecture

$$\mathcal{F}^{\Phi}_{\mathrm{NN}} = \{\Phi'_{\mathrm{NN}} | \Phi'_{\mathrm{NN}} = \Gamma'_{\mathrm{NN}} \circ E'_{\mathrm{NN}} \ \text{ for } \ \Gamma'_{\mathrm{NN}} \in \mathcal{F}^{\Gamma}_{\mathrm{NN}},\ E'_{\mathrm{NN}} \in \mathcal{F}^{E_{\mathcal{X}}}_{\mathrm{NN}}\}$$

and denote

$$\|\Phi'_{\mathrm{NN}}\|_n^2 = \frac{1}{n}\sum_{i=n+1}^{2n} \|\Phi'_{\mathrm{NN}} \circ \mathcal{S}_{\mathcal{X}}(u_i)\|^2_{\mathcal{S}_{\mathcal{Y}}}.$$

An upper bound of T_{1b} is given by the following lemma (see a proof in Appendix B.7):

Lemma 10. *Under conditions of Theorem 2, for any $\delta > 0$, we have*

$$\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n} \left\langle \Phi^n_{\mathrm{NN}} \circ \mathcal{S}_{\mathcal{X}}(u_i), \mathcal{S}_{\mathcal{Y}}(\epsilon_i)\right\rangle_{\mathcal{S}_{\mathcal{Y}}}\right]$$
$$\le 2\sqrt{|\Omega_{\mathcal{Y}}|}\sigma\left(\sqrt{\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\mathrm{NN}} - \Phi\|_n^2\right]} + \sqrt{D_2}\delta\right)\sqrt{\frac{4\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 6}{n}} + |\Omega_{\mathcal{Y}}|\delta\sigma. \tag{53}$$

Substituting (52) and (53) into (45) gives rise to

$$\mathrm{T}_1 = 2\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n} \|\Phi^n_{\mathrm{NN}} \circ \mathcal{S}_{\mathcal{X}}(u_i) - \mathcal{S}_{\mathcal{Y}} \circ \Psi(u_i)\|^2_{\mathcal{S}_{\mathcal{Y}}}\right] = 2\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\mathrm{NN}} - \Phi\|_n^2\right]$$
$$\le C_5 n^{-\frac{1}{2+d}}\log^3 n + 4|\Omega_{\mathcal{Y}}|\delta\sigma + 8\sqrt{|\Omega_{\mathcal{Y}}|}\sigma\left(\sqrt{\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\mathrm{NN}} - \Phi\|_n^2\right]} + \sqrt{|\Omega_{\mathcal{Y}}|}\delta\right)\sqrt{\frac{4\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 6}{n}}. \tag{54}$$

In (54), the term $\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\mathrm{NN}} - \Phi\|_n^2\right]$ appears on both sides. We next derive an upper bound of T_1 by applying some inequality to (54).

Denote

$$\omega = \sqrt{\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\mathrm{NN}} - \Phi\|_n^2\right]},$$
$$a = \frac{C_5 n^{-\frac{1}{2+d}}\log^3 n}{2} + 2|\Omega_{\mathcal{Y}}|\delta\sigma + 4|\Omega_{\mathcal{Y}}|\sigma\delta\sqrt{\frac{4\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 6}{n}},$$
$$b = 2\sqrt{|\Omega_{\mathcal{Y}}|}\sigma\sqrt{\frac{4\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 6}{n}}.$$

Relation (54) implies

$$\omega^2 \le a + 2b\omega.$$

We deduce

$$(\omega - b)^2 \le a + b^2 \Rightarrow |\omega - b| \le \sqrt{a + b^2} \le \sqrt{a} + b.$$

When $\omega \ge b$, we have

$$w - b \le \sqrt{a} + b \Rightarrow w \le \sqrt{a} + 2b \Rightarrow w^2 \le (\sqrt{a} + 2b)^2 \le 2a + 8b^2.$$

When $\omega < b$, we also have $\omega^2 \le 2a + 8b^2$. Substituting the expression of ω, a, b into the relation $\omega^2 \le 2a + 8b^2$, we have

$$\mathrm{T}_1 = 2\omega^2 \le 2C_5 n^{-\frac{1}{2+d}}\log^3 n + 8|\Omega_{\mathcal{Y}}|\delta\sigma + 16|\Omega_{\mathcal{Y}}|\sigma\delta\sqrt{\frac{4\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 6}{n}}$$
$$+ 128|\Omega_{\mathcal{Y}}|\sigma^2\frac{2\log\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}) + 3}{n}. \tag{55}$$

Bounding T_2. The upper bound of T_2 can be derived using the covering number $\mathcal{N}(\delta, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}})$ and Bernstein-type inequalities. The upper bound is summarized in the following lemma (see a proof in Appendix B.8).

Lemma 11. *Under conditions of Theorem 2, for any $\delta > 0$, we have*

$$\mathrm{T}_2 \le \frac{48R^2_{\mathcal{Y}}|\Omega_{\mathcal{Y}}|}{n}\log\mathcal{N}\left(\frac{\delta}{4R_{\mathcal{Y}}|\Omega_{\mathcal{Y}}|}, \mathcal{F}^{\Phi}_{\mathrm{NN}}, \|\cdot\|_{L^{\infty,\infty}}\right) + 6\delta. \tag{56}$$

Putting T_1 and T_2 together. Substituting (55) and (56) into (44), we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{J}}\mathbb{E}_{u\sim\gamma}\|\Phi^n_{\mathrm{NN}}\circ\mathcal{S}_{\mathcal{X}}(u)-\mathcal{S}_{\mathcal{Y}}\circ\Psi(u)\|^2_{\mathcal{S}_{\mathcal{Y}}}\\ \leq&2C_5n^{-\frac{1}{2+d}}\log^3 n+(8|\Omega_{\mathcal{Y}}|+6)\delta\sigma+16|\Omega_{\mathcal{Y}}|\sigma\delta\sqrt{\frac{4\log\mathcal{N}(\delta,\mathcal{F}^{\Phi}_{\mathrm{NN}},\|\cdot\|_{L^{\infty,\infty}})+6}{n}}+6\delta\\ &+64|\Omega_{\mathcal{Y}}|\sigma^2\frac{2\log\mathcal{N}(\delta,\mathcal{F}^{\Phi}_{\mathrm{NN}},\|\cdot\|_{L^{\infty,\infty}})+3}{n}+\frac{48R^2_{\mathcal{Y}}|\Omega_{\mathcal{Y}}|}{n}\log\mathcal{N}\left(\frac{\delta}{4R_{\mathcal{Y}}|\Omega_{\mathcal{Y}}|},\mathcal{F}^{\Phi}_{\mathrm{NN}},\|\cdot\|_{L^{\infty,\infty}}\right).\end{aligned}\tag{57}$$

The network architecture satisfies $\mathcal{F}^{\Phi}_{\mathrm{NN}}\subset\mathcal{F}_{\mathrm{NN}}(D_1,D_2,L_8,p_8,K_8,\kappa_8,M_8)$ with

$$L_8=O(\log\varepsilon^{-1}),\ p_8=O(\varepsilon^{-d}),\ K_8=O(\varepsilon^{-d}\log\varepsilon^{-1}),\ \kappa_8=O(\varepsilon^{-7}),\ M_8=R_{\mathcal{Y}}.\tag{58}$$

Substituting (58) with $\varepsilon=n^{-\frac{1}{2+d}},\delta=n^{-1}$ into Lemma 8, we get

$$\log\mathcal{N}\left(\delta,\mathcal{F}^{\Phi}_{\mathrm{NN}},\|\cdot\|_{L^{\infty,\infty}}\right)\leq C_6\varepsilon^{-d}\log^2\varepsilon^{-1}(\log\varepsilon^{-1}+\log\delta^{-1})\leq 2C_6n^{\frac{d}{2+d}}\log^3 n,\tag{59}$$

for some C_6 depending on $d,L_{\mathbf{f}},L_{\mathbf{g}},L_{\Psi},R_{\mathcal{X}},R_{\mathcal{Y}},|\Omega_{\mathcal{X}}|,|\Omega_{\mathcal{Y}}|$ and is polynomial in D_1 and is linear in D_2.

Substituting (59) into (57) gives rise to

$$\mathbb{E}_{\mathcal{J}}\mathbb{E}_{u\sim\gamma}\|\Phi^n_{\mathrm{NN}}\circ\mathcal{S}_{\mathcal{X}}(u)-\mathcal{S}_{\mathcal{Y}}\circ\Psi(u)\|^2_{\mathcal{S}_{\mathcal{Y}}}\leq C_7(1+\sigma^2)n^{-\frac{1}{2+d}}\log^3 n,$$

for some C_7 depending on $d,L_{\mathbf{f}},L_{\mathbf{g}},L_{\Psi},R_{\mathcal{X}},R_{\mathcal{Y}},|\Omega_{\mathcal{X}}|,|\Omega_{\mathcal{Y}}|$ and is polynomial in D_1 and is linear in D_2.

The resulting network architectures are stated in (19), (20) and (21). □

7. Conclusion and discussion

This paper explores the use of Auto-Encoder-based neural network (AENet) for operator learning in function spaces, leveraging Auto-Encoders-based nonlinear model reduction techniques. This approach is particularly effective when input functions are situated on a nonlinear manifold. In such cases, an Auto-Encoder is utilized to identify and represent input functions as latent variables. These latent variables are then transformed during the operator learning process to produce outputs. Our study establishes a comprehensive approximation theory and performs an in-depth analysis of generalization errors. The findings indicate that the efficiency of AENet, measured in terms of sample complexity, is closely linked to the intrinsic dimensionality of the underlying model.

We next discuss some potential applications and improvement of this work.

Network architecture: In this paper, we have an Auto-Encoder applied on the input functions, instead of two Auto-Encoders applied on the input and output functions, respectively, since in our numerical experiments, training the Auto-Encoder on the output is almost as hard as training the transformation from the input latent variable to the output. In literature, Auto-Encoders are applied on the output in Seidman et al. [58] and Kontolati et al. [30]. For the simulation of high-dimensional PDEs, it may be important to apply an Auto-Encoder on the output to reduce its dimension. Our proof technique may be extended to Auto-Encoder-based neural networks (AENets) where two Auto-Encoders are applied on the input and output functions, respectively. We will investigate this in our future work.

Generating the solution manifold: AENet has the advantage of producing the solution manifold of the operator Ψ from low-dimensional latent variables. With the transformation Γ^n_{NN} given in (1), we can express the solution manifold as $\{\Gamma^n_{\mathrm{NN}}(\mathbf{z}):\ \mathbf{z}\in[-1,1]^d\}$. In other words, AENet not only learns the operator Ψ, but also gives rise to the solution manifold. AENet is a potential tool to study the geometric structure of the solution manifold.

Data splitting in Stage I and II: Our algorithm involves a data splitting in Stage I and II, in order to create data independence in Stage I and II for the proof of the generalization error. This data splitting strategy is only for theory purpose. In experiments, we use all training data in Stage I and II.

Optimality of convergence rate: This paper provides the first generalization error analysis of nonlinear model reduction by deep neural networks. Theorem 2 proves a power-law convergence of the squared generalization error as n increases, and the exponent depends on the intrinsic dimension of the model. The rate of convergence (exponent) in Theorem 2 may not be optimal. One of our future works is to improve the rate of convergence.

Appendix A. Example 1 and the error bound in Remark 1 and Remark 3

A.1. *Proof of Example 1*

Proof of Example 1. For any $u=\sum_{k=-N}^N a_k e^{2\pi ikx}$, we have

$$\|u\|_{L^2([0,1])}=\sqrt{\sum_{k=-N}^N|a_k|^2}\geq a\sqrt{2N+1},$$

$$\sup_{x\in[0,1]}\left|\frac{du}{dx}\right| = \sup_{x\in[0,1]}\left|\sum_{k=-N}^{N} a_k 2\pi i k e^{2\pi i k x}\right| \le 2\pi A\sum_{k=-N}^{N}|k| = 2\pi AN(N+1).$$

Therefore, $\delta \ge \frac{a\sqrt{2N+1}}{2\pi AN(N+1)}$. $\square$

A.2. Derivation of (18)

We have

$$\begin{aligned}&\sup_{u\in\mathcal{M}}\|\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-\Phi(\widetilde{u})\|_\infty\\ \le&\sup_{u\in\mathcal{M}}\left[\|\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)-S_{\mathcal{Y}}\circ\Psi(u)\|_\infty+\|\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)\|_\infty\right]\\ \le&\varepsilon+\sup_{u\in\mathcal{M}}\|\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-\Phi_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)\|_\infty,\end{aligned}$$

where the last inequality is based on Corollary 1.

A.3. Error bound in (23)

We have

$$\begin{aligned}&\mathbb{E}_{\mathcal{J}}\mathbb{E}_{u\sim\gamma}\|\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-S_{\mathcal{Y}}\circ\Psi(u)\|^2_{S_{\mathcal{Y}}}\\ \le&\mathbb{E}_{\mathcal{J}}\mathbb{E}_{u\sim\gamma}[\|\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)-S_{\mathcal{Y}}\circ\Psi(u)\|^2_{S_{\mathcal{Y}}}+\|\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}}]\\ \le&C(1+\sigma^2)n^{-\frac{1}{2+d}}\log^3 n+\mathbb{E}_{\mathcal{J}}\mathbb{E}_{u\sim\gamma}\|\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(P_{\mathrm{intp},\mathcal{X}}(S'_{\mathcal{X}}(u)))-\Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u)\|^2_{S_{\mathcal{Y}}},\end{aligned}$$

where the last inequality is based on Theorem 2.

Appendix B. Proofs of lemmas

B.1. Proof of Lemma 1

Proof of Lemma 1. We have

$$\widetilde{\mathbf{g}}\circ\widetilde{\mathbf{f}}(S_{\mathcal{X}}(u)) = S_{\mathcal{X}}\circ\mathbf{g}\circ\mathbf{f}(u) = S_{\mathcal{X}}(u)$$

which proves (15). The Lipschitz property of $\widetilde{\mathbf{f}}$ follows from, for any $u_1,u_2\in\mathcal{M}$,

$$\|\widetilde{\mathbf{f}}(S_{\mathcal{X}}(u_1))-\widetilde{\mathbf{f}}(S_{\mathcal{X}}(u_2))\|_2 = \|\mathbf{f}(u_1)-\mathbf{f}(u_2)\|_2 \le L_{\mathbf{f}}\|u_1-u_2\|_{\mathcal{X}} \le 2L_{\mathbf{f}}\|S_{\mathcal{X}}(u_1)-S_{\mathcal{X}}(u_2)\|_{S_{\mathcal{X}}}.$$

We next show that $\widetilde{\mathbf{g}}$ is Lipschitz. For any $\mathbf{z}_1,\mathbf{z}_2\in[-1,1]^d$, we have

$$\|\widetilde{\mathbf{g}}(\mathbf{z}_1)-\widetilde{\mathbf{g}}(\mathbf{z}_2)\|_{S_{\mathcal{X}}} = \|S_{\mathcal{X}}\circ\mathbf{g}(\mathbf{z}_1)-S_{\mathcal{X}}\circ\mathbf{g}(\mathbf{z}_2)\|_{S_{\mathcal{X}}} \le 2\|\mathbf{g}(\mathbf{z}_1)-\mathbf{g}(\mathbf{z}_2)\|_{\mathcal{X}} \le 2L_{\mathbf{g}}\|\mathbf{z}_1-\mathbf{z}_2\|_2. \quad\square$$

B.2. Proof of Lemma 2

Proof of Lemma 2. By Lemma 1, we have

$$S_{\mathcal{X}}(\mathcal{M}) = \{\widetilde{\mathbf{g}}(\mathbf{z}):\ \mathbf{z}\in[-1,1]^d\}$$

where $\widetilde{\mathbf{g}}$ is Lipschitz. For any $\delta>0$, the covering number $\mathcal{N}(\delta,(0,1)^d,\|\cdot\|_2)$ is upper bounded by $C\delta^{-d}$ with a constant $C>0$ [59]. In other words, there exists a finite set $F_\delta\subset[-1,1]^d$ such that

- $\#F_\delta\le C\delta^{-d}$,
- $[-1,1]^d\subset\cup_{\mathbf{z}\in F_\delta}B_2^d(\theta,\delta)$.

The Lipschitz property of $\mathbf{g}$ and the condition $\|u\|_{L^\infty(\Omega_{\mathcal{X}})}\le c_1\|u\|_{\mathcal{X}}$ imply that

$$\begin{aligned}&\|\widetilde{\mathbf{g}}(\mathbf{z}_1)-\widetilde{\mathbf{g}}(\mathbf{z}_2)\|_\infty = \|S_{\mathcal{X}}\circ\mathbf{g}(\mathbf{z}_1)-S_{\mathcal{X}}\circ\mathbf{g}(\mathbf{z}_2)\|_\infty\\ \le&\|\mathbf{g}(\mathbf{z}_1)-\mathbf{g}(\mathbf{z}_2)\|_{L^\infty}\le c_1\|\mathbf{g}(\mathbf{z}_1)-\mathbf{g}(\mathbf{z}_2)\|_{\mathcal{X}}\le c_1L_{\mathbf{g}}\|\mathbf{z}_1-\mathbf{z}_2\|_2.\end{aligned} \tag{60}$$

The manifold $\mathcal{S}_{\mathcal{X}}(\mathcal{M})$ can be covered as

$$\mathcal{S}_{\mathcal{X}}(\mathcal{M}) \subset \widetilde{\mathbf{g}}\left(\cup_{\mathbf{z}\in F_\delta} B_2^d(\mathbf{z},\delta)\cap[-1,1]^d\right) \subset \cup_{\mathbf{z}\in F_\delta}\widetilde{\mathbf{g}}\left(B_2^d(\mathbf{z},\delta)\cap[-1,1]^d\right) \subset \cup_{\mathbf{z}\in F_\delta} B_\infty^D\left(\widetilde{\mathbf{g}}(\mathbf{z}), c_1 L_{\mathbf{g}}\delta\right)$$

where the last inclusion follows from (60). By setting $c_1 L_{\mathbf{g}}\delta = \varepsilon$, we have

$$\mathcal{N}(\varepsilon, \mathcal{S}_{\mathcal{X}}(\mathcal{M}), \|\cdot\|_\infty) \le \#F_\delta \le C\left(\frac{\varepsilon}{c_1 L_{\mathbf{g}}}\right)^{-d}.$$

Therefore, the Minkowski dimension of $\mathcal{S}_{\mathcal{X}}(\mathcal{M})$ is no more than d. □

B.3. Proof of Lemma 4

Proof of Lemma 4. Lemma 4 is a variant of [48, Theorem 5], and can be proved similarly. In [48, Proof of Theorem 5], the authors constructed $\Phi_\varepsilon^{f_1} := \Phi^{\max,5^D} \odot \Phi^{\text{sum}} \odot \Phi_{\varepsilon/2}^{\text{simul}}$ in order to have a constant number of layers. To relax such a requirement, we construct $\Phi_\varepsilon^{f_1}$ as $\Phi_\varepsilon^{f_1} = \Phi^{\max,\text{Card}\mathcal{I}} \odot \Phi_{\varepsilon/2}^{\text{simul}}$. Then the error bound can be derived by following the rest proof and the network architecture is specified in Lemma 4. □

B.4. Proof of Lemma 6

Proof of Lemma 6. We have

$$\begin{aligned}&\|\Gamma(\mathbf{z}_1)-\Gamma(\mathbf{z}_2)\|_{\mathcal{S}_{\mathcal{Y}}} = \|\mathcal{S}_{\mathcal{Y}}\circ\Psi\circ\mathbf{g}(\mathbf{z}_1) - \mathcal{S}_{\mathcal{Y}}\circ\Psi\circ\mathbf{g}(\mathbf{z}_2)\|_{\mathcal{S}_{\mathcal{Y}}}\\ \le& 2\|\Psi\circ\mathbf{g}(\mathbf{z}_1)-\Psi\circ\mathbf{g}(\mathbf{z}_2)\|_{\mathcal{Y}} \le 2L_\Psi\|\mathbf{g}(\mathbf{z}_1)-\mathbf{g}(\mathbf{z}_2)\|_{\mathcal{X}} \le 2L_\Psi L_{\mathbf{g}}\|\mathbf{z}_1-\mathbf{z}_2\|_2.\end{aligned}$$ □

B.5. Proof of Lemma 7

Proof of Lemma 7. To simplify the notation, denote $G_{\text{NN}}^n = D_{\mathcal{X}}^n \circ E_{\mathcal{X}}^n$. We decompose the error as

$$\begin{aligned}\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\left[\|G_{\text{NN}}^n(\widetilde{u})-\widetilde{u}\|_\infty^2\right] =& \underbrace{2\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{n}\sum_{i=1}^n \|G_{\text{NN}}^n(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]}_{\text{TX}_1} +\\ & \underbrace{\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u\sim\gamma}\left[\|G_{\text{NN}}^n(\widetilde{u})-\widetilde{u}\|_\infty^2\right] - 2\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{n}\sum_{i=1}^n\|G^n(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]}_{\text{TX}_2}.\end{aligned} \tag{61}$$

The term TX_1 captures the bias and the term TX_2 captures the variance.

To bound TX_1, we will use the approximation result in Theorem 1. Let $\mathcal{F}_{\text{NN}}^{E_{\mathcal{X}}}, \mathcal{F}_{\text{NN}}^{D_{\mathcal{X}}}$ be specified as in Lemma 7. According to Theorem 1 (i), there exists $\widetilde{\mathbf{f}}_{\text{NN}} \in \mathcal{F}_{\text{NN}}^{E_{\mathcal{X}}}, \widetilde{\mathbf{g}}_{\text{NN}} \in \mathcal{F}_{\text{NN}}^{D_{\mathcal{X}}}$ satisfying

$$\|\widetilde{\mathbf{g}}_{\text{NN}}\circ\widetilde{\mathbf{f}}_{\text{NN}}(\widetilde{u}) - \widetilde{\mathbf{g}}\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty \le \varepsilon_2 + 2\sqrt{d}L_{\mathbf{g}}\varepsilon_1$$

for any $\widetilde{u}\in\widetilde{\mathcal{M}}$. Denote $\widetilde{G}_{\text{NN}} = \widetilde{\mathbf{g}}_{\text{NN}}\circ\widetilde{\mathbf{f}}_{\text{NN}}$. We bound TX_1 as

$$\begin{aligned}\text{TX}_1 =& 2\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{n}\sum_{i=1}^n\|G_{\text{NN}}^n(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]\\ =& 2\mathbb{E}_{\mathcal{J}_1}\inf_{G'_{\text{NN}}\in\mathcal{F}_{\text{NN}}^G}\left[\frac{1}{n}\sum_{i=1}^n\|G'_{\text{NN}}(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]\\ \le& 2\inf_{G'_{\text{NN}}\in\mathcal{F}_{\text{NN}}^G}\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{n}\sum_{i=1}^n\|G'_{\text{NN}}(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]\\ \le& 2\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{n}\sum_{i=1}^n\|\widetilde{G}_{\text{NN}}(\widetilde{u}_i)-\widetilde{u}_i\|_\infty^2\right]\\ =& 2\mathbb{E}_{u\sim\gamma}\left[\|\widetilde{G}_{\text{NN}}(\widetilde{u})-\widetilde{\mathbf{g}}\circ\widetilde{\mathbf{f}}(\widetilde{u})\|_\infty^2\right]\\ \le& 16dL_{\mathbf{g}}^2\varepsilon_1^2 + 4\varepsilon_2^2.\end{aligned} \tag{62}$$

To bound TX_2, we will use the covering number of $\mathcal{F}^G_{\mathrm{NN}}$. We have the following lemma (see a proof in Appendix B.9).

Lemma 12. *Under the condition of Lemma 7, for any $\delta > 0$, we have*

$$\mathrm{TX}_2 \le \frac{48R^2_{\mathcal{X}}}{n}\log\mathcal{N}\left(\frac{\delta}{4R_{\mathcal{X}}}, \mathcal{F}^G_{\mathrm{NN}}, \|\cdot\|_{\infty,\infty}\right) + 6\delta. \tag{63}$$

Substituting Lemma 12 and (62) into (61) proves Lemma 7. □

B.6. Proof of Lemma 9

Proof of Lemma 9. We will use the following lemma, which is another variant of [48, Theorem 5].

Lemma 13. *Let D, d be positive integers with $d < D$, $M > 0$ and $\Xi \subset [-1,1]^D$ some set. Suppose $d \ge d_M\Xi$. For any $\varepsilon > 0$, consider a network class $\mathcal{F}_{\mathrm{NN}}(D,1,L,p,K,\kappa,M)$ with*

$$L = O(\log\varepsilon^{-1}),\ p = O(\varepsilon^{-d}),\ K = O(\varepsilon^{-d}),\ \kappa = O(\varepsilon^{-3-4(1+\lceil\log_2 M\rceil)}).$$

Then if $\varepsilon \in (0,1/4)$ and for any Lipschitz function $f : [-1,1]^D \to [-M,M]$ with Lipschitz constant bounded by L_f, there exists a network f_{NN} with this architecture so that

$$\|f_{\mathrm{NN}} - f\|_{L^\infty(T_\xi(\Xi))} \le CD(\varepsilon + \xi)$$

for some constant C depending on M. The constant hidden in $O(\cdot)$ depends on d, M, L_f, ξ and is only polynomial in D.

Proof of Lemma 13. Lemma 13 is another variant of [48, Theorem 5], and can be proved similarly. In [48, Proof of Theorem 5], the authors first cover Ξ using hyper-cubes with diameter r, which was set to $\varepsilon/3MD$. Denote set of cubes by $C_r = \{\Lambda_{k,r}\}_{k=1}^{C_\Xi}$. The authors in fact prove that if the network architecture is properly set, there exists a network f_{NN} with this architecture satisfying

$$\sup_{\mathbf{x}\in\bigcup_k \Lambda_{k,r}} |f_{\mathrm{NN}}(\widetilde{\mathbf{x}}) - f(\widetilde{\mathbf{x}})| \le \varepsilon. \tag{64}$$

Denote the center for $\Lambda_{k,r}$ by c_k. Instead covering Ξ by C_r, we will use $C_{r+2\xi} = \{\Lambda_{k,r+2\xi}\}_{k=1}^{C_\Xi}$, where $\Lambda_{k,r+2\xi}$ is the hyper-cube with center c_k and diameter $r+2\xi$. Then we have $T_\xi(\Xi) \subset \bigcup_k \Lambda_{k,r+2\xi}$.

Then similar to the proof of Lemma 4, we construct $\Phi^{f_1}_\varepsilon$ as $\Phi^{f_1}_\varepsilon = \Phi^{\max,\mathrm{Card}\mathcal{I}} \odot \Phi^{\mathrm{simul}}_{\varepsilon/2}$. By following the rest of the proof, we deduce that

$$\|f_{\mathrm{NN}}(\widetilde{\mathbf{x}}) - f(\widetilde{\mathbf{x}})\|_{L^\infty(T_\xi(\Xi))} = \sup_{\mathbf{x}\in T_\xi(\Xi)} |f_{\mathrm{NN}}(\widetilde{\mathbf{x}}) - f(\widetilde{\mathbf{x}})| \le \sup_{\mathbf{x}\in\bigcup_k \Lambda_{k,r}} |f_{\mathrm{NN}}(\widetilde{\mathbf{x}}) - f(\widetilde{\mathbf{x}})| \le CD(\varepsilon+\xi) \tag{65}$$

for some constant C depending on M. The network architecture is specified in Lemma 13. □

Lemma 9 can be proved by following the first part of the proof of Theorem 1 and replacing Lemma 4 by Lemma 13. □

B.7. Proof of Lemma 10

Proof of Lemma 10. Let $\{\phi_{\mathrm{NN},j}\}_{j=1}^{\mathcal{N}(\delta,\mathcal{F}^\Phi_{\mathrm{NN}},\|\cdot\|_{L^{\infty,\infty}})}$ be a δ-cover of $\mathcal{F}^\Phi_{\mathrm{NN}}$. There exists $\phi_{\mathrm{NN},*}$ in this cover satisfying $\|\phi_{\mathrm{NN},*} - \Phi^n_{\mathrm{NN}}\|_{L^{\infty,\infty}} \le \delta$. We have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
=&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u_i) - \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i) + \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i) - \Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
\le&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi^n_{\mathrm{NN}}\circ S_{\mathcal{X}}(u_i) - \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
&+\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i) - \Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right].
\end{aligned} \tag{66}$$

For the first term in (66), by Lemma 15 and Jensen's inequality, we have

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)-\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
\le&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\|\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)-\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)\|_{S_{\mathcal{Y}}}\|S_{\mathcal{Y}}(\epsilon_i)\|_{S_{\mathcal{Y}}}\right]\\
\le&\frac{\sqrt{|\Omega_{\mathcal{Y}}|}\delta}{n}\sum_{i=n+1}^{2n}\mathbb{E}_{\mathcal{J}}\left[\|S_{\mathcal{Y}}(\epsilon_i)\|_{S_{\mathcal{Y}}}\right]\\
\le&\frac{\sqrt{|\Omega_{\mathcal{Y}}|}\delta}{n}\sum_{i=n+1}^{2n}\sqrt{\mathbb{E}_{\mathcal{J}}\left[\|S_{\mathcal{Y}}(\epsilon_i)\|_{S_{\mathcal{Y}}}^2\right]}\\
\le&|\Omega_{\mathcal{Y}}|\delta\sigma.
\end{aligned}\tag{67}
$$

Substituting (67) into (66) gives rise to

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
\le&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]+|\Omega_{\mathcal{Y}}|\delta\sigma\\
=&\mathbb{E}_{\mathcal{J}}\left[\frac{\|\phi_{\mathrm{NN},*}-\Phi\|_n}{\sqrt{n}}\frac{\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}}{\sqrt{n}\|\phi_{\mathrm{NN},*}-\Phi\|_n}\right]+|\Omega_{\mathcal{Y}}|\delta\sigma.
\end{aligned}\tag{68}
$$

Note that

$$
\begin{aligned}
&\|\phi_{\mathrm{NN},*}-\Phi\|_n\\
=&\sqrt{\frac{1}{n}\sum_{i=n+1}^{2n}\|\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)+\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_{S_{\mathcal{Y}}}^2}\\
\le&\sqrt{\frac{2}{n}\sum_{i=n+1}^{2n}\left(\|\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)\|_{S_{\mathcal{Y}}}^2+\|\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_{S_{\mathcal{Y}}}^2\right)}\\
\le&\sqrt{\frac{2}{n}\sum_{i=n+1}^{2n}\left(|\Omega_{\mathcal{Y}}|\delta^2+\|\Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_{S_{\mathcal{Y}}}^2\right)}\\
\le&\sqrt{2}\|\Phi_{\mathrm{NN}}^n-\Phi\|_n+\sqrt{2|\Omega_{\mathcal{Y}}|}\delta,
\end{aligned}\tag{69}
$$

where we used $(a+b)^2\le 2a^2+2b^2$ in the first inequality, and $\sqrt{a^2+b^2}\le a+b$ for $a,b\ge 0$ in the last inequality.

Combining (69) and (68), we have

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]\\
\le&\mathbb{E}_{\mathcal{J}}\left[\frac{\sqrt{2}\|\Phi_{\mathrm{NN}}^n-\Phi\|_n+\sqrt{2|\Omega_{\mathcal{Y}}|}\delta}{\sqrt{n}}\frac{\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}}{\sqrt{n}\|\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_n}\right]+|\Omega_{\mathcal{Y}}|\delta\sigma.
\end{aligned}\tag{70}
$$

Denote $z_j=\frac{\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},j}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}}{\sqrt{n}\|\phi_{\mathrm{NN},j}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_n}$. Since $\phi_{\mathrm{NN},*}$ is one element of the δ-cover of $\mathcal{F}_{\Phi}$, we have

$$
\left|\frac{\sum_{i=n+1}^{2n}\left\langle \phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}}{\sqrt{n}\|\phi_{\mathrm{NN},*}\circ S_{\mathcal{X}}(u_i)-\Phi\circ S_{\mathcal{X}}(u_i)\|_n}\right|\le\max_j|z_j|.
$$

Apply Cauchy-Schwarz inequality to (70), we have

$$
\mathbb{E}_{\mathcal{J}}\left[\frac{1}{n}\sum_{i=n+1}^{2n}\left\langle \Phi_{\mathrm{NN}}^n\circ S_{\mathcal{X}}(u_i), S_{\mathcal{Y}}(\epsilon_i)\right\rangle_{S_{\mathcal{Y}}}\right]
$$

$$
\begin{aligned}
&\leq \mathbb{E}_{\mathcal{J}}\left[\frac{\sqrt{2}\|\Phi^n_{\rm NN}-\Phi\|_n+\sqrt{2|\Omega_{\mathcal{Y}}|\delta}}{\sqrt{n}}\max_j |z_j|\right]+|\Omega_{\mathcal{Y}}|\delta\sigma \\
&\leq \sqrt{\frac{2}{n}}\sqrt{\mathbb{E}_{\mathcal{J}}\left[\left(\|\Phi^n_{\rm NN}-\Phi\|_n+\sqrt{|\Omega_{\mathcal{Y}}|\delta}\right)^2\right]\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2\right]}+|\Omega_{\mathcal{Y}}|\delta\sigma \\
&\leq \sqrt{\frac{2}{n}}\left(\sqrt{\mathbb{E}_{\mathcal{J}}\left[2\|\Phi^n_{\rm NN}-\Phi\|_n^2\right]}+\sqrt{2|\Omega_{\mathcal{Y}}|\delta}\right)\sqrt{\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2\right]}+|\Omega_{\mathcal{Y}}|\delta\sigma \\
&=2\left(\sqrt{\mathbb{E}_{\mathcal{J}}\left[\|\Phi^n_{\rm NN}-\Phi\|_n^2\right]}+\sqrt{|\Omega_{\mathcal{Y}}|\delta}\right)\sqrt{\frac{\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2\right]}{n}}+|\Omega_{\mathcal{Y}}|\delta\sigma. \qquad (71)
\end{aligned}
$$

Since each element of $\mathcal{S}_{\mathcal{Y}}(\epsilon)$ is sub-Gaussian with variance parameter σ^2, for given $\{u_i\}_{i=1}^n$, each z_j is sub-Gaussian with variance parameter $|\Omega_{\mathcal{Y}}|\sigma^2$. Thus $\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2\right]$ involves a collection of squared sub-Gaussian variables. We bound it using moment generating function. For any $t>0$, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2|\{u_i\}_{i=1}^n\right] &=\frac{1}{t}\log\exp\left(t\mathbb{E}_{\mathcal{J}}\left[\max_j z_j^2|\{u_i\}_{i=1}^n\right]\right)\\
&\leq \frac{1}{t}\log\mathbb{E}_{\mathcal{J}}\left[\exp\left(t\max_j z_j^2|\{u_i\}_{i=1}^n\right)\right]\\
&\leq \frac{1}{t}\log\mathbb{E}_{\mathcal{J}}\left[\sum_j \exp(tz_j^2)|\{u_i\}_{i=1}^n\right]\\
&\leq \frac{1}{t}\log\mathcal{N}(\delta,\mathcal{F}^{\Phi}_{\rm NN},\|\cdot\|_{\infty,\infty})+\frac{1}{t}\log\mathbb{E}_{\mathcal{J}}\left[\exp(tz_1^2)|\{u_i\}_{i=1}^n\right]. \qquad (72)
\end{aligned}
$$

Since z_1 is sub-Gaussian with variance parameter $|\Omega_{\mathcal{Y}}|\sigma$ for given $\{u_i\}_{i=1}^n$, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{J}}\left[\exp(tz_1^2)|\{u_i\}_{i=1}^n\right] &=1+\sum_{k=1}^{\infty}\frac{t^p\mathbb{E}_{\mathcal{J}}\left[z_1^{2k}|\{u_i\}_{i=1}^n\right]}{k!}\\
&=1+\sum_{k=1}^{\infty}\left[\frac{t^p}{k!}\int_0^{\infty}\mathbb{P}(z_1\geq \eta^{1/2k})d\eta\right]\\
&\leq 1+2\sum_{k=1}^{\infty}\left[\frac{t^p}{k!}\int_0^{\infty}\exp\left(-\frac{\eta^{1/2p}}{2|\Omega_{\mathcal{Y}}|\sigma^2}\right)d\eta\right]\\
&=1+\sum_{k=1}^{\infty}\frac{2k(2t|\Omega_{\mathcal{Y}}|\sigma^2)^k}{k!}\Gamma_G(k)\\
&=1+2\sum_{k=1}^{\infty}(2t|\Omega_{\mathcal{Y}}|\sigma^2)^k, \qquad (73)
\end{aligned}
$$

where Γ_G denotes the Gamma function. Setting $t=(4|\Omega_{\mathcal{Y}}|\sigma^2)^{-1}$, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{J}}\left[\max_j |z_j|^2|\{u_i\}_{i=1}^n\right] &\leq 4|\Omega_{\mathcal{Y}}|\sigma^2\log\mathcal{N}(\delta,\mathcal{F}^{\Phi}_{\rm NN},\|\cdot\|_{\infty,\infty})+4|\Omega_{\mathcal{Y}}|\sigma^2\log 3\\
&\leq 4|\Omega_{\mathcal{Y}}|\sigma^2\log\mathcal{N}(\delta,\mathcal{F}^{\Phi}_{\rm NN},\|\cdot\|_{\infty,\infty})+6|\Omega_{\mathcal{Y}}|\sigma^2. \qquad (74)
\end{aligned}
$$

Substituting (74) into (70) proves the lemma. □

B.8. Proof of Lemma 11

Proof of Lemma 11. Lemma 11 can be proved by following the proof of Lemma 12. One only needs to replace the definition of $\widehat{g}(u)$ to $\widehat{g}(u)=\|\Phi^n_{\rm NN}\circ\mathcal{S}_{\mathcal{X}}(u)-\mathcal{S}_{\mathcal{Y}}\circ\Psi(u)\|^2_{\mathcal{S}_{\mathcal{Y}}}$. The proof is omitted here. □

B.9. Proof of Lemma 12

Proof of Lemma 12. Denote $\widehat{g}(u) = \|G^n \circ \mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2 = \|G^n(\widetilde{u}) - \widetilde{u}\|_\infty^2$. We have $\|\widehat{g}\|_{L^\infty(\mathcal{M})} \leq 4R_{\mathcal{X}}^2$. We deduce

$$\begin{aligned}
\mathrm{TX}_2 =& \mathbb{E}_{\mathcal{J}_1}\left[\mathbb{E}_{u\sim\gamma}[\widehat{g}(u)] - \frac{2}{n}\sum_{i=1}^n \widehat{g}(u_i)\right] \\
=& 2\mathbb{E}_{\mathcal{J}_1}\left[\frac{1}{2}\mathbb{E}_{u\sim\gamma}[\widehat{g}(u)] - \frac{1}{n}\sum_{i=1}^n \widehat{g}(u_i)\right] \\
=& 2\mathbb{E}_{\mathcal{J}_1}\left[\mathbb{E}_{u\sim\gamma}[\widehat{g}(u)] - \frac{1}{n}\sum_{i=1}^n \widehat{g}(u_i) - \frac{1}{2}\mathbb{E}_{u\sim\gamma}[f\widehat{g}(u)]\right].
\end{aligned} \tag{75}$$

Note that

$$\begin{aligned}
\mathbb{E}_{u\sim\gamma}[\widehat{g}^2(u)] =& \mathbb{E}_{u\sim\gamma}\left[\|G^n \circ \mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^4\right] \\
=& \mathbb{E}_{u\sim\gamma}\left[\|G^n \circ \mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2 \widehat{g}(u)\right] \\
\leq& \mathbb{E}_{u\sim\gamma}\left[4R_{\mathcal{X}}^2 \widehat{g}(u)\right].
\end{aligned} \tag{76}$$

Using relation (76), we have

$$\mathrm{TX}_2 \leq 2\mathbb{E}_{\mathcal{J}_1}\left[\mathbb{E}_{u\sim\gamma}[\widehat{g}(u)] - \frac{1}{n}\sum_{i=1}^n \widehat{g}(u_i) - \frac{1}{8R_{\mathcal{X}}^2}\mathbb{E}_{u\sim\gamma}[\widehat{g}^2(u)]\right]. \tag{77}$$

Let $\{u_i'\}_{i=1}^n$ be independent copies of $\{u_i\}_{i=1}^n$. Denote the function class

$$\mathcal{G} = \left\{g(u) = \|G_{\mathrm{NN}}' \circ \mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2 | G_{\mathrm{NN}}' \in \mathcal{F}_{\mathrm{NN}}^G\right\}.$$

We have $\|g\|_{L^\infty(\mathcal{M})} \leq 4R_{\mathcal{X}}^2$ for any $g \in \mathcal{G}$. We bound (77) as

$$\begin{aligned}
\mathrm{TX}_2 \leq& 2\mathbb{E}_{\mathcal{J}_1}\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}_{u'\sim\gamma}[g(u')] - \frac{1}{n}\sum_{i=1}^n g(u_i) - \frac{1}{8R_{\mathcal{X}}^2}\mathbb{E}_{u\sim\gamma}[g^2(u)]\right)\right] \\
=& 2\mathbb{E}_{\mathcal{J}_1}\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}_{u'\sim\gamma}\left[\frac{1}{n}\sum_{i=1}^n \left(g(u') - g(u_i)\right)\right] - \frac{1}{16R_{\mathcal{X}}^2}\mathbb{E}_{u',u\sim\gamma}\left[g^2(u') + g^2(u)\right]\right)\right].
\end{aligned} \tag{78}$$

We then consider a δ-cover of $\mathcal{G}$: $\mathcal{G}^* = \{g_i^*\}_{i=1}^{\mathcal{N}(\delta,\mathcal{G},\|\cdot\|_{L^\infty})}$, where $\mathcal{N}(\delta,\mathcal{G},\|\cdot\|_{L^\infty})$ is the covering number. For any $g \in \mathcal{G}$, there is a $g^* \in \mathcal{G}^*$ so that $\|g - g^*\|_{L^\infty} \leq \delta$.

We will derive an upper bound for (78) by replacing g by g^*. Then the problem is converted to analyzing the concentration result on a finite set. First note that

$$\begin{aligned}
g(u') - g(u) =& g(u') - g^*(u') + g^*(u') - g^*(u) + g^*(u) - g(u) \\
\leq& g^*(u') - g^*(u) + 2\delta,
\end{aligned} \tag{79}$$

and

$$\begin{aligned}
g^2(u') + g^2(u) =& \left(g^2(u') - (g^*)^2(u')\right) + \left((g^*)^2(u') + (g^*)^2(u)\right) - \left((g^*)^2(u) - g^2(u)\right) \\
=& (g^*)^2(u') + (g^*)^2(u) + (g(u') - g^*(u'))(g(u') + g^*(u')) - (g^*(u) - g(u))(g^*(u) + g(u)) \\
\geq& (g^*)^2(u') + (g^*)^2(u) - |g(u') - g^*(u')||g(u') + g^*(u')| - |g^*(u) - g(u)||g^*(u) + g(u)| \\
\geq& (g^*)^2(u') + (g^*)^2(u) - 8R_{\mathcal{X}}^2\delta - 8R_{\mathcal{X}}^2\delta \\
=& (g^*)^2(u') + (g^*)^2(u) - 16R_{\mathcal{X}}^2\delta.
\end{aligned} \tag{80}$$

Utilizing (79) and (80) in (78), we get

$$\begin{aligned}
\mathrm{TX}_2 \leq& 2\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\sup_{g\in\mathcal{G}}\left(\frac{1}{n}\sum_{i=1}^n \left(g^*(u_i') - g^*(u_i)\right) - \frac{1}{16R_{\mathcal{X}}^2}\mathbb{E}_{u',u\sim\gamma}\left[(g^*)^2(u') + (g^*)^2(u)\right]\right)\right] + 6\delta \\
=& 2\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\max_j\left(\frac{1}{n}\sum_{i=1}^n \left(g_j^*(u_i') - g_j^*(u_i)\right) - \frac{1}{16R_{\mathcal{X}}^2}\mathbb{E}_{u',u\sim\gamma}\left[(g_j^*)^2(u') + (g_j^*)^2(u)\right]\right)\right] + 6\delta.
\end{aligned} \tag{81}$$

Denote $h_j(i) = g_j^*(u_i') - g_j^*(u_i)$. We have $\mathbb{E}_{u_i',u_i\sim\gamma} h_j(i) = 0$ and

$$\mathrm{Var}[h_j(i)] = \mathbb{E}_{u_i',u_i\sim\gamma}\left[h_j^2(i)\right] = \mathbb{E}_{u_i',u_i\sim\gamma}\left[\left(g_j^*(u_i') - g_j^*(u_i)\right)^2\right] \le 2\mathbb{E}_{u_i',u_i\sim\gamma}\left[(g_j^*)^2(u_i') + (g_j^*)^2(u_i)\right].$$

Thus (81) can be written as

$$\begin{aligned}&\mathrm{TX}_2 \le \mathrm{TX}_2' + 6\delta\\ &\text{with } \mathrm{TX}_2{}' = 2\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\max_j\left(\frac{1}{n}\sum_{i=1}^n h_j(i) - \frac{1}{32R_{\mathcal{X}}^2}\frac{1}{n}\sum_{i=1}^n \mathrm{Var}\left[h_j(i)\right]\right)\right].\end{aligned} \tag{82}$$

We will derive an upper bound for TX_2 using moment generating function. Note that $\|h_j\|_{L^\infty} \le 8R_{\mathcal{X}}^2$. For $0 < t/n < 3/(8R_{\mathcal{X}}^2)$, we have

$$\begin{aligned}\mathbb{E}_{u_i',u_i\sim\gamma}\left[\exp\left(\frac{t}{n}h_j(i)\right)\right] =& \mathbb{E}_{u_i',u_i\sim\gamma}\left[1 + \frac{t}{n}h_j(i) + \sum_{k=2}^\infty \frac{(t/n)^k h_j^k(i)}{k!}\right]\\ \le& \mathbb{E}_{u_i',u_i\sim\gamma}\left[1 + \frac{t}{n}h_j(i) + \sum_{k=2}^\infty \frac{(t/n)^k h_j^2(i)(8R_{\mathcal{X}}^2)^{k-2}}{2\times 3^{k-2}}\right]\\ =& \mathbb{E}_{u_i',u_i\sim\gamma}\left[1 + \frac{t}{n}h_j(i) + \frac{(t/n)^2h_j^2(i)}{2}\sum_{k=2}^\infty \frac{(t/n)^{k-2}(8R_{\mathcal{X}}^2)^{k-2}}{3^{k-2}}\right]\\ =& \mathbb{E}_{u_i',u_i\sim\gamma}\left[1 + \frac{t}{n}h_j(i) + \frac{(t/n)^2h_j^2(i)}{2}\frac{1}{1-8tR_{\mathcal{X}}^2/(3n)}\right]\\ =& 1 + (t/n)^2\mathrm{Var}\left[h_j(i)\right]\frac{1}{2-16R_{\mathcal{X}}^2/3}\\ \le& \exp\left(\mathrm{Var}\left[h_j(i)\right]\frac{3(t/n)^2}{6-48tR_{\mathcal{X}}^2/n}\right),\end{aligned} \tag{83}$$

where the last inequality used the relation $1 + a \le \exp(a)$.

We use (83) to bound $\mathrm{TX}_2{}'$ as

$$\begin{aligned}\exp\left(t\frac{\mathrm{T}_2'}{2}\right) =& \exp\left(t\mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\max_j\left(\frac{1}{n}\sum_{i=1}^n h_j(i) - \frac{1}{32R_{\mathcal{X}}^2}\frac{1}{n}\sum_{i=1}^n\mathrm{Var}\left[h_j(i)\right]\right)\right]\right)\\ \le& \mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\exp\left(t\max_j\left(\frac{1}{n}\sum_{i=1}^n h_j(i) - \frac{1}{32R_{\mathcal{X}}^2}\frac{1}{n}\sum_{i=1}^n\mathrm{Var}\left[h_j(i)\right]\right)\right)\right]\\ \le& \mathbb{E}_{\mathcal{J}_1}\mathbb{E}_{u'\sim\gamma}\left[\sum_j\exp\left(\sum_{i=1}^n\left(\frac{t}{n}h_j(i) - \frac{t}{n}\frac{1}{32R_{\mathcal{X}}^2}\mathrm{Var}\left[h_j(i)\right]\right)\right)\right]\\ \le& \sum_j\exp\left(\sum_{i=1}^n\left(\mathrm{Var}\left[h_j(i)\right]\frac{3(t/n)^2}{6-48tR_{\mathcal{X}}^2/n} - \frac{t}{n}\frac{1}{32R_{\mathcal{X}}^2}\mathrm{Var}\left[h_j(i)\right]\right)\right)\\ =& \sum_j\exp\left(\sum_{i=1}^n\frac{t}{n}\mathrm{Var}\left[h_j^2(i)\right]\left(\frac{3t/n}{6-48tR_{\mathcal{X}}^2/n} - \frac{1}{32R_{\mathcal{X}}^2}\right)\right).\end{aligned} \tag{84}$$

Set $t = \frac{n}{24R_{\mathcal{X}}^2}$ so that $\frac{3t/n}{6-48tR_{\mathcal{X}}^2/n} - \frac{1}{32R_{\mathcal{X}}^2} = 0$. We have

$$t\frac{\mathrm{T}_2'}{2} \le \log\sum_j\exp(0) \Rightarrow \mathrm{T}_2' \le \frac{2}{t}\log\mathcal{N}(\delta,\mathcal{G},\|\cdot\|_{L^\infty}) = \frac{48R_{\mathcal{X}}^2}{n}\log\mathcal{N}(\delta,\mathcal{G},\|\cdot\|_{L^\infty}). \tag{85}$$

We then derive a relation between $\mathcal{N}(\delta,\mathcal{G},\|\cdot\|_{L^\infty})$ and $\mathcal{N}(\delta,\mathcal{F}_{\mathrm{NN}}^G,\|\cdot\|_{\infty,\infty})$. Note that for any $g_1, g_2 \in \mathcal{G}$, there are $G_{\mathrm{NN},1}', G_{\mathrm{NN},2}' \in \mathcal{F}_{\mathrm{NN}}^G$ with $g_1(u) = \|G_{\mathrm{NN},1}'\circ\mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2, g_2(u) = \|G_{\mathrm{NN},2}'\circ\mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2$.

We have

$$\begin{aligned}&\|g_1 - g_2\|_{L^\infty(\mathcal{M})}\\ &= \sup_{u\in\mathcal{M}}\left|\|G_{\mathrm{NN},1}'\circ\mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2 - \|G_{\mathrm{NN},2}'\circ\mathcal{S}_{\mathcal{X}}(u) - \mathcal{S}_{\mathcal{X}}(u)\|_\infty^2\right|\\ &= \sup_{u\in\mathcal{M}}\left|\max_k\left([G_{\mathrm{NN},1}'\circ\mathcal{S}_{\mathcal{X}}(u)]_k - [\mathcal{S}_{\mathcal{X}}(u)]_k\right)^2 - \max_{k'}\left([G_{\mathrm{NN},2}'\circ\mathcal{S}_{\mathcal{X}}(u)]_{k'} - [\mathcal{S}_{\mathcal{X}}(u)]_{k'}\right)^2\right|\end{aligned}$$

$$\begin{aligned}
&\le \sup_{u\in\mathcal{M}} \left|\max_k \left[\left([G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u)]_k - [\mathcal{S}_{\mathcal{X}}(u)]_k\right)^2 - \left([G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u)]_k - [\mathcal{S}_{\mathcal{X}}(u)]_k\right)^2\right]\right| \\
&= \sup_{u\in\mathcal{M}} \left|\max_k \left[\left([G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u)]_k - [G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u)]_k\right)\left([G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u)]_k + [G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u)]_k - 2[\mathcal{S}_{\mathcal{X}}(u)]_k\right)\right]\right| \\
&\le \sup_{u\in\mathcal{M}} \left\|G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u) - G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u)\right\|_\infty \left\|G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u) + G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u) - 2\mathcal{S}_{\mathcal{X}}(u)\right\|_\infty \\
&\le 4R_{\mathcal{X}} \sup_{u\in\mathcal{M}} \left\|G'_{\mathrm{NN},1}\circ\mathcal{S}_{\mathcal{X}}(u) - G'_{\mathrm{NN},2}\circ\mathcal{S}_{\mathcal{X}}(u)\right\|_\infty \\
&= 4R_{\mathcal{X}} \left\|G'_{\mathrm{NN},1} - G'_{\mathrm{NN},2}\right\|_{\infty,\infty}. \qquad (86)
\end{aligned}$$

Substituting (86) and (85) into (82) gives rise to

$$\mathrm{TX}_2 \le \frac{48R_{\mathcal{X}}^2}{n}\log\mathcal{N}\left(\frac{\delta}{4R_{\mathcal{X}}}, \mathcal{F}^G_{\mathrm{NN}}, \|\cdot\|_{\infty,\infty}\right) + 6\delta.$$

The lemma is proved. □

Appendix C. Basic properties about $\|\cdot\|_{\mathcal{S}_{\mathcal{X}}}$ and $\|\cdot\|_{\mathcal{S}_{\mathcal{Y}}}$

In this section, we provide some basic properties of $\|\cdot\|_{\mathcal{S}_{\mathcal{X}}}$ and $\|\cdot\|_{\mathcal{S}_{\mathcal{Y}}}$. These properties will be used frequently in the proof of our main results.

Lemma 14. *Suppose Assumption 1 holds. The discretization operator $\mathcal{S}_{\mathcal{X}}, \mathcal{S}_{\mathcal{Y}}$ is Lipschitz with Lipschitz constant 2:*

$$\|\mathcal{S}_{\mathcal{X}}(u_1) - \mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}} \le 2\|u_1 - u_2\|_{\mathcal{X}}, \quad \textit{and} \quad \|\mathcal{S}_{\mathcal{Y}}(v_1) - \mathcal{S}_{\mathcal{Y}}(v_2)\|_{\mathcal{S}_{\mathcal{Y}}} \le 2\|v_1 - v_2\|_{\mathcal{Y}}$$

for any $u_1, u_2 \in \mathcal{X}, v_1, v_2 \in \mathcal{Y}$.

Proof of Lemma 14. For any $u_1, u_2 \in \mathcal{X}$, we have $\|\mathcal{S}_{\mathcal{X}}(u_1) - \mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}} = \|\mathcal{S}_{\mathcal{X}}(u_1 - u_2)\|_{\mathcal{S}_{\mathcal{X}}} \le 2\|u_1 - u_2\|_{\mathcal{X}}$. The case for $\mathcal{S}_{\mathcal{Y}}$ can be proved similarly. □

Lemma 15. *The operation $\langle\cdot,\cdot\rangle_{\mathcal{S}_{\mathcal{X}}}$ and $\langle\cdot,\cdot\rangle_{\mathcal{S}_{\mathcal{Y}}}$ satisfies*

$$\begin{aligned}
&|\langle\mathcal{S}_{\mathcal{X}}(u_1), \mathcal{S}_{\mathcal{X}}(u_2)\rangle_{\mathcal{S}_{\mathcal{X}}}| \le \|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}\|\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}}, \\
&|\langle\mathcal{S}_{\mathcal{Y}}(v_1), \mathcal{S}_{\mathcal{Y}}(v_2)\rangle_{\mathcal{S}_{\mathcal{Y}}}| \le \|\mathcal{S}_{\mathcal{Y}}(v_1)\|_{\mathcal{S}_{\mathcal{Y}}}\|\mathcal{S}_{\mathcal{Y}}(v_2)\|_{\mathcal{S}_{\mathcal{Y}}}
\end{aligned}$$

for any $u_1, u_2 \in \mathcal{X}$ and $v_1, v_2 \in \mathcal{Y}$.

Proof of Lemma 15. We prove the inequality for $\langle\cdot,\cdot\rangle_{\mathcal{S}_{\mathcal{X}}}$. The inequality for $\mathcal{S}_{\mathcal{Y}}$ can be proved similarly. Denote $\widetilde{\mathbf{u}} = [w_1^{1/2}u(\mathbf{x}_1), \ldots, w_{D_1}^{1/2}u(\mathbf{x}_{D_1})]^\top$. By Hölder's inequality, we have

$$\begin{aligned}
|\langle\mathcal{S}_{\mathcal{X}}(u_1), \mathcal{S}_{\mathcal{X}}(u_2)\rangle_{\mathcal{S}_{\mathcal{X}}}| &= \sum_{i=1}^{D_1} w_i u_1(\mathbf{x}_i)u_2(\mathbf{x}_i) = \left|\sum_{i=1}^{D_1}(w_i^{1/2}u_1(\mathbf{x}_i))(w_i^{1/2}u_2(\mathbf{x}_i))\right| \\
&= |\langle\widetilde{\mathbf{u}}_1, \widetilde{\mathbf{u}}_2|\rangle \le \|\widetilde{\mathbf{u}}_1\|_2\|\widetilde{\mathbf{u}}_2\|_2 = \sqrt{\sum_{i=1}^{D_1} w_i u_1^2(\mathbf{x}_i)}\sqrt{\sum_{i=1}^{D_1} w_i u_2^2(\mathbf{x}_i)} = \|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}\|\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}}. \quad \square
\end{aligned}$$

Lemma 16. *$\|\cdot\|_{\mathcal{S}_{\mathcal{X}}}$ and $\|\cdot\|_{\mathcal{S}_{\mathcal{Y}}}$ are norms in $\mathbb{R}^{D_1}$ and $\mathbb{R}^{D_2}$, respectively.*

Proof of Lemma 16. For any $u \in \mathcal{X}$, we have $\|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}} \ge 0$ since the w_i's are positive. Furthermore, $\|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}} = 0$ if and only if $\mathcal{S}_{\mathcal{X}}(u) = \mathbf{0}$. For any $\lambda \in \mathbb{R}$, we have

$$\|\mathcal{S}_{\mathcal{X}}(\lambda u)\|_{\mathcal{S}_{\mathcal{X}}} = \sqrt{\sum_{i=1}^{D_1} w_i\lambda^2 u(\mathbf{x}_i)^2} = |\lambda|\|\mathcal{S}_{\mathcal{X}}(u)\|_{\mathcal{S}_{\mathcal{X}}}.$$

We next prove the triangle inequality: For any $u_1, u_2 \in \mathcal{X}$, we have

$$\begin{aligned}
&\|\mathcal{S}_{\mathcal{X}}(u_1) + \mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}} \\
&= \sqrt{\langle\mathcal{S}_{\mathcal{X}}(u_1) + \mathcal{S}_{\mathcal{X}}(u_2), \mathcal{S}_{\mathcal{X}}(u_1) + \mathcal{S}_{\mathcal{X}}(u_2)\rangle_{\mathcal{X}}}
\end{aligned}$$

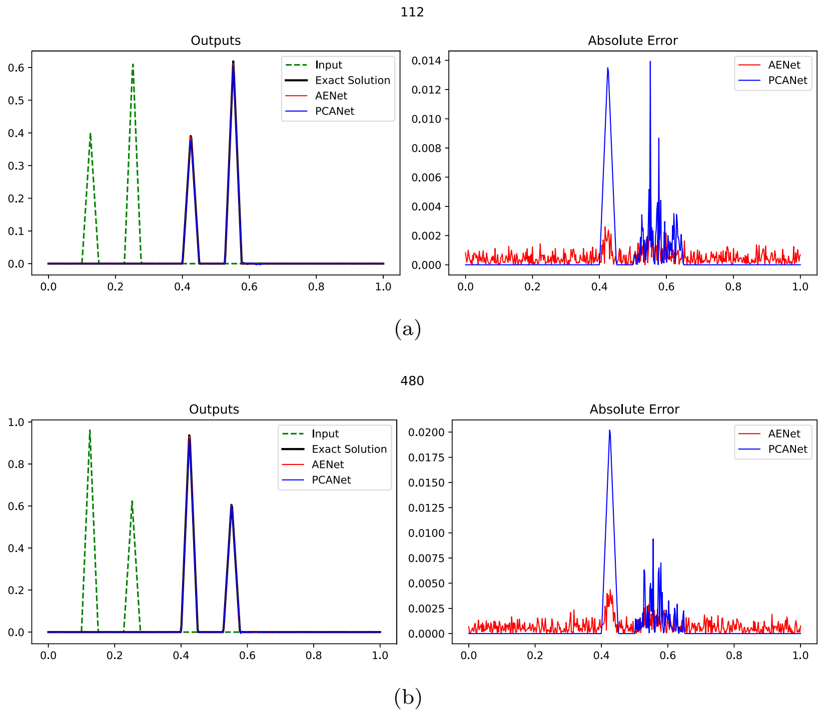


Fig. 20. Comparison of PCANet and AENet on transport data. The subplot number indicates the test example number which was randomly selected.

$$=\sqrt{\sum_{i=1}^{D_1} w_i(u_1(\mathbf{x}_i)+u_2(\mathbf{x}_i))^2}=\sqrt{\sum_{i=1}^{D_1} w_i u_1^2(\mathbf{x}_i)+2\sum_{i=1}^{D_1} w_i u_1(\mathbf{x}_i)u_2(\mathbf{x}_i)+\sum_{i=1}^{D_1} w_i u_2(\mathbf{x}_i)^2)}$$
$$=\sqrt{\|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}^2+2\langle \mathcal{S}_{\mathcal{X}}(u_1),\mathcal{S}_{\mathcal{X}}(u_2)\rangle_{\mathcal{S}_{\mathcal{X}}}+\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}}^2}$$
$$\leq\sqrt{\|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}^2+2\|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}\|\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}}+\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}}^2}=\|\mathcal{S}_{\mathcal{X}}(u_1)\|_{\mathcal{S}_{\mathcal{X}}}+\|\mathcal{S}_{\mathcal{X}}(u_2)\|_{\mathcal{S}_{\mathcal{X}}},$$

where we use Lemma 15 in the first inequality. The result about $\|\cdot\|\mathcal{S}_{\mathcal{Y}}$ can be proved similarly. $\square$

Appendix D. Experimental figures

D.1. Comparison to PCANet figures

Fig. 20 shows the predicted output of AENet and PCANet on various input data for the transport equation. As in the main paper, we use a depth 3 width 500 feedforward architecture for the components of AENet and PCANet, and we consider a fixed reduced dimension of 2, see Section 5 for more details. Figs. 21 and 22 are analogous plots for the Burgers' and KdV equation respectively.

D.2. CNN figures

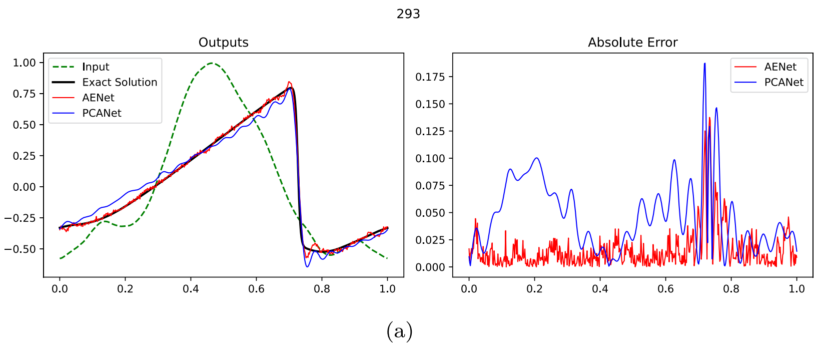
Let $C_1, C_2, K, S, P, D \in \mathbb{N}$. A (one-dimensional) **convolution** layer with C_1 input channels, C_2 output channels, kernel size K, stride S, (two-sided) padding P, and input dimension D is a function $C_{w,b}: \mathbb{R}^{C_1\times D}\to\mathbb{R}^{C_2\times D'}$ where $D' = 1+\left\lfloor \frac{d+2P-K}{S}\right\rfloor$ that is associated with a "kernel" $w\in\mathbb{R}^{C_2\times C_1\times K}$ and bias $b\in\mathbb{R}^{C_2\times D'}$ such that (where $*$ is defined in the equation)

$$(C_w(x))_{i,j}=b_{i,j}+\sum_{c=1}^{C_1} w_{i,c,\cdot}*x'_{c,\cdot}=b_{i,j}+\sum_{c=1}^{C_1}\sum_{k=1}^{K} w_{i,c,k}\cdot x'_{c,S\cdot j+k-1},$$

where $x'\in\mathbb{R}^{C_1\times(D+2P)}$ is obtained by padding P zero entries on both sides of each channel of x', i.e.

$$(x')_{c,i}=x_{c,i-p} \text{ if } p<i\leq d+p \text{ otherwise } (x')_{c,i}=0.$$

After applying the convolution layer, we will subsequently apply a max pooling layer which subsamples the entry of each channel by the maximum value in a sliding window. Let $C, K_{max}, S, D\in\mathbb{N}$. A **max pooling** layer with C channels, kernel size K_{max}, and input dimension D is a function $\mathcal{P}_{max}:\mathbb{R}^{C\times D}\to\mathbb{R}^{C\times D'}$ where $D'=\lfloor\frac{D}{K_{max}}\rfloor$ such that



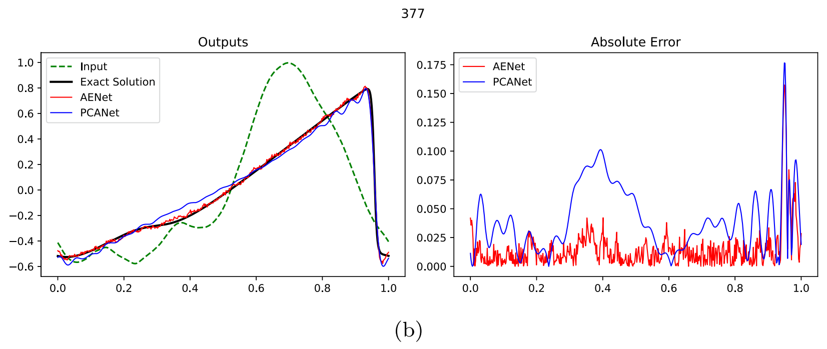
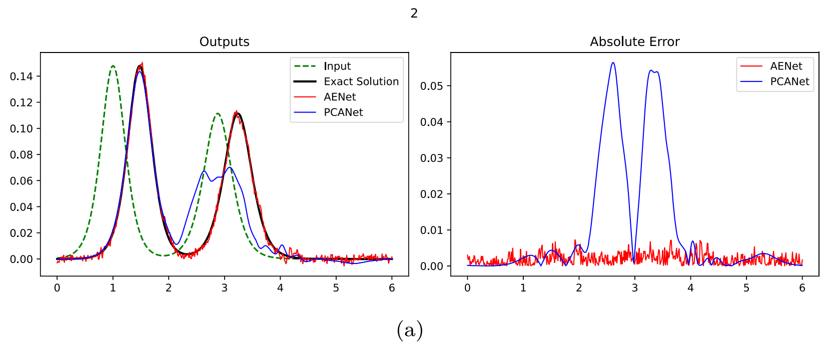


Fig. 21. Comparison of PCANet and AENet on Burgers' data. The subplot number indicates the test example number which was randomly selected.



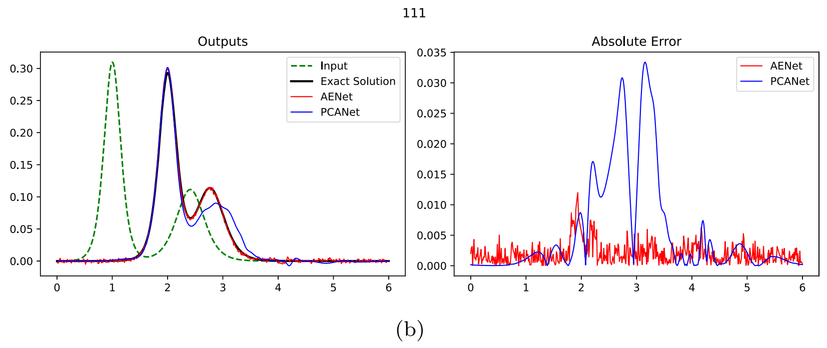
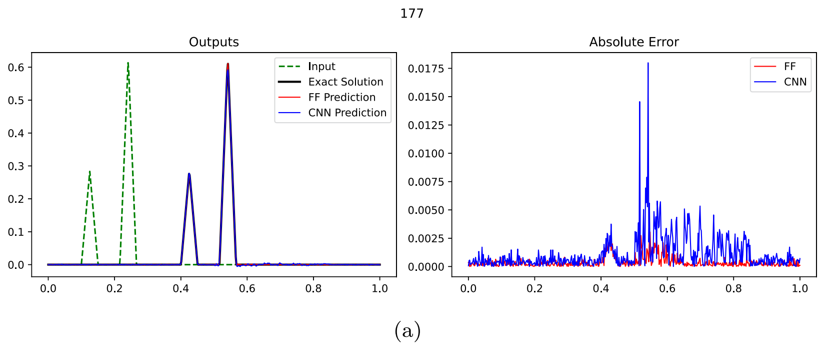


Fig. 22. Comparison of PCANet and AENet on KdV data. The subplot number indicates the test example number which was randomly selected.

$$\mathcal{P}_{max}(x)_{i,j} = \max_{k=1,\ldots,K_{max}} \left\{ x_{i,K_{max}\cdot(j-1)+k} \right\}.$$

For the encoder component of the autoencoder, we use convolution layers with kernel size $K = 8$, padding $P = 8$, and stride $S = 2$, followed by max pooling layers with kernel size $K_{max} = 2$. Let $d_{in}, d_{out}, L, K, K_{max} \in \mathbb{N}$. A (ReLU) **convolutional neural network**



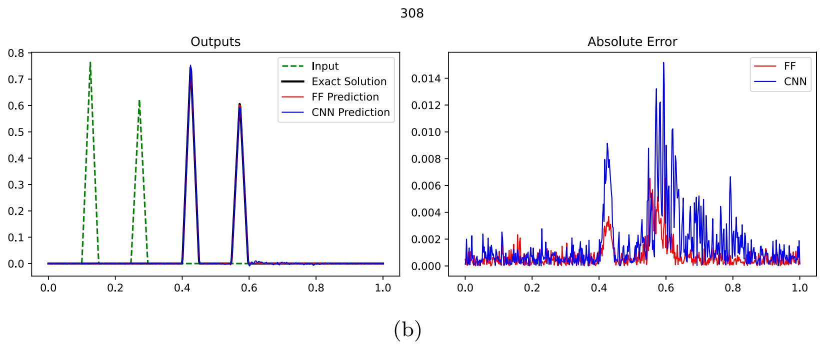


Fig. 23. Comparison of convolution AENet and feedforward AENet on transport test examples. The subplot number indicates the test example number which was randomly selected.

(CNN) with input dimension d_{in}, output dimension d_{out}, depth L, kernel size K, and max pooling kernel size K_{max} consists of a composition of convolutional layers, max pooling layers, and the ReLU activation function L times. Specifically, for $1 \le l \le L$, define the kernel weights $w^l \in \mathbb{R}^{C_l \times C_{l-1} \times K}$ and bias $b^l \in \mathbb{R}^{C_l \times D_l}$ where $C_l = 2^{l+3}$ for $l > 0$ and $C_0 = 1$ for number of channels. After the convolution layers, we flatten the result which has C_L channels and D_L entries, and then multiply by a matrix $W \in \mathbb{R}^{d_{out} \times C_L D_L}$ and add a bias $B \in \mathbb{R}^{d_{out}}$ to obtain a vector of output dimension d_{out}. Thus the CNN is a function of the form (where σ is ReLU):

$$x \mapsto B + W \cdot \left(\sigma \circ \mathcal{P}_{max} \circ \mathcal{C}_{w^L} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{P}_{max} \circ \mathcal{C}_{w^1}\right)(x)$$

Now let $C_1, C_2, K, S, H, D \in \mathbb{N}$. A (one-dimensional) **transpose convolutional layer** with C_1 input channels, C_2 output channels, kernel size K, stride S, output padding H, and input dimension D is a function $\mathcal{C}^{\top}_{w,b} : \mathbb{R}^{C_1 \times D} \to \mathbb{R}^{C_2 \times D'}$ where $D' = S(D-1) + H$ that is associated with a "kernel" $w \in \mathbb{R}^{C_2 \times C_1 \times K}$ and bias $b \in \mathbb{R}^{C_2 \times D'}$ such that (where $*'$ is defined in the equation)

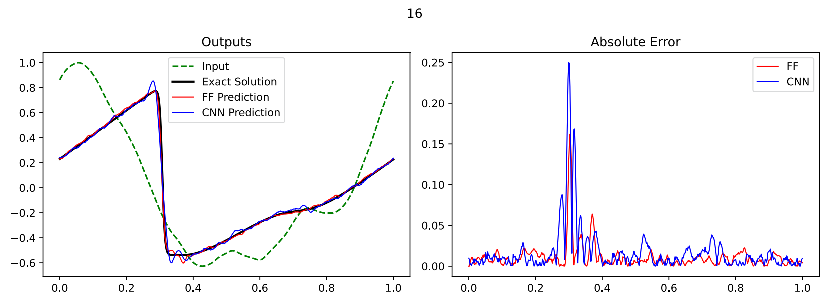
$$(\mathcal{C}^{\top}_{w,b}(x'))_{i,j} = b_{i,j} + \sum_{c=1}^{C_1} w_{i,c,\cdot} *' x'_{c,\cdot} = b_{i,j} + \sum_{c=1}^{C_1} \sum_{k=1}^{K} w_{i,c,k} \cdot x'_{c,S\cdot j-k+1},$$

for $1 \le j \le D' - H$, and $(\mathcal{C}^{\top}_{w,b}(x))_{i,j} = 0$ otherwise (this is output padding by H). Here, $x' \in \mathbb{R}^{C_1 \times (D+2P)}$ is obtained by adding $\lfloor K/2 \rfloor$ entries with value 0 on both sides of each channel of x'.

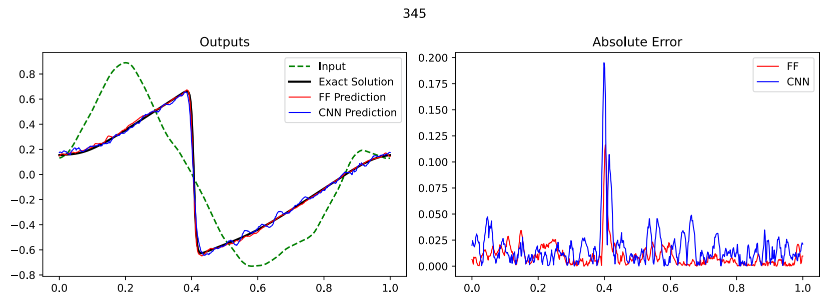
For the decoder component of the Auto-Encoder and the solution network, we use transpose convolution layers with kernel size $K = 8$, output padding $H = 1$, and stride $S = 2$. Let $d_{in}, d_{out}, L, K \in \mathbb{N}$. A (ReLU) **transpose convolutional neural network** (TCNN) with input dimension d_{in}, output dimension d_{out}, depth L, and kernel size K is a composition of transpose convolutional layers with the ReLU activation function L times. Specifically, for $1 \le l \le L$, define the kernel weights $w^l \in \mathbb{R}^{C_l \times C_{l-1} \times K}$ and bias $b^l \in \mathbb{R}^{C_l \times D_l}$, where $C_l = 2^{l+3}$ for $l > 0$ and $C_0 = 1$ for number of channels. After the transpose convolution layers, we flatten the result which has C_L channels and D_L entries, and then multiply by a matrix $W \in \mathbb{R}^{k \times C_L D_L}$ and add a bias $B \in \mathbb{R}^{d_{out}}$ to obtain a vector of output dimension d_{out}. Thus the TCNN is a function of the form (where σ is ReLU):

$$x \mapsto B + W \cdot \left(\sigma \circ \mathcal{C}^{\top}_{w^L} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^{\top}_{w^1}\right)(x).$$

Fig. 23 shows the predicted output of a convolutional AENet on various input data for the transport equation. We compare the output of the convolutional AENet architecture (3 layers with kernel size 8) with the feedforward AENet architecture. Figs. 24 and 25 are analogous plots for the Burgers' and KdV equation respectively.

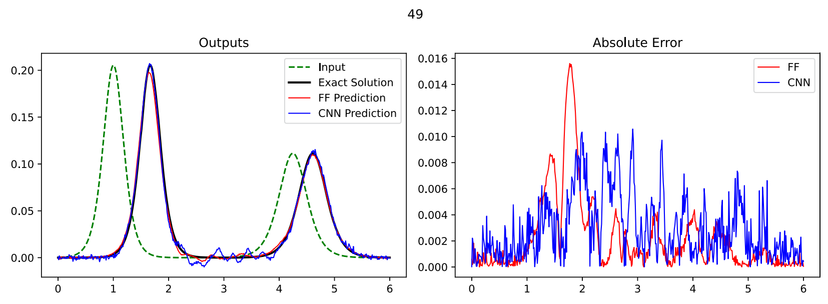


(a)

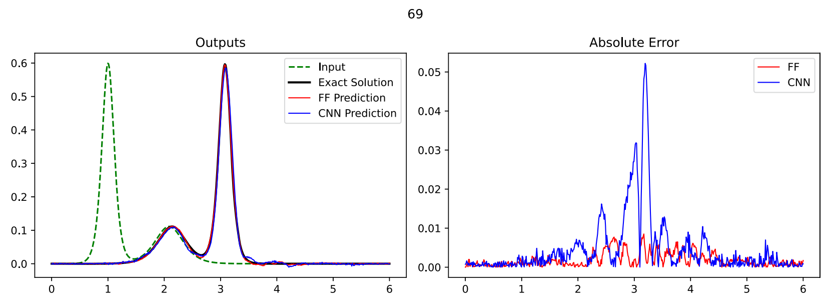


(b)

Fig. 24. Comparison of convolution AENet and feedforward AENet on Burgers test examples. The subplot number indicates the test example number which was randomly selected.



(a)



(b)

Fig. 25. Comparison of convolution AENet and feedforward AENet on KdV test examples. The subplot number indicates the test example number which was randomly selected.

Data availability

Data will be made available on request.

References

[1] A. Anandkumar, K. Azizzadenesheli, K. Bhattacharya, N. Kovachki, Z. Li, B. Liu, A. Stuart, Neural operator: graph kernel network for partial differential equations, in: ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, 2020, https://openreview.net/forum?id=fg2ZFmXFO3.
[2] K. Atkinson, An Introduction to Numerical Analysis, John Wiley & Sons, 1991.
[3] F. Bartolucci, E. de Bezenac, B. Raonic, R. Molinaro, S. Mishra, R. Alaifari, Representation equivalent neural operators: a framework for alias-free operator learning, Adv. Neural Inf. Process. Syst. 36 (2024).
[4] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (2015) 483–531.
[5] P. Benner, M. Ohlberger, A. Cohen, K. Willcox, Model Reduction and Approximation: Theory and Algorithms, SIAM, 2017.
[6] K. Bhattacharya, B. Hosseini, N.B. Kovachki, A.M. Stuart, Model reduction and neural networks for parametric pdes, SMAI J. Comput. Math. 7 (2021) 121–157.
[7] K. Carlberg, C. Farhat, A low-cost, goal-oriented 'compact proper orthogonal decomposition' basis for model reduction of static systems, Int. J. Numer. Methods Eng. 86 (2011) 381–402.
[8] 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) 8174–8184.
[9] M. Chen, H. Jiang, W. Liao, T. Zhao, Nonparametric regression on low-dimensional manifolds using deep relu networks: function approximation and statistical recovery, Inf. Inference 11 (2022) 1203–1253.
[10] T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, IEEE Trans. Neural Netw. 6 (1995) 911–917.
[11] 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.
[12] E. Facco, M. d'Errico, A. Rodriguez, A. Laio, Estimating the intrinsic dimension of datasets by a minimal neighborhood information, Sci. Rep. 7 (2017) 12140.
[13] A.L. Ferguson, A.Z. Panagiotopoulos, P.G. Debenedetti, I.G. Kevrekidis, Systematic determination of order parameters for chain dynamics using diffusion maps, Proc. Natl. Acad. Sci. 107 (2010) 13597–13602.
[14] A.L. Ferguson, A.Z. Panagiotopoulos, I.G. Kevrekidis, P.G. Debenedetti, Nonlinear dimensionality reduction in molecular simulation: the diffusion map approach, Chem. Phys. Lett. 509 (2011) 1–11.
[15] N. Franco, A. Manzoni, P. Zunino, A deep learning approach to reduced order modelling of parameter dependent partial differential equations, Math. Comput. 92 (2023) 483–524.
[16] S. Fresca, L. Dede, A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes, J. Sci. Comput. 87 (2021) 1–36.
[17] F.J. Gonzalez, M. Balajewicz, Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, arXiv preprint arXiv:1808.01346, 2018.
[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[19] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013.
[20] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017.
[21] J. Han, A. Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (2018) 8505–8510.
[22] J.S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78.
[23] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, 2012.
[24] H. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educ. Psychol. 24 (1933) 417.
[25] Y. Khoo, J. Lu, L. Ying, Solving parametric pde problems with artificial neural networks, Eur. J. Appl. Math. 32 (2021) 421–435.
[26] Y. Khoo, L. Ying, Switchnet: a neural network model for forward and inverse scattering problems, SIAM J. Sci. Comput. 41 (2019) A3182–A3201.
[27] Y. Kim, Y. Choi, D. Widemann, T. Zohdi, Efficient nonlinear manifold reduced order model, arXiv preprint arXiv:2011.07727, 2020.
[28] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends Mach. Learn. 12 (2019) 307–392.
[29] M. Kirszbraun, Über die zusammenziehende und lipschitzsche transformationen, Fundam. Math. 22 (1934) 77–108.
[30] K. Kontolati, S. Goswami, G.E. Karniadakis, M.D. Shields, Learning in latent spaces improves the predictive accuracy of deep neural operators, arXiv preprint arXiv:2304.07599, 2023.
[31] N. Kovachki, S. Lanthaler, S. Mishra, On universal approximation and error bounds for Fourier neural operators, J. Mach. Learn. Res. 22 (2021).
[32] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: learning maps between function spaces, arXiv preprint arXiv:2108.08481, 2021.
[33] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (1991) 233–243.
[34] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012.
[35] S. Lanthaler, S. Mishra, G.E. Karniadakis, Error estimates for deeponets: a deep learning framework in infinite dimensions, Trans. Math. Appl. 6 (2022) tnac001.
[36] S. Lanthaler, A.M. Stuart, The curse of dimensionality in operator learning, arXiv preprint arXiv:2306.15924, 2023.
[37] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, J. Comput. Phys. 404 (2020) 108973.
[38] E. Levina, P. Bickel, Maximum likelihood estimation of intrinsic dimension, Adv. Neural Inf. Process. Syst. 17 (2004).
[39] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895, 2020.
[40] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. 807 (2016) 155–166.
[41] 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, 2021.
[42] H. Liu, A. Havrilla, R. Lai, W. Liao, Deep nonparametric estimation of intrinsic data structures by chart autoencoders: generalization error and robustness, arXiv preprint arXiv:2303.09863, 2023.
[43] 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.
[44] J. Lu, Z. Shen, H. Yang, S. Zhang, Deep network approximation for smooth functions, SIAM J. Math. Anal. 53 (2021) 5465–5506.

[45] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2021) 218–229.
[46] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis, DeepXDE: a deep learning library for solving differential equations, SIAM Rev. 63 (2021) 208–228.
[47] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinform. 19 (2017) 1236–1246.
[48] R. Nakada, M. Imaizumi, Adaptive approximation and generalization of deep neural network with intrinsic dimensionality, J. Mach. Learn. Res. 21 (2020) 1–38.
[49] G. Ongie, A. Jalal, C.A. Metzler, R.G. Baraniuk, A.G. Dimakis, R. Willett, Deep learning techniques for inverse problems in imaging, IEEE J. Sel. Areas Inf. Theory 1 (2020) 39–56.
[50] S.E. Otto, C.W. Rowley, Linearly recurrent autoencoder networks for learning dynamics, SIAM J. Appl. Dyn. Syst. 18 (2019) 558–593.
[51] T. O'Leary-Roseberry, U. Villa, P. Chen, O. Ghattas, Derivative-informed projected neural networks for high-dimensional parametric maps governed by pdes, Comput. Methods Appl. Mech. Eng. 388 (2022) 114199.
[52] P. Petersen, F. Voigtlaender, Equivalence of approximation by convolutional neural networks and fully-connected networks, Proc. Am. Math. Soc. 148 (2020) 1567–1581.
[53] C. Prud'Homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods, J. Fluids Eng. 124 (2002) 70–80.
[54] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[55] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[56] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (2008) 229–275.
[57] S. Schonsheck, J. Chen, R. Lai, Chart auto-encoders for manifold structured data, arXiv preprint arXiv:1912.10094, 2019.
[58] J. Seidman, G. Kissas, P. Perdikaris, G.J. Pappas, Nomad: nonlinear manifold decoders for operator learning, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, in: Curran Associates, vol. 35, Inc., 2022, https://proceedings.neurips.cc/paper_files/paper/2022/file/24f49b2ad9fbe65eefbfd99d6f6c3fd2-Paper-Conference.pdf.
[59] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
[60] J. Sirignano, K. Spiliopoulos, Dgm: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364.
[61] T. Suzuki, Adaptivity of deep relu network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality, arXiv preprint arXiv:1810.08033, 2018.
[62] R. Tang, Y. Yang, On empirical Bayes variational autoencoder: an excess risk bound, in: Conference on Learning Theory, in: PMLR, 2021.
[63] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[64] Q. Wang, J.S. Hesthaven, D. Ray, Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem, J. Comput. Phys. 384 (2019) 289–307.
[65] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Int. J. Numer. Methods Fluids 86 (2018) 255–268.
[66] Y. Xian, X. Sun, W. Liao, Y. Zhang, D. Nowacek, L. Nolte, Intrinsic structure study of whale vocalizations, in: OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016.
[67] D. Yarotsky, Error bounds for approximations with deep relu networks, Neural Netw. 94 (2017) 103–114.
[68] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, J. Comput. Phys. 411 (2020) 109409.
[69] Z. Zhang, L. Wing Tat, H. Schaeffer, Belnet: basis enhanced learning, a mesh-free neural operator, Proc. R. Soc. A 479 (2023) 20230043.