# STATIONARY DENSITY ESTIMATION OF ITÔ DIFFUSIONS USING DEEP LEARNING

**Yiqi Gu**
Department of Mathematics
National University of Singapore, 10 Lower Kent Ridge Road, Singapore, 119076
(`matguy@nus.edu.sg`)

**John Harlim**
Department of Mathematics, Department of Meteorology and Atmospheric Science,
Institute for Computational and Data Sciences
The Pennsylvania State University, University Park, PA 16802, USA
`jharlim@psu.edu`

**Senwei Liang**
Department of Mathematics, Purdue University, IN 47907, USA
`liang339@purdue.edu`

**Haizhao Yang**
Department of Mathematics, Purdue University, IN 47907, USA
`haizhao@purdue.edu`

September 8, 2021

## ABSTRACT

In this paper, we consider the density estimation problem associated with the stationary measure of ergodic Itô diffusions from a discrete-time series that approximate the solutions of the stochastic differential equations. To take an advantage of the characterization of density function through the stationary solution of a parabolic-type Fokker-Planck PDE, we proceed as follows. First, we employ deep neural networks to approximate the drift and diffusion terms of the SDE by solving appropriate supervised learning tasks. Subsequently, we solve a steady-state Fokker-Plank equation associated with the estimated drift and diffusion coefficients with a neural-network-based least-squares method. We establish the convergence of the proposed scheme under appropriate mathematical assumptions, accounting for the generalization errors induced by regressing the drift and diffusion coefficients, and the PDE solvers. This theoretical study relies on a recent perturbation theory of Markov chain result that shows a linear dependence of the density estimation to the error in estimating the drift term, and generalization error results of nonparametric regression and of PDE regression solution obtained with neural-network models. The effectiveness of this method is reflected by numerical simulations of a two-dimensional Student's t distribution and a 20-dimensional Langevin dynamics.

***Keywords*** Stochastic differential equations · Data-driven method · Deep neural network · Fokker-Plank equation

# 1 Introduction

Many phenomena subject to random perturbations can be modeled by stochastic differential equations (SDEs) driven by Brownian noises. Under some regularity assumption, the time evolution of the probability measure can be characterized by the Fokker-Planck equation, a parabolic partial differential equation that depicts the time evolution of the density function of the underlying stochastic processes. Despite its wide applications in modeling physical or biological systems. [53, 25, 15, 4, 20], solving the Fokker-Planck PDE associated to high-dimensional Itô diffusion processes is computationally a challenging task. In this paper, we are interested in estimating the density function associated with the stationary solution of the Fokker-Planck PDE from a discrete-time series of approximate solutions of the underlying SDEs without knowing the explicit drift and diffusion components.

Density estimation is a long-standing problem in computational statistics and machine learning. Among the existing approaches, it is widely accepted that the classical Kernel Density Estimation (KDE) [54] is not effective for problems with dimension higher than three (see e.g., [24, 39, 65]). Along this line, the kernel embedding (another class of linear estimator) also suffered from the curse of dimension [71]. Another class of popular parametric density estimators is the Gaussian Mixture Models (which is also known as the Radial Basis Models in some literature) [24]. This class of approaches is considered as a nonlinear estimator method since the training involves the minimization of a loss function that depends nonlinearly on the latent parameters. A practical issue of such a convex nonlinear optimization problem is the difficulty in identifying the global minimizer using numerical methods. While this issue is not solved, recent advances in deep learning theory show that the deep neural network (DNN), as a composition of multiple linear transformations and simple nonlinear activation functions, has the capacity of approximating various kinds of functions, overcoming or mitigating the curse of dimensionality [46, 14, 48, 51, 67, 37, 47, 23, 58]. Besides, it is shown that with over-parametrization and random initialization, the DNN-based least square optimization achieves a global minimizer by gradient descent with a linear convergence rate in both the setting of regression [27, 11, 68, 7, 43, 40, 9, 8] and PDE solvers [41, 34]. In parallel to this finding, several density estimators have adopted DNN, such as the Neural Autoregressive Distribution Estimation [63] and its variant, the Masked Autoregressive Flow [50].

Building on these encouraging results, we consider solving the density estimation problem where the target function is the density associated with the stationary measure of an Itô process. With this prior knowledge, we propose to solve the density estimation problem following these two steps. First, we employ a deep learning algorithm to solve appropriate supervised learning tasks to uncover the drift and diffusion coefficients of the SDEs. Second, we solve the stationary Fokker-Planck PDE generated from the estimated drift and diffusion coefficients. While traditional grid-based numerical methods, such as finite element methods and finite difference methods [61, 32, 56] can be employed to solve the Fokker-Plank equation, they are usually limited to low-dimensional problems. On the other hand, neural network-based methods has been successfully used in solving high dimensional PDEs [29, 18, 36, 69, 52, 30, 70, 16], including the recent application in solving the high-dimensional Fokker-Plank equation [66, 70, 35]. These successes encourage us to also use deep learning to solve the approximate Fokker-Planck PDE.

We will also develop a new theory for the proposed approach with numerical verifications on low and relatively high-dimensional test examples, especially when the parameters of the Fokker-Planck equations have to be estimated, which has not been considered in the literature. Our theory can also explain and support the empirical success of existing deep learning approaches lacking the theoretical analysis of deep learning. The main goals of this theoretical study are to 1) understand under which mathematical assumptions can the density estimation problem be well-posed, 2) establish the convergence of the proposed scheme, and 3) identify the error in terms of training sample size, width/length of the neural-network models, discretization time step and noise amplitudes in the training data, and the dimension of the stochastic processes. In conjunction, we will also verify whether the perturbation theory [72] is valid. Particularly, we will check whether the stochastic process associated with the estimated drift and diffusion terms (obtained from deep learning regression in the first step) can indeed estimate the underlying invariant measure accurately. This verification is a by-product that can practically be used to generate more samples if needed.

The organization of this paper is as follows. In Section 2, we introduce the problem of stationary density estimation associated with Itô diffusions. In Section 3, the deep learning method is discussed. In Section 4, we provide the convergence theoretical analysis. In Section 5, we present the numerical experiments of Student's distribution and Langevin dynamics. We conclude the paper with some remarks and open questions in Section 6. To improve the readability, we report the proofs of the lemmas of Section 4 in Appendix A.

## 2 Problem Setup

Consider the following SDE,

$$dX_t = \boldsymbol{a}(X_t)\,dt + \boldsymbol{b}(X_t)\,dW_t, \tag{1}$$

with an initial condition randomly drawn from an arbitrary well-defined distribution, $X_0 \sim \pi_0$. The SDE in (1) is defined with a drift term, $\boldsymbol{a}: \mathbb{R}^d \to \mathbb{R}^d$ and a diffusion tensor, $\boldsymbol{b}: \mathbb{R}^d \to \mathbb{R}^{d \times m}$, where $m \le d$. Here, $W_t$ denotes the standard $m-$dimensional Wiener process. We assume that $\boldsymbol{a}$ and $\boldsymbol{b}$ are globally Lipschitz such that the SDE in (1) with the initial condition $X_0 = x$ has a unique solution. In addition, we also assume that the Markov process $X_t$ is ergodic. This implies that the transition kernel corresponding to the Markov process $X_t$ converges to a unique stationary measure $\pi$ as $t \to \infty$. When the probability measure $\pi$ is absolutely continuous with respect to the Lebesque measure, $d\pi(x) = p(x)\,dx$, the density function $p: \mathbb{R}^d \to \mathbb{R}$ is the solution of the stationary Fokker-Planck equation,

$$\mathscr{L}^* p := -\mathrm{div}(\boldsymbol{a}p) + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} ((\boldsymbol{b}\boldsymbol{b}^\top)_{ij} p) = 0, \tag{2}$$

where $p \ge 0$ and $\int_{\mathbb{R}^d} p(x)\,dx = 1$. We will state these (and additional) assumptions in Section 4 for the convergence analysis study.

In this work, we aim to estimate the stationary density $p$ of the SDE (1) without the knowledge of $\boldsymbol{a}$ and $\boldsymbol{b}$. What is available is a time series $\{\boldsymbol{x}^n\}_{n\ge 0}$ generated by a numerical SDE solver of (1) that is assumed to possess an ergodic invariant measure, $\tilde{\pi}$, whose "distance" from $\pi$ can be controlled by the numerical discretization time step $\delta t$. We should point out that when $\boldsymbol{a}$ is globally Lipschitz and $\boldsymbol{b}$ is a full rank matrix and if the underlying Markov process in $X_t$ in (1) is geometrically ergodic, then the Markov chain $\{\boldsymbol{x}^n\}$ induced by the Euler-Maruyama discretization is also geometrically ergodic [42]. In Section 4, we will restrict our convergence study to this case. In a less stringent case, e.g., $\boldsymbol{a}$ is locally Lipschitz, the Markov chain induced by EM discretization is not ergodic in general. While one can generate an ergodic Markov chain by solving the SDE in (1) with a stochastic backward Euler discretization [42], consistent learning from samples of such an ergodic chain will induce a more complicated loss function that incorporates the backward Euler scheme. While this case can be incorporated numerically, we neglect it in this paper since generally speaking the discretization scheme is unknown and the inconsistency of the numerical schemes that are used in generating the time series and in the construction of loss function in the learning algorithm induces an additional bias. For simplicity, we consider discrete Markov chain $\boldsymbol{x}^n$ generated by EM scheme,

$$\boldsymbol{x}^{n+1} - \boldsymbol{x}^n = \boldsymbol{a}(\boldsymbol{x}^n)\delta t + \boldsymbol{b}(\boldsymbol{x}^n)\sqrt{\delta t}\boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I}_m), \tag{3}$$

where $\delta t$ denotes the time step size and $\boldsymbol{I}_m$ is an $m \times m$ identity matrix. In the next section, we will use the same discretization to construct the appropriate loss functions to approximate $\boldsymbol{a}$ and $\boldsymbol{b}\boldsymbol{b}^\top$. Since the available training data are sampled from $\tilde{\pi}$, the learning algorithm can only (at best) achieve a population risk defined with respect to $\tilde{\pi}$ and we will characterize the error induced by the EM discretization using an existing perturbation theory result.

While the SDE is defined on an entire unbounded domain $\mathbb{R}^d$ (the measure is not compactly supported or the density is strictly positive away from zero), numerically we can only solve the PDE on a bounded domain. Following existing approaches of solving Fokker-Planck PDEs with neural-networks [66, 64, 70], we consider a simply connected compact domain $\Omega \subset \mathbb{R}^d$ large enough such that the density on $\mathbb{R}^d \setminus \Omega$ is effectively negligible. Practically, this assumption implies that the training data $\boldsymbol{x}^n \in \Omega$, and the stationary solution that we are looking for can be normalized with respect to $\Omega$, that is, $\int_\Omega p(\boldsymbol{x})\,d\boldsymbol{x} = 1$. This assumption is critical especially when the vector field $\boldsymbol{a}$ is unknown and needs to be numerically estimated with deep learning, for which one can only (at best) guarantee the error in $L^2$- topology over a compact domain. In Section 4, we will clarify this assumption.

## 3 Deep learning method for density estimation

In this section, we introduce a deep learning method to estimate the stationary density of SDE (1) from a time series of its solution, which consists of two steps. We begin the discussion by reviewing two deep learning architectures that we will use in our numerical simulations, the fully connected neural network (FNN) and the residual neural network (ResNet) in Section 3.1. Given a time series of the SDEs in (1), we fit the drift $\boldsymbol{a}$ and diffusion coefficients $\boldsymbol{b}\boldsymbol{b}^\top$ in the SDE (1) by NNs, denoted as $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$, respectively (see Section 3.2). Define $\hat{\mathscr{L}}^*$ as the Fokker-Planck (FP) differential operator generated from the estimated networks $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$ approximating the underlying (FP) operator $\mathscr{L}^*$ in (2). Our approach in estimating the stationary density $p$ is to solve the homogeneous PDE $\hat{\mathscr{L}}^* \hat{p} = 0$, where $\hat{p}$ is a solution parameterized by an FNN. The PDE can be solved via the network-based least square method introduced in Section 3.3.

### 3.1 Neural networks

We now give a brief overview of the two basic neural networks that have been widely employed in deep learning. The first one is the fully connected neural network (FNN). Suppose $d$ is the dimensions of inputs. Given an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, $L \in \mathbb{N}^+$, and $w_\ell \in \mathbb{N}^+$ for $\ell = 1, \ldots, L$, an FNN is constructed as the composition of $L$ simple nonlinear functions as follows

$$\phi_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}) := \boldsymbol{c}^\top \boldsymbol{h}_L \circ \boldsymbol{h}_{L-1} \circ \cdots \circ \boldsymbol{h}_1(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \mathbb{R}^d,$$

where $\boldsymbol{c} \in \mathbb{R}^{w_L \times 1}$; $\boldsymbol{h}_\ell(\boldsymbol{x}_\ell) := \sigma\left(\boldsymbol{W}_\ell \boldsymbol{x}_\ell + \boldsymbol{g}_\ell\right)$ with $\boldsymbol{W}_\ell \in \mathbb{R}^{w_\ell \times w_{\ell-1}}$ and $\boldsymbol{g}_\ell \in \mathbb{R}^{w_\ell}$ for $\ell = 1, \ldots, L$ ($W_0 := d$). With the abuse of notations, $\sigma(\boldsymbol{x})$ means that $\sigma$ is applied entry-wise to a vector $\boldsymbol{x}$ to obtain another vector of the same size. $w_\ell$ is the width of the $\ell$-th layer and $L$ is the depth of the FNN. $\boldsymbol{\theta} := \{\boldsymbol{c}, \boldsymbol{W}_\ell, \boldsymbol{g}_\ell : 1 \le \ell \le L\}$ is the set of all parameters in $\phi_{\mathrm{NN}}$ to determine the underlying neural network.

Besides FNN, in our numerical simulations, we will also consider the residual neural network (ResNet) [19]. Using similar notations above, ResNet can be defined recursively as follows,

$$
\begin{aligned}
&\boldsymbol{h}_0 = x, \boldsymbol{h}_{-1} = \boldsymbol{0}, \\
&\mathbf{v}_\ell = \sigma\left(\boldsymbol{W}_\ell \boldsymbol{h}_{\ell-1} + \boldsymbol{g}_\ell\right), \quad \ell = 1, 2, \cdots, L, \\
&\boldsymbol{h}_\ell = \mathrm{pad}(\boldsymbol{h}_{\ell-2}) + \mathbf{v}_\ell, \quad \ell = 1, 2, \cdots, L, \\
&\phi_{\mathrm{NN}}(x; \boldsymbol{\theta}) = \boldsymbol{c}^\top \boldsymbol{h}_L.
\end{aligned}
\tag{4}
$$

Here, the function $\mathrm{pad}(\cdot)$ is used to pad zeros to the vector such that two vectors in the summation (4) are of same size. Popular types of activation functions include the rectified linear unit (ReLU) $\sigma(x) = \max\{0, x\}$, ReLU$^3$ $\sigma(x) = \max\{0, x^3/6\}$, Tanh $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and Mish $\sigma(x) = x\mathrm{Tanh}(\log(1 + e^x))$ [45]. We use $\mathscr{F}_{L,W,\sigma}$ to denote the class of FNNs with depth $L$, width $W$ for all layers and activation $\sigma$.

### 3.2 Regression of drift and diffusion coefficients

Taking the expectation of (3) with respect to $\boldsymbol{\xi}_n$, one can see that

$$\mathbb{E}[\boldsymbol{x}^{n+1} - \boldsymbol{x}^n - \boldsymbol{a}(\boldsymbol{x}^n)\delta t] = 0. \tag{5}$$

With this identity, we consider a supervised learning method for estimating $\boldsymbol{a}(\boldsymbol{x})$ with neural networks. More precisely, we approximate every component of $\boldsymbol{a}(\boldsymbol{x})$ by an FNN $a_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ parameterized by a set of trainable parameters $\boldsymbol{\theta}$. In practice, letting $\boldsymbol{y}^n := \frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{\delta t}$, by (5), we define $\boldsymbol{\theta}_i^{\mathrm{a}}$ as follows,

$$\boldsymbol{\theta}_i^{\mathrm{a}} := \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=0}^{N-1} \left| y_i^n - a_{\mathrm{NN}}(\boldsymbol{x}^n; \boldsymbol{\theta}) \right|^2, \tag{6}$$

for $i = 1, \cdots, d$, where $y_i^n$ is the $i$-th component of $\boldsymbol{y}^n$. Then we define the vector-valued function

$$\boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^{\mathrm{a}}) := \left[ a_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_1^{\mathrm{a}}), \cdots, a_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_d^{\mathrm{a}}) \right]^\top \tag{7}$$

as the drift estimator to approximate $\boldsymbol{a}(\boldsymbol{x})$, where $\boldsymbol{\theta}^{\mathrm{a}}$ consists of $\{\boldsymbol{\theta}_i^{\mathrm{a}}\}$.

This is a supervised learning task to estimate $\boldsymbol{a} : \mathbb{R}^d \to \mathbb{R}^d$ from a pair of labelled training data set, $\{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n=0}^{N-1}$. To simplify the analysis in the next section, we assume that $\boldsymbol{x}^i$ are i.i.d. samples of the stationary random distribution $\tilde{\pi}$. While we do not employ this simplification in our numerical study, practically, such i.i.d. samples can be obtained by sub-sampling from the Markov chain $\{\boldsymbol{x}^n\}_{n\ge0}$ such that their temporal correlation is negligible. For convenience of the following discussion, we denote $\mathscr{X} := \{\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{N-1}\}$ and $\mathscr{Y} := \{\boldsymbol{y}^0, \ldots, \boldsymbol{y}^{N-1}\}$. In (6), the parameter $\boldsymbol{\theta}_i^{\mathrm{a}}$ is a global minimizer of the empirical loss function. Practically, since tochastic gradient descent or the Adam method [31] is used, such a global minimizer may not necessarily be identified.

Next, we approximate $\boldsymbol{b}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})^\top$ in similar ways. The $(i, j)$-th component of $\boldsymbol{b}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})^\top$ can be approximated by an FNN $B_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}_{ij}^{\mathrm{b}})$. Since $\boldsymbol{\xi}_n$ is independent of $\boldsymbol{x}^n$, using the fact $\mathbb{E}[\boldsymbol{\xi}_n \boldsymbol{\xi}_n^\top] = \boldsymbol{I}_n$ and (3) we have

$$\mathbb{E}\left[ (\boldsymbol{x}^{n+1} - \boldsymbol{x}^n - \boldsymbol{a}(\boldsymbol{x}^n)\delta t)(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n - \boldsymbol{a}(\boldsymbol{x}^n)\delta t)^\top - \boldsymbol{b}(\boldsymbol{x}^n)\boldsymbol{b}(\boldsymbol{x}^n)^\top \delta t \right] = 0.$$

Based on this identity, assuming that we have obtained the network $\boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^{\mathrm{a}}) \approx \boldsymbol{a}(\boldsymbol{x})$, we can compute $\boldsymbol{\theta}_{ij}^{\mathrm{b}}$ by

$$\boldsymbol{\theta}_{ij}^{\mathrm{b}} := \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=0}^{N-1} \left| (\boldsymbol{y}_i^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n, \boldsymbol{\theta}_i^{\mathrm{a}}))(\boldsymbol{y}_j^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n, \boldsymbol{\theta}_j^{\mathrm{a}})))^\top - \frac{1}{\delta t} B_{\mathrm{NN}}(\boldsymbol{x}^n; \boldsymbol{\theta}) \right|^2. \tag{8}$$

for $1 \le i, j \le d$. Similarly, the global minimizer $\boldsymbol{\theta}_{ij}^{\mathrm{b}}$ may not be identified in practice. To summarize, If these global minimizers are identified, the training procedure gives $\boldsymbol{B}_{\mathrm{NN}}(\boldsymbol{x}) := \left[ B_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{\theta}_{ij}^{\mathrm{b}}) \right]_{i=1,\cdots,d}^{j=1,\cdots,d} \approx \boldsymbol{b}(\boldsymbol{x}) \boldsymbol{b}(\boldsymbol{x})^{\top}$.

We should also point out that when the diffusion tensor is a constant matrix, $\boldsymbol{b} \in \mathbb{R}^{d \times m}$, we do not need to solve the optimization problem (8) by deep learning. In such a case, $\boldsymbol{B}_{\mathrm{NN}}$ is specified as a matrix and we will empirically estimate $\boldsymbol{b}\boldsymbol{b}^{\top}$ using the residual from the drift estimator $\boldsymbol{a}_{\mathrm{NN}}(\cdot)$. Particularly,

$$\boldsymbol{B}_{\mathrm{NN}} := \frac{\delta t}{N} \sum_{n=1}^{N} \left( \boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n; \boldsymbol{\theta}^{\mathrm{a}}) \right) \left( \boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n; \boldsymbol{\theta}^{\mathrm{a}}) \right)^{\top}, \tag{9}$$

where we used the same notation $\boldsymbol{B}_{\mathrm{NN}}$ and understand that it is a $d \times d$ matrix in this case.

### 3.3 Estimation of the stationary density

Given the approximate drift $\boldsymbol{a}_{\mathrm{NN}} \approx \boldsymbol{a}$ and diffusion coefficients, $\boldsymbol{B}_{\mathrm{NN}} \approx \boldsymbol{b}\boldsymbol{b}^{\top}$, we define the estimated FP operator,

$$\hat{\mathcal{L}}^* p := -\mathrm{div}(\boldsymbol{a}_{\mathrm{NN}} p) + \frac{1}{2} \sum_{i,j=1}^{d} \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} (B_{\mathrm{NN}}^{ij} p), \tag{10}$$

where $B_{\mathrm{NN}}^{ij}$ is the $(i, j)$-entry of $B_{\mathrm{NN}}$.

Subsequently, the stationary density is estimated by solving the approximate stationary FP equation,

$$\hat{\mathcal{L}}^* \hat{p} = 0, \quad \text{in } \Omega \tag{11}$$

where $\hat{p} : \Omega \to (0, \infty)$ denotes the analytical solution of this PDE that satisfies,

$$\int_{\Omega} \hat{p}(\boldsymbol{x}) d\boldsymbol{x} = 1. \tag{12}$$

Numerically, we set $\Omega$ to be a rectangular domain that is large enough yet tightly covers most of the data points in $\mathcal{X}$.

We solve the equation (11) with the condition (12) by the popular network-based least square method [10, 33]. Specifically, We use a neural network $\hat{p}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$ with a parameter set $\boldsymbol{\theta}$ determined by solving the following minimization problem,

$$\min_{\boldsymbol{\theta}} J[\hat{p}_{\mathrm{NN}}(\cdot; \boldsymbol{\theta})],$$

where

$$J[q] := \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)}^2 + \lambda_1 \left| \int_{\Omega} q(\boldsymbol{x}) d\boldsymbol{x} - 1 \right|^2 + \lambda_2 \|q\|_{L^2(\partial\Omega)}^2, \quad \forall q : \Omega \to \mathbb{R}, \tag{13}$$

where $\lambda_1$ is a regularization constant corresponding to the normalization factor in (12) such to ensure nontrivial solution; $\lambda_2$ is a regularization parameter corresponding to an artificial Dirichlet boundary condition. In our numerical simulation, we empirically found that the artificial boundary constraint can be neglected if the function values at the prescribed boundary is sufficiently small.

In the practical computation, when $d$ is moderately large, the first term of (13) is usually computed via a Monte-Carlo integration. For example, if the data $\{\boldsymbol{x}_{\mathrm{I}}^n\}_{n=1}^{N_1}$ are uniformly distributed points in $\Omega$, then

$$\|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)}^2 \approx \frac{|\Omega|}{N_1} \sum_{n=1}^{N_1} \left| \hat{\mathcal{L}}^* q(\boldsymbol{x}_{\mathrm{I}}^n) \right|^2, \tag{14}$$

where $|\Omega|$ denotes the volume of the domain $\Omega$.

Similarly, as for the second term in (13), Monte-Carlo integral is formulated as

$$\int_{\Omega} q(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} q(\boldsymbol{x}_{\mathrm{II}}^n), \tag{15}$$

where $\{\boldsymbol{x}_{\mathrm{II}}^n\}_{n=1}^{N_2}$ are uniformly distributed sampled points in $\Omega$.

5

For the third term in (13), we approximate,

$$\|q\|_{L^2(\partial\Omega)} \approx \frac{|\partial\Omega|}{N_3} \sum_{n=1}^{N_3} |q(\boldsymbol{x}_{\mathrm{III}}^n)|^2, \tag{16}$$

where $\{\boldsymbol{x}_{\mathrm{III}}^n\}_{n=1}^{N_3}$ are uniformly distributed sampled points in $\partial\Omega$.

Combining (14), (15), and (16), the training procedure is to minimize the following empirical loss function,

$$J_S[q] := \frac{|\Omega|}{N_1} \sum_{n=1}^{N_1} \left|\mathscr{L}^* q(\boldsymbol{x}_{\mathrm{I}}^n)\right|^2 + \lambda_1 \left|\frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} q(\boldsymbol{x}_{\mathrm{II}}^n) - 1\right|^2 + \lambda_2 \frac{|\partial\Omega|}{N_3} \sum_{n=1}^{N_3} \left|q(\boldsymbol{x}_{\mathrm{III}}^n)\right|^2. \tag{17}$$

Let

$$\boldsymbol{\theta}^S = \arg\min_{\boldsymbol{\theta}} J_S[\hat{p}_{\mathrm{NN}}(\cdot, \boldsymbol{\theta})], \tag{18}$$

then the density estimator is given by $\hat{p}_{\mathrm{NN}}(\cdot; \boldsymbol{\theta}^S) \approx p(\cdot)$ with $\hat{p}_{\mathrm{NN}} : \Omega \to \mathbb{R}$ and $\int_\Omega \hat{p}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^S)\mathrm{d}\boldsymbol{x} \approx 1$.

We should point out that in our numerical simulations, since the time series $\{\boldsymbol{x}^n\}_{n=1}^N$ that are distributed in accordance to $\tilde{\pi}$ are available, we conveniently replace the first component in the loss function in (13) with a weighted norm, $L^2(\Omega, \tilde{\pi})$ and accordingly adjust the Monte-Carlo sum in the first component in the empirical loss function in (17). While the convergence analysis corresponding to a weighted norm is equivalent to that of the unweighted norm when $\tilde{pi}$ is absolutely continuous with respect to Lebesque measure with bounded density function, for simplicity of the exposition, we will consider the analysis corresponding to loss functions in (13) with unweighted $L^2(\Omega)$ norms. If the dimension $d$ is lower, one can also adopt numerical quadrature rules such as Gauss-type quadrature to evaluate the integrals in (13) for higher accuracy.

## 4 Convergence Theory

In this section, we deduce an error bound for the estimator $\hat{p}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^S)$, where $\boldsymbol{\theta}^S$ is the global minimizer of the empirical loss function in (17). Throughout the discussion in this section, we restrict the diffusion coefficient $\boldsymbol{b} \in \mathbb{R}^{d \times m}$ to be a full column rank matrix. We use the notation $\|\cdot\|$ for the Euclidean norm in $\mathbb{R}^d$.

### 4.1 Preliminary remarks

Let us set the stage for our discussion by specifying the class of FNNs. In Section 3.1, we introduced the general class of FNNs $\mathscr{F}_{L,M,\sigma}$. While for the simplicity of analysis, we choose special classes of FNNs as the hypothesis spaces of the optimization.

On one hand, we consider using deep ReLU FNNs with uniform bounds in the minimization (6), the regression of true drift $\boldsymbol{a}(\boldsymbol{x})$. Specifically, for any $P > 0$, we denote

$$\mathscr{F}_{L,M,\mathrm{ReLU}}^P = \left\{\phi \in \mathscr{F}_{L,M,\mathrm{ReLU}} : |\phi(\boldsymbol{x})| \le P, \ \forall \boldsymbol{x} \in \Omega\right\}, \tag{19}$$

as the class of ReLU FNNs with depth $L$, width $M$, and a uniform bound $P$ in $\Omega$.

On the other hand, we consider using two-layer ReLU³ FNNs with parameter bounds in the minimization (18), the approximation of the true density $p(\boldsymbol{x})$. More precisely, for any $Q > 0$, we explicitly specify

$$\mathscr{F}_{2,M,\dot{\sigma},Q} = \left\{\phi : \Omega \to \mathbb{R} : \phi(\boldsymbol{x}) = \frac{1}{M} \sum_{m=1}^M c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}), \ |c_m|, \|\boldsymbol{w}_m\|_1 \le Q\right\}, \tag{20}$$

where $\dot{\sigma} = \max(0, x^3/6)$ denoting the ReLU³ activation function widely used in network-based methods for second-order PDEs. For simplicity, we omit the biases $\boldsymbol{g}_\ell$ in the definition of FNNs in Section 3.1.

Since the analysis depends on the results of the perturbation theory on the ergodic Itô diffusion in [72], we will briefly review the concepts of geometric ergodicity and other relevant results.

We will now make precise the assumptions mentioned in Section 2.

**Assumption 4.1.** *The following are key assumptions of the underlying system that generates the process $X_t$:*

i. **Lipschitz & Linear growth bound:** *The vector field $\boldsymbol{a} : \mathbb{R}^d \to \mathbb{R}^d$ is globally Lipschitz with Lipschitz constant $\lambda_{\boldsymbol{a}} > 0$ to ensure the existence and uniqueness of the solution of the SDE in* (1) *given an initial condition. There exists a constant $K \in (0, +\infty)$ such that*

$$\|\boldsymbol{a}(\boldsymbol{x})\|^2 \le K^2(1 + \|\boldsymbol{x}\|^2), \forall \boldsymbol{x} \in \mathbb{R}^d.$$

*This linear growth assumption will ensure that the even order moments can be bounded under the same rate.*

ii. **Geometric ergodicity:** *The Markov process $X_t$ is geometrically ergodic with a unique invariant measure $\pi$. See e.g. Assumptions 2.2-2.3 in [72] for the detailed conditions to achieve the geometric ergodicity for the SDE driven by additive Brownian noises. One of the conditions that is important for our discussion is that there exists a Lyapunov function $V : \mathbb{R}^d \to [1, \infty)$ with $\lim_{x \to \infty} V(\boldsymbol{x}) = +\infty$, and $c_1, c_2 \in (0, +\infty)$ such that*

$$\mathscr{L} V(\boldsymbol{x}) \le -c_1 V(\boldsymbol{x}) + c_2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d,$$

*where $\mathscr{L}$ is the $L^2(\mathbb{R}^d)$ adjoint of the FP operator $\mathscr{L}^*$ defined in* (2).

iii. **Essentially quadratic:** *The Lyapunov function $V = W^\ell$ for some $\ell \ge 1$, where $W$ is essentially quadratic, i.e., there exist constants $C_i \in (0, +\infty)$, $i = 1, 2, 3$, such that*

$$C_1 \left(1 + \|\boldsymbol{x}\|^2\right) \le W(\boldsymbol{x}) \le C_2 \left(1 + \|\boldsymbol{x}\|^2\right), \quad \|\nabla W(\boldsymbol{x})\| \le C_3 \left(1 + \|\boldsymbol{x}\|\right), \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

*Together with the previous two assumptions, there exists $\delta_0 > 0$ such that for all $\delta t \in (0, \delta_0)$, the discrete Markov chain induced by the EM algorithm in* (3) *is geometrically ergodic with the invariant measure, $\tilde{\pi}$, and that,*

$$\sup_{f \in \mathscr{G}_\ell} \left|\pi(f) - \tilde{\pi}(f)\right| \le K_1 (\delta t)^\nu \pi(V),$$

*for some $K_1 = K_1(\ell)$ and $\nu \in (0, 1/2)$. Here, the supremum is defined over a set of locally Lipschitz functions bounded above by $V$,*

$$\mathscr{G}_\ell := \left\{ f(\boldsymbol{x}) \le V(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^d \text{ and } \big| |f(\boldsymbol{x}) - f(\boldsymbol{y})| \le C_\ell \left(1 + \|\boldsymbol{x}\|^{2\ell-1} + \|\boldsymbol{y}\|^{2\ell-1}\right) \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d \right\}. \quad (21)$$

**Lemma 4.1.** *Under the assumptions 4.1, for any small $0 < \epsilon \ll 1$, suppose that the estimator $\hat{\boldsymbol{a}} : \mathbb{R}^d \to \mathbb{R}^d$ is globally Lipschitz with Lipschitz constant independent of $\epsilon$ and is a consistent estimator in the following sense,*

$$\|\boldsymbol{a}(\boldsymbol{x}) - \hat{\boldsymbol{a}}(\boldsymbol{x})\|^2 \le K_2(1 + \|\boldsymbol{x}\|^2)\epsilon^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d, \quad (22)$$

*for some constant $K_2 > 0$ that is independent of $\epsilon$. Let us denote $\hat{X}_n := \hat{X}(t_n)$, where $t_n = n\delta t$ to be a Markov chain generated by the solution to,*

$$d\hat{X} = \hat{\boldsymbol{a}}(\hat{X}_t) \, dt + \hat{\boldsymbol{b}} \, dW_t, \quad \hat{X}_0 = \boldsymbol{x}, \quad (23)$$

*with $\hat{\boldsymbol{b}}\hat{\boldsymbol{b}}^\top := \hat{\boldsymbol{B}}$. For any $\boldsymbol{x} \in \mathbb{R}^d$, there exists $0 < \rho < 1$ and $K_1 > 0$ such that,*

$$\sup_{f \in \mathscr{G}_\ell} |\pi(f) - \mathbb{E}^{\boldsymbol{x}}[f(\hat{X}_n)]| \le K_3 \left[\left(\rho^n + \frac{1 - \rho^n}{1 - \rho} \epsilon\right) V(\boldsymbol{x})\right], \quad \forall n \ge 0, \quad (24)$$

*where the set $\mathscr{G}_\ell$ is defined in* (21). *If the process $\hat{X}$ associated to* (23) *has an invariant measure $\hat{\pi}$, then there exist $0 < \alpha < 1$, $0 < \beta < \infty$, and $0 < \gamma < 1 - \alpha$ such that, $\hat{\pi}(V) \le \frac{\beta}{1 - \alpha - \gamma}$.*

The result above holds for all $\boldsymbol{x} \in \mathbb{R}^d$ by requiring the condition in (22) and that underlying process $X(t)$ is ergodic in $\mathbb{R}^d$ with a unique invariant measure $\pi$. Similar conclusion was reported in [22] under a much stronger uniform convergence in placed of (22). One of the key issue in applying this result directly to the learning configuration is that the assumption in (22) can be difficult to achieve unless if one consider learning with a loss function defined with the topology that is used to deduced the error bound in (24), which relies on the perturbation theory of Markov chain. The usual practical machine learning computations solve a supervised learning problem induced by a weaker topology (commonly $L^2$) on a bounded domain. In such a weaker topology (relative to the sup norm in (24)), one can at best expect to construct an estimator with convergence guaranteed under an $L^2(\Omega, \tilde{\pi})$ error on a compact domain $\Omega \supset \mathscr{X}$ that contains all the training data. In the numerical section, we will empirically show that the pointwise accuracy of $\boldsymbol{a}$ and verify the accuracy of the invariant mean and covariance statistics induced by a Markov chain generated by the estimated drift and diffusion coefficients.

To overcome the incompatibility of the domains, we consider the following assumption.

**Assumption 4.2.** *Let $\Omega \subset \mathbb{R}^d$ be a simply connected compact domain such that $P(X \notin \Omega) \le \epsilon_0$ for some $0 < \epsilon_0 \ll 1$. For example, let $\Omega := B(0, R) = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \le R\}$ be a closed Euclidean ball of radius $R > 1$ and suppose that $X$ has mean zero (centered) and is a sub-exponentially distributed random variable, $SE(v^2, \alpha)$, with $v, \alpha > 0$, then by concentration inequality for sub-exponential distribution, one obtains*

$$\mathbb{P}(\|X\| \ge R) \le 2e^{-\frac{R}{2\alpha}} := \epsilon_0, \quad \forall R > v^2 \alpha^{-1}. \tag{25}$$

*Let $\tilde{X}$ be a random variable corresponding to the stationary distribution induced by the Euler-Maruyama discretization in (3), using the Markov inequality and strong error bound of EM scheme, one can deduce that $\mathbb{P}[\|X - \tilde{X}\| \le (\delta t)^{1/4}] \le (\delta t)^{-1/4}\mathbb{E}[|X - \tilde{X}] \le C(\delta t)^{1/4}$, which means that $\mathbb{P}[\|\tilde{X}\| \ge R + (\delta t)^{1/4}] \le \mathbb{P}[\|X\| \ge R]\mathbb{P}[\|X - \tilde{X}\| \ge (\delta t)^{1/4}] \le O(\epsilon_0)$. Even if $X$ (resp. $\tilde{X}$) is defined on $\mathbb{R}^d$, one can almost surely realize $\|X\| \le R$ (resp. $\|\tilde{X}\| \le R + \delta t^{1/4}$) for large enough $R > 0$. This assumption effectively means that the process $X$ satisfies the Assumption 4.1 for $\boldsymbol{x} \in \Omega = B(0, R)$ almost surely for large enough $R$. This also implies that Lemma 4.1 is valid for $\boldsymbol{x} \in \Omega$, where we understood $\pi(f) := \int_\Omega f(\boldsymbol{x})\pi(d\boldsymbol{x})$ in (24). In the convergence theory below, without loss of generality, we will assume that $\Omega = [0, 1]^d$. For general $\Omega$, similar results can be derived easily by rescaling $\Omega$ to $[0, 1]^d$ with an isomorphic map.*

With the above assumption, we only need to restrict our attention to a compact domain $\Omega$ and, hence, the assumption that $\hat{\boldsymbol{a}}$ is globally Lipschitz with Lipschitz constant independent of $\epsilon$ is reasonable. In our algorithm, we use the ReLU activation functions to construct $\hat{\boldsymbol{a}}$ and, hence, $\hat{\boldsymbol{a}}$ is a globally Lipschitz continuous function. By the simultaneous approximation of ReLU neural networks in [17, 21], as long as $\boldsymbol{a} \in C^s$ with $s > 1$, there exists a ReLU network $\hat{\boldsymbol{a}}$ approximating $\boldsymbol{a}$ in the Sobolev norm of $W^{1,\infty}(\Omega)$ with $\Omega$ as a compact set. This means that the Lipschitz constant of $\hat{\boldsymbol{a}}$ can be bounded by a constant depending on $\boldsymbol{a}$ instead of the approximation accuracy. However, how to identify $\hat{\boldsymbol{a}}$ satisfying these assumptions is a problem of the optimization algorithm.

We use the notations $\tilde{\pi}(f) = \int_\Omega f(\boldsymbol{x})d\tilde{\pi}(\boldsymbol{x})$ and $\hat{\pi}(f) = \int_\Omega f(\boldsymbol{x})d\hat{\pi}(\boldsymbol{x})$ for integrals over $\Omega$. With Assumption 4.2, we now let the solution $\hat{p} : \Omega \to (0, \infty)$ of the approximate FP equation be the density of $\hat{\pi}$, defined with respect to the Lebesque measure, $d\hat{\pi} = \hat{p}(x)dx$. Since the PDE in (11) is defined with the estimated coefficients, namely, $\boldsymbol{a}_{NN} : \Omega \to \Omega$ as defined in (7) and $\boldsymbol{B}_{NN} \in \mathbb{R}^{d \times d}$ as defined in (9), the error analysis below will need to account for the errors induced by these estimations. Recall that $\boldsymbol{b}$ is a constant matrix and $\boldsymbol{a}_{NN}$ is the best empirical estimator from the chosen hypothesis space (e.g., a class of FNN-functions of the chosen architecture), obtained by regressing the labeled training data $\{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1}^N$, where $\boldsymbol{x}^i \in \mathcal{X}$ and $\boldsymbol{y}^i := \boldsymbol{a}(\boldsymbol{x}^i) + \boldsymbol{\eta}^i, \boldsymbol{\eta}^i \sim \mathcal{N}(\boldsymbol{0}, (\delta t)^{-1}\boldsymbol{bb}^\top)$.

To quantify the error of the diffusion estimator, one can subtract $\boldsymbol{bb}^\top$ from the empirical estimator defined in (9) and derive the following upper bound,

$$\|\boldsymbol{bb}^\top - \boldsymbol{B}_{NN}\|_2 \le \left\|\sum_{i=1}^N D_i\right\|_2 + \delta t \mathbb{E}_{\tilde{\pi}}\left[\left\|(\boldsymbol{a}(X) - \boldsymbol{a}_{NN}(X; \boldsymbol{\theta}^a))\right\|^2\right], \tag{26}$$

where for each $i = 1, \dots, N$,

$$D_i := \frac{\delta t}{N}\left(\boldsymbol{y}^i - \boldsymbol{a}_{NN}(\boldsymbol{x}^i; \boldsymbol{\theta}^a)\right)\left(\boldsymbol{y}^i - \boldsymbol{a}_{NN}(\boldsymbol{x}^i; \boldsymbol{\theta}^a)\right)^\top - \frac{1}{N}\left(\delta t \mathbb{E}_{\tilde{\pi}}[(\boldsymbol{a}(X) - \boldsymbol{a}_{NN}(X; \boldsymbol{\theta}^a))(\boldsymbol{a}(X) - \boldsymbol{a}_{NN}(X; \boldsymbol{\theta}^a))^\top] + \boldsymbol{bb}^\top\right), \tag{27}$$

is an independent, random, symmetric matrix of mean zero. Since $\boldsymbol{x}^i$ is bounded almost surely, one can bound $D_i$ almost surely with large enough $R > 0$. In such a case, one can use a matrix concentration inequality to bound the first term in (27) with large enough training sample $N$. Particularly, using the Matrix Bernstein inequality (e.g., Theorem 1.6.2 in [62]), if we define

$$\epsilon := \delta t \mathbb{E}_{\tilde{\pi}}\left[\left\|(\boldsymbol{a}(X) - \boldsymbol{a}_{NN}(X; \boldsymbol{\theta}^a)\right\|^2\right], \tag{28}$$

and denote $\|D_i\| \le \frac{D}{N}$ for some $D > 0$, then the first term in (26) is smaller than $\epsilon$ with probability $1 - 2d\exp(-\frac{\epsilon^2/2}{O(N^{-2}) + DN^{-1}\epsilon/3}) > 0$. This means, one can bound $\|\sum_{i=1}^N D_i\| \le \epsilon$ with high probability by choosing $N \ge C\epsilon^{-1}\log 2d$. We can therefore conclude that the spectral error in (26) is of order-$\epsilon$, which is the generalization error rate as defined in (28).

We should point out that the result in Lemma 4.1 does not assume the ergodicity of the Markov process $\hat{X}(t)$ generated by the SDE in (23). Suppose that $\hat{X}(t)$ is generated with $\hat{\boldsymbol{a}} = \boldsymbol{a}_{NN}$ and $\hat{\boldsymbol{b}}\hat{\boldsymbol{b}}^\top = \boldsymbol{B}_{NN}$ has an invariant measure $\hat{\pi}$ on $\Omega$. Integrating (24) with respect to $\hat{\pi}$, we obtain,

$$\left|\pi(f) - \hat{\pi}(f)\right| = \left|\pi(f) - \int_\Omega f(\boldsymbol{x})\hat{\pi}(d\boldsymbol{x})\right| = \left|\pi(f) - \int_\Omega \mathbb{E}^{\boldsymbol{x}}[f(\hat{X}_n)]\hat{\pi}(d\boldsymbol{x})\right| \le K_3\hat{\pi}(V)\epsilon \tag{29}$$

as $n \to \infty$. To obtain (29), we have used (24). With this background, the error bound for $\hat{p}_{NN}(x; \theta^S)$ can be deduced by accounting the regression error of $a$ and the error from the proposed PDE solver,

$$
\begin{aligned}
\left| \pi(f) - \int_\Omega f(x) \hat{p}_{NN}(x; \theta^S) \, dx \right| &\leq \left| \pi(f) - \int_\Omega f(x) \hat{p}(x) dx \right| + \left| \int_\Omega f(x) \big( \hat{p}(x) - \hat{p}_{NN}(x; \theta^S) \big) \, dx \right| \\
&\leq K_3 \hat{\pi}(V) \delta t \underbrace{\mathbb{E}_{\tilde{\pi}} \left[ \left\| (a(X) - a_{NN}(X; \theta^a)) \right\|^2 \right]}_{(I)} + \|f\|_{L^2(\Omega)} \underbrace{\| \hat{p} - \hat{p}_{NN}(\cdot; \theta^S) \|_{L^2(\Omega)}}_{(II)}, \quad (30)
\end{aligned}
$$

where we have used (29), (28), and the Cauchy-Schwartz inequality. In the next two subsections, we will bound the terms (I) and (II) in (30).

## 4.2 Regression error for the drift estimator

Now let us consider the error in the regression of the drift coefficients, namely, the minimization problem (6). We will derive the $L^2$ error with respect to $\tilde{\pi}$ between the estimator $a_{NN}(x; \theta^a)$ and the true drift function $a(x)$. For this purpose, given a class $\mathscr{F}$ of functions: $\Omega \to \mathbb{R}$, we denote its pseudo dimension by $\mathrm{Pdim}(\mathscr{F})$, which is the largest integer $m$ for which there is some $(x_1, \cdots, x_m, y_1, \cdots, y_m) \in \Omega^m \times \mathbb{R}^m$ such that for any $(b_1, \cdots, b_m) \in \{0, 1\}^m$, there exists $f \in \mathscr{F}$ satisfying $f(x_i) > y_i \Leftrightarrow b_i = 1 \ \forall i$. The prediction error analysis of FNNs have been studied in several papers, e.g., [55, 49, 5, 41, 28, 38, 44, 12]. In particular, we introduce the following lemma concerning the prediction error of the FNN-based least square regression, which is studied in [28].

**Lemma 4.2** ([28], Theorem 4.2). *Let $f_0 : [0, 1]^d \to \mathbb{R}$ be a Hölder continuous function, i.e., there exist $\lambda \geq 0$ and $\alpha \in (0, 1]$ such that $|f_0(x) - f_0(y)| \leq \lambda \|x - y\|^\alpha$ for all $x, y \in [0, 1]^d$. Suppose $\|f_0\|_{L^\infty([0,1]^d)} \leq P$ for some $P \geq 1$. Let $v$ be a probability measure that is absolutely continuous with respect to the Lebesgue measure and a random variable $x \sim v$. Let $\eta$ be a random variable satisfying $\mathbb{E}[\eta] = 0$ and $\mathrm{Var}[\eta] = \sigma^2$. Let $\{x^n\}_{n=1}^N$ be $N$ independent and identically distributed samples of $x$, and $y^n = f_0(x^n) + \eta$ is the response with noise $\eta$ for each $n$. For any $I_1, I_2 \in \mathbb{N}^+$, let*

$$
\theta^{f_0} := \arg\min_\theta \frac{1}{N} \sum_{n=1}^N \left| y^n - f_{NN}(x^n; \theta) \right|^2,
$$

*where $f_{NN} \in \mathscr{F}_{L,W,ReLU}^P$ having depth $L = 12 I_2 + 14$ and width $W = \max\{4 d \lfloor I_1^{\frac{1}{d}} \rfloor + 3d, 12 I_1 + 8\}$ for all hidden layers. Then the prediction error is given by*

$$
\mathbb{E}_v \left[ |f_{NN}(\cdot, \theta^{f_0}) - f_0|^2 \right] \leq C [P^2 W L (d + WL) \log(Wd + W^2 L)(\log N)^3 N^{-1} + \lambda^2 d (I_1 I_2)^{-4\alpha/d}], \quad (31)
$$

*for $N \geq \mathrm{Pdim}(\mathscr{F}_{L,W,ReLU}^P)$, where $C$ is a constant that does not depend on $d, N, L, W, \lambda, \alpha, I_1, I_2, P$.*

In Lemma 4.2, the exponent of the error bound in (33) can be improved to be dimension-independent if we assume $f_0$ is in Barron-type spaces, which are first studied in [2] and further developed in [14, 13, 38, 60, 59, 6, 3]. Here we follow the Barron space with respect to two-layer ReLU networks proposed in [13]. Suppose $f : \Omega \to \mathbb{R}$ is a function having the following form,

$$
f(x) = \int_{\mathbb{R} \times \mathbb{R}^d} c \max(w^\top x, 0) \rho(dc, dw) = \mathbb{E}_\rho[c \max(w^\top x, 0)], \quad x \in \Omega
$$

for some probability measure $\rho$ on $\mathbb{R} \times \mathbb{R}^d$, then its Barron norm is defined by

$$
\|f\|_{\mathscr{B}_{ReLU}} = \inf_{\rho \in P_f} (\mathbb{E}_\rho |c| \|w\|_1), \quad (32)
$$

where $P_f := \{\rho : f(x) = \mathbb{E}_\rho[c \max(w^\top x, 0)]\}$. And the ReLU Barron space is defined by $\mathscr{B}_{ReLU} = \{f \in C^0 : \|f\|_{\mathscr{B}_{ReLU}} < \infty\}$. Now we have the following result.

**Lemma 4.3.** *Let $f_0 : [0, 1]^d \to \mathbb{R}$ such that $\|f_0\|_{\mathscr{B}_{ReLU}} \leq P$ and $\|f_0\|_{L^\infty([0,1]^d)} \leq P$ for some $P \geq 1$. For the least square regression proposed in Lemma 4.2, we let $f_{NN} \in \mathscr{F}_{2,W,ReLU}^P$ for some $W \in \mathbb{N}^+$, Then the prediction error is given by*

$$
\mathbb{E}_v \left[ |f_{NN}(\cdot, \theta^{f_0}) - f_0|^2 \right] \leq C \left[ P^2 W(d + W) \log(Wd + W^2)(\log N)^3 N^{-1} + \|f_0\|_{\mathscr{B}_{ReLU}}^2 d W^{-1} \right], \quad (33)
$$

*for $N \geq \mathrm{Pdim}(\mathscr{F}_{2,W,ReLU}^P)$, where $C$ is a constant that does not depend on $d, N, W, f_0, P$.*

*Proof.* See Appendix A. $\qquad \square$

In our case, we set in the hypothesis of Lemma 4.2 that $L = O(I_2)$ and $W = O(I_1)$ are both large integers. Combining with Lemma 4.3, the error estimation for the minimization problem (6) can be directly obtained.

**Lemma 4.4.** *In addition to the Assumption 4.1, we let $\tilde{\pi}$ be absolutely continuous with respect to the Lebesgue measure. Denote $P_{\boldsymbol{a}} = \max\{\|\boldsymbol{a}\|_{L^\infty(\Omega)}, 1\}$.*

1. *Let $L$ and $W$ be integers large enough, then the estimator $\boldsymbol{a}_{NN}$ defined in (7) with components $a_{NN} \in \mathscr{F}_{L,W,ReLU}^{P_{\boldsymbol{a}}}$ satisfies*

$$\mathbb{E}_{\tilde{\pi}}\left[|\boldsymbol{a}_{NN} - \boldsymbol{a}|^2\right] \le C_{\boldsymbol{a}}\left(d^2 WLN^{-1} + d(WL)^2 N^{-1} + d^2(WL)^{-4/d}\right), \tag{34}$$

*for $N \ge Pdim(\mathscr{F}_{L,W,ReLU}^{P_{\boldsymbol{a}}})$;*

2. *Suppose all components of $\boldsymbol{a}$ are in $\mathscr{B}_{ReLU}$ with Barron norms no greater than $P_{\boldsymbol{a}}$. Let $W \in \mathbb{N}^+$, then the estimator $\boldsymbol{a}_{NN}$ defined in (7) with components $a_{NN} \in \mathscr{F}_{2,W,ReLU}^{P_{\boldsymbol{a}}}$ satisfies*

$$\mathbb{E}_{\tilde{\pi}}\left[|\boldsymbol{a}_{NN} - \boldsymbol{a}|^2\right] \le C_{\boldsymbol{a}}\left(d^2 WN^{-1} + dW^2 N^{-1} + d^2 W^{-1}\right), \tag{35}$$

*for $N \ge Pdim(\mathscr{F}_{2,W,ReLU}^{P_{\boldsymbol{a}}})$,*

*where $C_{\boldsymbol{a}} > 0$ is a term that depends on $\boldsymbol{a}$ and at most a polynomial in the logarithm of $N, L, W$.*

In Lemma 4.4, the Barron assumption on the target function helps to overcome the curse of dimensionality. In the following analysis for the solution error in the approximate FP equation, we will specify a Barron space for ReLU[3] networks and assume that the true solution is in this space; therefore the derived solution error is also exponentially independent of dimensions.

Another situation to mitigate the curse of dimensionality is when the data points are supported on a neighborhood of a low-dimensional Riemannian submanifold in $\Omega$ [5, 28]. Since it does not apply to the current problem in practice, we will not discuss more on this situation.

### 4.3 Solution error for the approximate FP equation

Now let us consider the error between $\hat{p}_{NN}(\cdot; \boldsymbol{\theta}^S)$ and the true solution $\hat{p}$ of the approximate stationary FP equation (11). In this section, we only consider the case that $\{\boldsymbol{x}_{MC}^n\}_{n=1}^{N_1}$ in (15) are uniformly distributed in $\Omega$. Similar results apply to other measures with smooth densities supported on $\Omega$.

First, we rewrite the approximate stationary FP equation (15) in the following divergence form

$$-\hat{\mathscr{L}}^* \hat{p} = -\sum_{i,j=1}^d \left(\frac{1}{2} B_{NN}^{ij} \hat{p}_{x_j}\right)_{x_i} + \sum_{i=1}^d a_{NN}^i \hat{p}_{x_i} + \left(\sum_{i=1}^d \frac{\partial a_{NN}^i}{\partial x_i}\right) \hat{p} = 0, \quad \text{in } \Omega. \tag{36}$$

The error analysis is valid only when the equation (36) is well-posed. So we need to set up specific assumptions on the coefficients of (36). First, note that $\boldsymbol{B}_{NN}$ is positive semi-definite and (36) is elliptic, so we assume further that (36) is non-degenerate by specifying the smallest eigenvalue of $\boldsymbol{B}_{NN}$ as a positive number. Also, we assume that the coefficients have a uniform bound, which is common in the analysis of elliptic equations.

**Assumption 4.3.** *The smallest eigenvalue of the symmetric matrix $\boldsymbol{B}_{NN}$, denoted as $\Lambda$, is positive. Besides, $|B_{NN}^{ij}| < 2B_1$, $|a_{NN}^i(\boldsymbol{x})| < B_1$, $\left|\sum_{i=1}^d \partial a_{NN}^i(\boldsymbol{x})/\partial x_i\right| < B_1$, $\forall i, j$ and $\forall \boldsymbol{x} \in \Omega$, for some $B_1 > 0$.*

Next, considering (36) is defined in a compact domain, we can not guarantee the uniqueness of the solution $\hat{p}$ since no boundary condition is specified. Moreover, even if we impose a boundary condition, say Dirichlet condition $\hat{p} = g$ on $\partial\Omega$, we still need extra assumptions on the coefficients to ensure the uniqueness. For the latter, it suffices to take the following assumption.

**Assumption 4.4.**

$$\int_\Omega \sum_{i=1}^d a_{NN}^i v_{x_i} \cdot v + \left(\sum_{i=1}^d \frac{\partial a_{NN}^i}{\partial x_i}\right) v^2 \mathrm{d}\boldsymbol{x} \ge 0, \quad \forall v \in H^1(\Omega).$$

Under Assumption 4.4, one can show that (36) with any Dirichlet condition admits a unique solution by Fredholm alternative and Lax-Milgram theorem. However, we can not specify such a boundary condition since no information on $\partial\Omega$ is provided. Fortunately, we note that the true density $p$ vanishes as $|x| \to \infty$, so it can be assumed that the approximate density $\hat{p}$ has a similar behavior. Although we do not specify any boundary value for $\hat{p}$, we can assume that $\hat{p}$ "almost" vanishes on $\partial\Omega$ as follows.

**Assumption 4.5.** *Let $\|\hat{p}\|_{L^\infty(\partial\Omega)} \le \epsilon_{\hat{p}}$ and $\|\hat{p}\|_{H^1(\partial\Omega)} \le \epsilon_{\hat{p}}$ for some small positive number $\epsilon_{\hat{p}} > 0$.*

Under Assumption 4.4 and 4.5, it can be shown that any two solutions of (36) are close to each other up to accuracy $\epsilon_{\hat{p}}$ by standard elliptic equation analysis.

Now we indicate that the error $\|q - \hat{p}\|_{L^2(\Omega)}$ for any function $q$ is bounded by the loss function $J[q]$ and $\epsilon_{\hat{p}}$.

**Lemma 4.5.** *Assume $\hat{p}$ is a classical solution of (11) with the condition (12). Let $q \in C^2(\bar{\Omega})$ and assume $\|\nabla q\|_{L^2(\partial\Omega)} \le B_2$ for some $B_2 > 0$. If Assumptions 4.3-4.5 hold, then*

$$\|q - \hat{p}\|_{L^2(\Omega)}^2 \le C \left( J[q] + d(1 + \epsilon_{\hat{p}}) J[q]^{\frac{1}{2}} + d(1 + \epsilon_{\hat{p}})\epsilon_{\hat{p}} \right),$$

*where $C$ only depends on $\Omega$, $\Lambda$, $B_1$, $B_2$, $\lambda_1$, $\lambda_2$.*

*Proof.* See Appendix A. □

Next, we estimate $J[\hat{p}]$ via the generalization analysis of FNNs. In the analysis, we redefine the Barron space for two-layer ReLU$^3$ networks and assume $\hat{p}$ is in this Barron space. The definition directly follows the ReLU Barron space proposed in Section 4.2 except that we replace the ReLU activation with the ReLU$^3$ activation. Accordingly, we slightly modify the Barron norm, which is also proposed in [41]. Recall that $\dot{\sigma}$ denotes the ReLU$^3$ activation function, i.e. $\dot{\sigma} = \max(0, x^3/6)$.

Suppose $f : \Omega \to \mathbb{R}$ is a function having the following form,

$$f(\boldsymbol{x}) = \int_{\mathbb{R}\times\mathbb{R}^d} c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x})\rho(\mathrm{d}c, \mathrm{d}\boldsymbol{w}) = \mathbb{E}_\rho[c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x})], \quad \boldsymbol{x} \in \Omega$$

for some probability measure $\rho$ on $\mathbb{R} \times \mathbb{R}^d$, then its ReLU$^3$ Barron norm is defined by

$$\|f\|_{\mathscr{B}_{\dot{\sigma}}} = \inf_{\rho \in P_f} (\mathbb{E}_\rho |c| \|\boldsymbol{w}\|_1^3), \tag{37}$$

where $P_f := \{\rho : f(\boldsymbol{x}) = \mathbb{E}_\rho[c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x})]\}$. And the ReLU$^3$ Barron space is defined by $\mathscr{B}_{\dot{\sigma}} = \{f \in C^0 : \|f\|_{\mathscr{B}_{\dot{\sigma}}} < \infty\}$. Now let us derive the uniform approximation of FNNs in $\mathscr{F}_{2,M,\dot{\sigma},Q}$ for Barron functions.

**Lemma 4.6.** *Given $f \in \mathscr{B}_{\dot{\sigma}}$, there exists some $p_{\mathrm{NN}} \in \mathscr{F}_{2,M,\dot{\sigma},\max\{\|f\|_{\mathscr{B}_{\dot{\sigma}}}/M,1\}}$ such that*

$$\sup_{\boldsymbol{x}\in\Omega} \left| \hat{\mathscr{L}}^* p_{\mathrm{NN}}(\boldsymbol{x}) - \hat{\mathscr{L}}^* f(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x}\in\Omega} \left| p_{\mathrm{NN}}(\boldsymbol{x}) - f(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x}\in\partial\Omega} \left| p_{\mathrm{NN}}(\boldsymbol{x}) - f(\boldsymbol{x}) \right| \le (4B_1 + 2) \|f\|_{\mathscr{B}_{\dot{\sigma}}} \sqrt{d/M}, \tag{38}$$

*Proof.* See Appendix A. □

Next, we introduce the error estimate for the Monte-Carlo integration, which can be directly proved using the Hoeffding's inequality.

**Lemma 4.7.** *Given a compact domain $\Omega$. Suppose $f : \Omega \to \mathbb{R}$ is a function with $\|f\|_\infty < \infty$. Let $\{\boldsymbol{x}_n\}_{n=1}^N$ be a set of uniformly distributed points in $\Omega$. Then for any $\delta \in (0,1)$, with probability at least $1 - \delta$ over the choice of $\boldsymbol{x}_n$,*

$$\left| \frac{|\Omega|}{N} \sum_{n=1}^N f(\boldsymbol{x}_n) - \int_\Omega f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right| \le \sqrt{\frac{2\|f\|_\infty^2 \log(2/\delta)}{N}}.$$

Now, the error estimate for the approximate FP equation is given as follows.

**Lemma 4.8.** *Under Assumption 4.3-4.5, further assume $\hat{p} \in \mathscr{B}_{\dot{\sigma}}$. Let $\boldsymbol{\theta}^S = \mathrm{argmin}_{\boldsymbol{\theta}} J_S[\hat{p}_{\mathrm{NN}}(\cdot, \boldsymbol{\theta})]$ with $\hat{p}_{\mathrm{NN}} \in \mathscr{F}_{2,M,\dot{\sigma},Q}$. Also, suppose $\{\boldsymbol{x}_I^n\}_{n=1}^{N_1} \subset \Omega$, $\{\boldsymbol{x}_{II}^n\}_{n=1}^{N_2} \subset \Omega$, $\{\boldsymbol{x}_{III}^n\}_{n=1}^{N_3} \subset \partial\Omega$ in (17) are uniformly distributed. Then for any $\delta \in (0,1)$, with probability of at least $1 - \delta$ over the choice of these points,*

$$\|\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S) - \hat{p}\|_{L^2(\Omega)}^2 \le C \left( J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)] + d(MQ^4 d^{\frac{1}{2}} + \epsilon_{\hat{p}}) J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)]^{\frac{1}{2}} + d(MQ^4 d^{\frac{1}{2}} + \epsilon_{\hat{p}})\epsilon_{\hat{p}} \right), \tag{39}$$

*and*

$$J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)] \le C \left[ I_1(Q,d,\delta,M,N_1,N_2,N_3) + I_2(Q,\delta,M,N_2) + I_3(\hat{p},d,\delta,M,N_2) \right],$$

*with*

$$I_1 = (Q^8 + 1)\left(d^2\sqrt{\log(d)} + \log(Q^4 + 1) + \sqrt{\log(1/\delta)}\right)M^2(1/\sqrt{N_1} + 1/\sqrt{N_3}),$$

$$I_2 = MQ^4\sqrt{\log(6/\delta)/N_2}\left(MQ^4(\sqrt{\log(6/\delta)/N_2} + 1) + 1\right),$$

$$I_3 = \|\hat{p}\|^2_{\mathcal{B}_{\dot{\sigma}}}d/M + \|\hat{p}\|^2_\infty\log(6/\delta)/N_2 + \epsilon^2_{\hat{p}},$$

*where $C$ only depends on $\Omega, \Lambda, B_1, \lambda_1,$ and $\lambda_2$. Especially, suppose $J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)] \le 1$ and let $N_p := \min\{N_1, N_2, N_3\}$. Take $Q \le O(M^{-\frac{1}{4}}d^{-\frac{1}{8}})$ and $N_p \ge O(\log(1/\delta))$, then*

$$\|\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S) - \hat{p}\|^2_{L^2(\Omega)} \le O\left(d^2(\log(d))^{\frac{1}{4}}MN_p^{-\frac{1}{4}} + d^{\frac{3}{2}}M^{-\frac{1}{2}} + dN_p^{-\frac{1}{2}} + d\epsilon_{\hat{p}}\right),$$

*with an order constant depending on $\Omega, \Lambda, B_1, \lambda_1, \lambda_2, \delta,$ and $\hat{p}$.*

*Proof.* See Appendix A. □

In Lemma 4.8, it implies using $N_{\mathrm{p}} \sim O(M^s)$ with $s > 4$ will reduce the solution error up to $O(d\epsilon_{\hat{p}})$ as $M, N_{\mathrm{p}} \to \infty$. In practice, as an approximation of the original density $p$ which vanishes as $\|\boldsymbol{x}\| \to \infty$, the solution $\hat{p}$ could have a similar behavior. Hence $\epsilon_{\hat{p}}$ is small enough if $\Omega$ is moderately large. And this also leads to a small solution error $\|\hat{p}_{\mathrm{NN}} - \hat{p}\|^2_L(\Omega)$.

Recall in the analysis of regression error for the drift estimator $\boldsymbol{a}$, we derive an error estimate for deep networks of any width and depth. While in the error analysis for the FP solution $\hat{p}$, only results for two-layer shallow networks are derived in the current work. It is promising to develop this analysis for deep networks in the future work.

## 4.4 The main error estimation

Inserting the two error bounds in Lemma 4.4 and Lemma 4.8 into the inequality in (30) and collecting all the assumptions, we can show the following main theorem for the error estimation of the proposed algorithm.

**Theorem 4.1.** *Let $\pi$ be the invariant measure of a Markov process $X_t$ that satisfies Assumptions 4.1 and 4.2. Let $P_{\boldsymbol{a}} \ge 1$ such that $\|\boldsymbol{a}\|_{L^\infty(\Omega)} \le P_{\boldsymbol{a}}$. Given discrete samples $\{\boldsymbol{x}^n\}_{n=0}^N$ of an ergodic measure $\tilde{\pi}$ that is absolutely continuous with respect to the Lebesque measure in $\mathbb{R}^d$, suppose that $\boldsymbol{a}_{NN}$ defined by (7) with components $a_{NN} \in \mathscr{F}_{L,W,ReLU}^{P_{\boldsymbol{a}}}$ is a consistent estimator in the sense of (22) for all $\boldsymbol{x} \in \Omega = [0,1]^d$. Suppose also that $N \ge Pdim(\mathscr{F}_{L,W,ReLU})$. Let the assumptions in Lemma 4.8 be valid, namely the Assumptions 4.3-4.5. Suppose that $\hat{p} \in \mathscr{B}_{\dot{\sigma}}$ is estimated by $\hat{p}_{NN}(\cdot,\boldsymbol{\theta}^S) \in \mathscr{F}_{2,M,\dot{\sigma},Q}$ with $Q \le O(M^{-\frac{1}{4}}d^{-\frac{1}{8}})$, where $\boldsymbol{\theta}_S$ is the global minimizer of the empirical loss function (17). Then, for all $f \in \mathscr{G}_\ell$ as defined in (21), and for any $\delta \in (0,1)$, with probability of at least $1 - \delta$ over the choice of $\{\boldsymbol{x}_I^n\}_{n=1}^{N_1}$, $\{\boldsymbol{x}_{II}^n\}_{n=1}^{N_2}$ and $\{\boldsymbol{x}_{III}^n\}_{n=1}^{N_3}$,*

$$\sup_{f \in \mathscr{G}_\ell}\left|\pi(f) - \int_\Omega f(\boldsymbol{x})\hat{p}_{NN}(\boldsymbol{x},\boldsymbol{\theta}^S)\,d\boldsymbol{x}\right| \le K_3\hat{\pi}(V)\delta t C_{\boldsymbol{a}}\left(d^2WLN^{-1} + d(WL)^2N^{-1} + d^2(WL)^{-4/d}\right)$$

$$+ C_{\hat{p}}\left(d^2(\log(d))^{\frac{1}{4}}MN_p^{-\frac{1}{4}} + d^{\frac{3}{2}}M^{-\frac{1}{2}} + dN_p^{-\frac{1}{2}} + d\epsilon_{\hat{p}}\right), \tag{40}$$

*where, $N_p := \min\{N_1, N_2, N_3\}$ that satisfies $N_p \ge O(\log(1/\delta))$. Here, the term $C_{\boldsymbol{a}} > 0$ depends on $\boldsymbol{a}$ and at most a polynomial in the logarithm of $N, L, W$, and the constant $C_{\hat{p}} > 0$ depends on $\Omega, \delta, \hat{p}, \|f\|_{L^2(\Omega)}$, the regularization weights $\lambda_1, \lambda_2$, and the upper bounds constants $\Lambda, B_1$ defined in Assumption 4.3.*

And the error bound independent of dimensions exponentially is given as follows.

**Theorem 4.2.** *Under the hypothesis of Theorem 4.1, we further assume all components of $\boldsymbol{a}$ are in $\mathscr{B}_{ReLU}$ with Barron norms no greater than $P_{\boldsymbol{a}}$, and let $a_{NN} \in \mathscr{F}_{2,W,ReLU}^{P_{\boldsymbol{a}}}$, then the error bound term $d^2WLN^{-1} + d(WL)^2N^{-1} + d^2(WL)^{-4/d}$ in (40) can be improved to be $d^2WN^{-1} + dW^2N^{-1} + d^2W^{-1}$.*

We should point out that while the results are valid for the global minimizers $\boldsymbol{\theta}_i^a$ in (6) and $\boldsymbol{\theta}_S$ in (18), we do not specify the condition for which such global minimizers are attainable. We directly assume that the minimizers are found, and do not consider the error from the optimization algorithms. In practice, one can not ensure that the global minimizers can be necessarily found by usual optimizers like gradient descent.

Moreover, throughout the convergence analysis, we consider using special FNN class (19) with uniform bounds or (20) with parameter bounds as the hypothesis space, and derive corresponding approximation errors. However, in practical deep learning, one usually uses the general FNN class $\mathscr{F}_{L,W,\sigma}$ since it is closed under gradient descent optimizers and therefore easy for implementation.

## 5    Numerical Examples

In this section, we numerically demonstrate the effectiveness of our proposed methods on two test problems. The first example is a two-dimensional SDE with Student's t stationary distribution. The second example is a 20-dimensional Langevin dynamics associated to Lenard-Jones potential with Gibbs invariant measure.

In our examples, we replace the norm in the first term in (13) with a weighted $L^2(\Omega, \tilde{\pi})$, recalling that $\tilde{\pi}$ denotes the stationary measure of the discrete Markov chain induced by (3). Hence we can directly use the available data set $\mathscr{X} := \{\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{N-1}\}$ as the Monte Carlo integration points. Empirically, we approximate the first term of (13) via the following Monte-Carlo average,

$$\|\hat{\mathscr{L}}q\|^2_{L^2(\Omega, \tilde{\pi})} \approx \frac{1}{|\mathscr{X} \cap \Omega|} \sum_{n=0}^{N-1} \left|\hat{\mathscr{L}}q(\boldsymbol{x}^n)\right|^2 \mathbb{1}_{\Omega}(\boldsymbol{x}^n), \tag{41}$$

where $\mathbb{1}_{\Omega}$ denotes the characteristic function over domain $\Omega$.

### 5.1    Student's t-distribution

Consider a two-dimensional SDE (1) for Student's t-distribution [1] with

$$\boldsymbol{a}(\boldsymbol{x}) = \begin{bmatrix} -\frac{3}{2}x_1 + x_2 \\ \frac{1}{4}x_1 - \frac{3}{2}x_2 \end{bmatrix}, \quad \boldsymbol{b}(\boldsymbol{x}) = \begin{bmatrix} \sqrt{\phi(x_1, x_2)} & 0 \\ -\frac{11}{8}\sqrt{\phi(x_1, x_2)} & \frac{\sqrt{255}}{8}\sqrt{\phi(x_1, x_2)} \end{bmatrix}$$

where $\boldsymbol{x} = (x_1, x_2)$ and $\phi(x_1, x_2) = 1 + \frac{2}{15}(4x_1^2 - x_1 x_2 + x_2^2)$. The stationary density is explicitly given by

$$p(x_1, x_2) = \frac{2}{\pi\sqrt{15}}\left(\phi(x_1, x_2)\right)^{-3}. \tag{42}$$

#### 5.1.1    Data generation and implementation details

The time series dataset $\{\boldsymbol{x}^i\}_{i=0}^{N}$ is generated by EM scheme (3) with $\delta t = 0.05$ and $N = 2 \times 10^7$. The bounded domain $\Omega$ is set as $[-4, 4] \times [-6, 6]$ such that over 98% points are in $\Omega$.

In our implementation, we use 6-hidden-layer ResNets (discussed in Section 3.1) with the same width 50 per hidden layer and the smooth Mish activation function [45]. To learn $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{b}_{\mathrm{NN}}$, Adam algorithm is applied to optimize the loss (6) and (8) with batch size 10,000 for $T = 20,000$ iterations. We use an initial learning rate of $10^{-4}$. The learning rate follows cosine decay with the increasing training iterations, i.e., the learning rate decays by multiplying a factor $0.5(cos(\frac{\pi t}{T}) + 1)$, where $t$ is the current iteration. To solve the PDE (10), we optimize the loss (13) with $\lambda = 1, \gamma = 500$. In Adam, we use the batch size 10,000 for first term in (13) and 4,000 for the boundary term while the second term is approximated by $300^2$ Gaussian quadrature points. The learning rate is initialized by $10^{-3}$ and follows cosine decay.

#### 5.1.2    Identification of the drift and diffusion coefficients.

To evaluate the accuracy, we define relative $L_2$ error as follows,

$$\frac{\|f - \hat{f}\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}}, \tag{43}$$

where $f$ and $\hat{f}$ represents the true function and the approximate function, respectively. Numerically, we approximate the integral over 10000 Gaussian quadrature points in $\Omega$.

The relative $L_2$ error between $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{a}$ is $2.94 \times 10^{-2}$. Figure 1 displays the spatial profile of the first components of $\boldsymbol{a}$, $\boldsymbol{a}_{\mathrm{NN}}$, and their difference on the computational domain $\Omega$. The relative $L_2$ error between $\boldsymbol{B}_{\mathrm{NN}}$ and $\boldsymbol{bb}^\top$ is $6.98 \times 10^{-2}$. To check the pointwise accuracy of the estimates, We plot the first diagonal components of $\boldsymbol{bb}^\top$, $\boldsymbol{B}_{\mathrm{NN}}$ on the computational domain $\Omega$ and their difference in Figure 2. We can see our method works well on fitting the drift and diffusion terms.
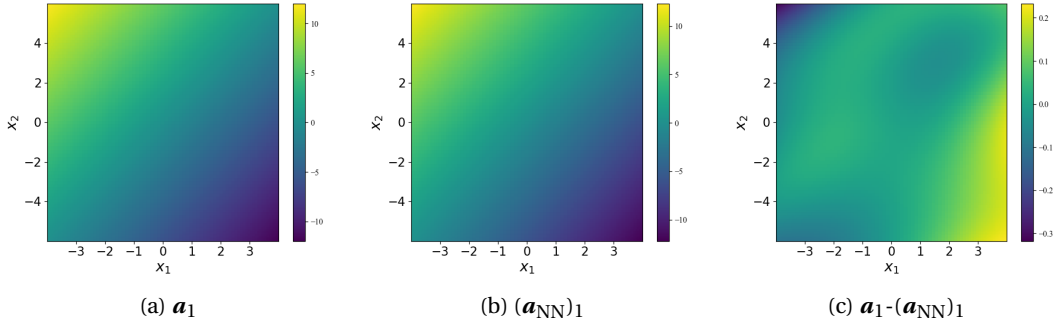
13

Figure 1: The comparison of the first component of drift term. (a) $\boldsymbol{a}_1$, (b) $(\boldsymbol{a}_{\text{NN}})_1$, and (c) their difference.

Given the approximate drift and diffusion coefficients, we now empirically validate the result in Lemma 4.1 on the computational domain

$$\boldsymbol{x}^{n+1} - \boldsymbol{x}^n = \boldsymbol{a}_{\text{NN}}(\boldsymbol{x}^n)\delta t + \boldsymbol{U}(\boldsymbol{x}^n)\boldsymbol{S}(\boldsymbol{x}^n)^{\frac{1}{2}}\boldsymbol{U}(\boldsymbol{x}^n)^\top \sqrt{\delta t}\boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I}_2), \tag{44}$$

where $\boldsymbol{U}(\boldsymbol{x}^n)\boldsymbol{S}(\boldsymbol{x}^n)\boldsymbol{U}(\boldsymbol{x}^n)^\top$ is the eigendecomposition of $\boldsymbol{B}_{\text{NN}}(\boldsymbol{x}^n)$. We denote these empirical statistics to be defined with respect to the distribution $\hat{\pi}^{EM}$ that approximates $\hat{\pi}$. Compare to the ground truth statistics, the statistics of $\hat{\pi}^{EM}$ are subjected to errors from the estimation of $\boldsymbol{a}$, $\boldsymbol{B}$, and from the EM integration. In Table 1, we note that when $\delta t = 0.05$, the covariance error of $\boldsymbol{a}_{\text{NN}}, \boldsymbol{B}_{\text{NN}}$ is comparable to the error of $\boldsymbol{a}, \boldsymbol{b}$. When $\delta t = 0.01$, the covariance of $\boldsymbol{a}_{\text{NN}}, \boldsymbol{B}_{\text{NN}}$ becomes much closer to the ground truth than $\delta t = 0.05$.

| Distribution | $\pi$ | | $\tilde{\pi} := \pi^{\text{EM}}$ | | $\hat{\pi}^{\text{EM}}$ | | | |
|---|---|---|---|---|---|---|---|---|
| $\delta t$ | N/A | | 0.05 | | 0.05 | | 0.01 | |
| mean | $[0.000$ | $0.000]$ | $[-0.002$ | $0.000]$ | $[0.001$ | $0.004]$ | $[0.002$ | $-0.003]$ |
| covariance | $\begin{bmatrix}1.000 & 0.500 \\ 0.500 & 4.000\end{bmatrix}$ | | $\begin{bmatrix}1.127 & 0.499 \\ 0.499 & 4.398\end{bmatrix}$ | | $\begin{bmatrix}1.115 & 0.490 \\ 0.490 & 4.347\end{bmatrix}$ | | $\begin{bmatrix}1.013 & 0.501 \\ 0.501 & 3.984\end{bmatrix}$ | |

Table 1: Comparison of mean and covariance statistics corresponding to the ground truth distribution $\pi$, discrete Markov chain induced by EM scheme in (3), $\tilde{\pi} := \pi^{EM}$, and the discrete Markov chain generated by (44) for various $\delta t$ whose invariant distribution is denoted as $\hat{\pi}^{\text{EM}}$.

### 5.1.3   Computation of the density function

We optimize the loss (13) with $\boldsymbol{a}_{\text{NN}}$ and $\boldsymbol{B}_{\text{NN}}$ and obtain the solution $\hat{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$. The relative $L_2$ error between $\hat{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$ and the true density (42) is $6.78 \times 10^{-2}$. To quantify the error induced by the regression alone, we replace $\boldsymbol{a}_{\text{NN}}$ and $\boldsymbol{B}_{\text{NN}}$ of $\hat{\mathscr{L}}^*$ in (13) with the underlying coefficients, $\boldsymbol{a}$ and $\boldsymbol{b}\boldsymbol{b}^\top$, and optimize (13) with differential operator $\mathscr{L}^*$ in the first term. We denote the corresponding solution by $\tilde{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$. The relative $L_2$ error between $\tilde{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$ and the true density (42) is $3.97 \times 10^{-2}$. We can see $\hat{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$ achieves the error of same magnitude as $\tilde{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$. Figure 3 shows the true density and the differences between the true density and the network solutions $\hat{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$, $\tilde{p}_{\text{NN}}(\cdot; \boldsymbol{\theta})$, plotted as functions of the computational domain $\Omega$. Notice that the errors are more prominent when the coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$ are estimated, as expected.

### 5.2   The Langevin dynamics

We consider a molecular model describing the dynamics of $M$ atoms with mass 1. We assume the $M$ particles are spaced in a chain with a periodic boundary condition. Let the equilibrium distance between two neighboring particles be $a_0$, then the equilibrium position of the $m$-th particle is $ma_0$. Denote $r_m$ as the displacement of the $m$-th particle from its equilibrium position, and denote $v_m$ as its velocity. The Langevin dynamics of this model is described as follows

$$\dot{\boldsymbol{v}} = -\nabla_{\boldsymbol{r}} U(\boldsymbol{r}) - \gamma \boldsymbol{v} + \sqrt{2k_B T \gamma}\dot{\boldsymbol{W}}_t,$$
$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \tag{45}$$

14
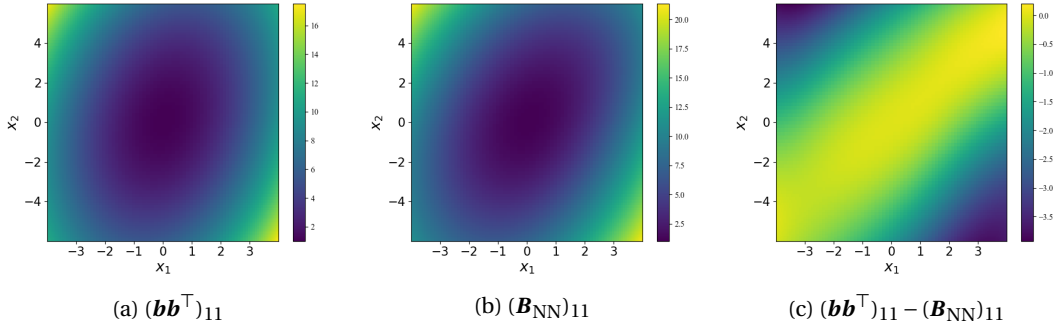
Figure 2: The comparison of first component of $\boldsymbol{b}\boldsymbol{b}^\top$. (a) $(\boldsymbol{b}\boldsymbol{b}^\top)_{11}$, (b) $(\boldsymbol{B}_{\mathrm{NN}})_{11}$ and (c) their difference.
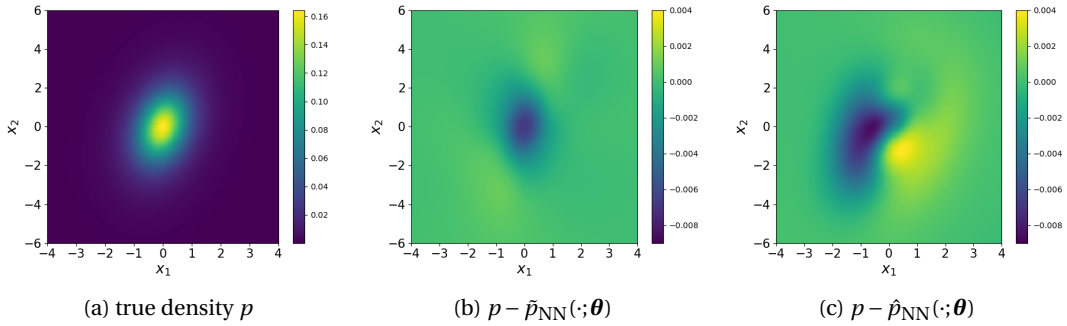


Figure 3: The comparison of solutions. (a) True density $p$, (b) difference between $p$ and $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ and (c) difference between $p$ and $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$. Here $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ is obtained by optimizing (13) with $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$, while $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ is obtained by optimizing (13) with $\boldsymbol{a}$ and $\boldsymbol{b}\boldsymbol{b}^\top$.

where $\boldsymbol{v} = [v_1,\cdots,v_M]^\top$ and $\boldsymbol{r} = [r_1,\cdots,r_M]^\top$ are the velocities and displacement of all particles; $\boldsymbol{W}_t = [W_t^{(1)},\cdots,W_t^{(M)}]^\top$ is an $M$-dimensional Wiener process; $U$ is some potential function; $\gamma$ is the friction constant; $k_B T$ is the temperature. The mass of particles is set to be unity in (61). The equilibrium distribution of (61) is given by

$$p(\boldsymbol{v},\boldsymbol{r}) \propto \exp\left[-\frac{1}{k_B T}\left(U(\boldsymbol{r}) + \frac{1}{2}|\boldsymbol{v}|^2\right)\right]. \tag{46}$$

In the numerical simulation, we take the Lennard-Jones potential [26], which is given by

$$U(\boldsymbol{r}) = \sum_{i=1}^{M}\sum_{j=i-2}^{i-1} \psi(r_i - r_j + (i-j)a_0), \quad r_0 := r_M, \ r_{-1} := r_{M-1} \tag{47}$$

with

$$\psi(r) = |r|^{-12} - |r|^{-6}. \tag{48}$$

The model parameters of this example is set to be $a_0 = 1$, $\gamma = 0.5$, $k_B T = 0.25$, $M = 10$.

### 5.2.1 Data generation

We generate the data by Euler-Maruyama discretization, namely,

$$\begin{aligned}
\boldsymbol{v}^{n+1} &= \boldsymbol{v}^n - (\nabla_{\boldsymbol{r}^n} U(\boldsymbol{r}^n) + \gamma\boldsymbol{v}^n)\delta t + \sqrt{2k_B T\gamma\delta t}\,\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim (\mathcal{N}(0,1))^M, \\
\boldsymbol{r}^{n+1} &= \boldsymbol{r}^n + \boldsymbol{v}^n\delta t,
\end{aligned} \tag{49}$$

for $n = 0,1,\cdots,N-1$ with the initial states

$$\boldsymbol{v}^0 = 0, \quad \boldsymbol{r}^0 \sim (\mathcal{N}(0,0.01))^M.$$

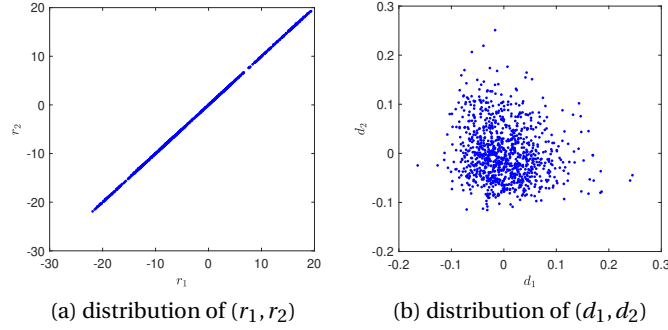(a) distribution of $(r_1, r_2)$         (b) distribution of $(d_1, d_2)$

Figure 4: *The distribution of the original dataset $\mathscr{X}$ in the $(r_1, r_2)$-plane and the distribution of the transformed dataset $\hat{\mathscr{X}}$ in the $(d_1, d_2)$-plane.*

In this example, we set $\delta t = 0.0005$ and $N = 10^7$. Following the notation in Section 3.2, we denote $\mathscr{X} := \{v^n, r^n\}_{n=0}^N$ as the original data set. If we visualize the distribution of $\mathscr{X}$ by projecting it onto the $(r_1, r_2)$-plane (Figure 4), it is observed that displacement components are distributed near a straight line. To simplify the computation and visualization, we consider a coordinate transformation that will map $\mathscr{X}$ to be enclosed by a hyper-rectangle. Specifically, we introduce the following coordinate transformation,

$$\mathscr{T} : \mathbb{R}^M \to \mathbb{R}^{M-1}, \quad d := [d_1, \cdots, d_{M-1}]^\top = \mathscr{T}(r) = [r_2 - r_1, \cdots, r_M - r_{M-1}]^\top,$$

where $d$ is called the relative displacement. Note that the map $\mathscr{T}$ implies $r_1 - r_M = -\sum_{m=1}^{M-1} d_m$. If we define the transformed dateset $\hat{\mathscr{X}} := \{v^n, d^n\}_{n=0}^N$ with $d^n = \mathscr{T}(r^n)$ and project it onto the $(d_1, d_2)$-plane (Figure 4), then it is observed that most points in $\hat{\mathscr{X}}$ are located near the origin and form a circular region. Consequently, we apply the proposed method to the transformed dataset $\hat{\mathscr{X}}$ in the practical computation.

### 5.2.2 Identification of the drift and diffusion coefficients.

Now we aim to identify the drift term $a(v, r)$ and the diffusion $bb^\top$ of the underlying dynamics. Due to transformation $\mathscr{T}$, we define $\hat{a}(v, d) := a(v, r)$ and aim to identify $\hat{a}$ by the optimization (6) using the dataset $\hat{\mathscr{X}}$. Note $\hat{a}(v, d)$ is a vector-valued function with $(2M - 1)$-dimensional inputs and $2M$-dimensional outputs. In this example, to obtain higher accuracy, we use an individual neural network with $(2M - 1)$-dimensional inputs and scalar outputs to approximate the each component of $\hat{a}(v, d)$, solving the regression problem in (6). In this application, this is a regression over training data set $(\hat{\mathscr{X}}, \mathscr{Y})$, where $\mathscr{Y} := \{\frac{v^{n+1} - v^n}{\delta t}, \frac{r^{n+1} B r^n}{\delta t}\}_{n=0}^{N-1}$

In practice. we set each component of $\hat{a}_{\text{NN}}$ to be a fully connected ReLU network with 3 layers and 100 neurons in each layer. We employ Adams optimizer with 1000 epochs, and the learning rates are set to decay from $10^{-3}$ to $10^{-5}$. The relative $\ell^2$ training errors for the first $M$ components corresponding to the velocity are observed to be between $3.87 \times 10^{-2}$ and $5.60 \times 10^{-2}$, and the errors for the next $M$ components are between $5.04 \times 10^{-5}$ and $8.99 \times 10^{-5}$.

Next, we consider the approximation $B_{NN}$ to the constant matrix $bb^\top$ using the formula in (9). In this example, since $bb^\top$ is a diagonal matrix, we also set $B_{\text{NN}}$ to be diagonal with components $(b_{11}, \ldots, b_{2M,2M})$. The errors $|b_{kk} - (bb^\top)_{kk}|$ for the first $M$ components are observed to be between $6.32 \times 10^{-6}$ and $2.67 \times 10^{-6}$, and the errors for the next $M$ components are between $8.07 \times 10^{-13}$ and $3.63 \times 10^{-12}$.

Similar to the previous example, we simulate the dynamics by the obtained $\hat{a}_{\text{NN}}$ and $B_{\text{NN}}$,

$$\begin{aligned}
v^{n+1} - v^n &= (\hat{a}_{\text{NN}})_{1:M}(v^n, d^n)\delta t + (B_{\text{NN}})^{\frac{1}{2}} \sqrt{\delta t} \xi_n, \qquad \xi_n \sim \mathcal{N}(0, I_M), \\
r^{n+1} - r^n &= (\hat{a}_{\text{NN}})_{M+1:2M}(v^n, d^n)\delta t,
\end{aligned} \tag{50}$$

and compare it with the ground truth. For the covariance of $\pi$, Monte Carlo integration with $10^8$ points is used. For the statistics of $\tilde{\pi}$ and $\hat{\pi}^{EM}$, we generate a sequence of $10^7$ points. The information is shown in Table 2 for the components $v_1$ and $d_1$. Notice that in this case, the statistical error for estimating $\hat{\pi}^{EM}$ is not much worse than the Monte-Carlo error of $\tilde{\pi}$.

| Distribution | $\pi$ | | $\tilde{\pi}$ | | $\hat{\pi}^{\text{EM}}$ | |
|---|---|---|---|---|---|---|
| $\delta t$ | N/A | | 0.0005 | | 0.0005 | |
| mean | $\begin{bmatrix} 0 & 0 \end{bmatrix}$ | | $\begin{bmatrix} -0.00363 & -0.00013 \end{bmatrix}$ | | $\begin{bmatrix} -0.00153 & -0.00003 \end{bmatrix}$ | |
| covariance | $\begin{bmatrix} 0.40229 & -0.01749 \\ -0.01749 & 0.00245 \end{bmatrix}$ | | $\begin{bmatrix} 0.37816 & 0.00008 \\ 0.00008 & 0.00292 \end{bmatrix}$ | | $\begin{bmatrix} 0.40916 & 0.00041 \\ 0.00041 & 0.00314 \end{bmatrix}$ | |

Table 2: Comparison of mean and covariance statistics ($\boldsymbol{v}_1$ and $\boldsymbol{d}_1$) corresponding to the ground truth distribution $\pi$ , discrete Markov chain induced by EM scheme in (3), $\tilde{\pi}$, and the discrete Markov chain generated by (50) for $\delta t = 0.0005$ whose invariant distribution is denoted as $\hat{\pi}^{\text{EM}}$.

.

### 5.2.3 Computation of the density function

In this section, we aim to recover the equilibrium density function based on the obtained $\{\hat{a}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_k)\}$ and $\boldsymbol{B}_{\text{NN}}$. We let $p(\boldsymbol{v}, \boldsymbol{r})$ be the original density function in $(\boldsymbol{v}, \boldsymbol{r})$-coordinates, and define $\hat{p}(\boldsymbol{v}, \boldsymbol{d}) := p(\boldsymbol{v}, \boldsymbol{r})$ be the density function under transformation $\mathcal{T}$. Since $p(\boldsymbol{v}, \boldsymbol{r})$ satisfies (11), we can derive the PDE for $\hat{p}(\boldsymbol{v}, \boldsymbol{d})$, which is given by

$$-\sum_{k=1}^{M} \frac{\partial}{\partial v_k}(\hat{p}\hat{a}_k) - \sum_{k=M+1}^{2M-1} \frac{\partial}{\partial d_{k-M}}\left(\hat{p}(\hat{a}_{k+1} - \hat{a}_k)\right)$$
$$+ \frac{1}{2}\sum_{k=1}^{M}(\boldsymbol{bb}^\top)_{kk}\frac{\partial^2}{\partial v_k^2}\hat{p} + \frac{1}{2}(\boldsymbol{bb}^\top)_{M+1,M+1}\frac{\partial^2}{\partial d_1^2}\hat{p}$$
$$+ \frac{1}{2}\sum_{k=2}^{M-1}(\boldsymbol{bb}^\top)_{k+M,k+M}\left(\frac{\partial}{\partial d_k} - \frac{\partial}{\partial d_{k-1}}\right)^2\hat{p} + \frac{1}{2}(\boldsymbol{bb}^\top)_{2M,2M}\frac{\partial^2}{\partial d_{M-1}^2}\hat{p} = 0, \quad (51)$$

where $\hat{a}_k$ denotes the $k$-th component of $\hat{\boldsymbol{a}}$.

Once the drift and diffusion coefficients are estimated, we substitute $\hat{a}_k$ with the $k$th FNN estimate, denoted as $\hat{a}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_k)$, and $(\boldsymbol{bb}^\top)_{k,k}$ with the diagonal components of the estimated diffusion matrix, $b_{kk} := (\boldsymbol{B}_{\text{NN}})_{kk}$, such that (51), becomes,

$$-\sum_{k=1}^{M} \frac{\partial}{\partial v_k}(\hat{p}\hat{a}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_k)) - \sum_{k=M+1}^{2M-1} \frac{\partial}{\partial d_{k-M}}\left(\hat{p}(\hat{a}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_{k+1}) - \hat{a}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_k))\right)$$
$$+ \frac{1}{2}\left(\sum_{k=1}^{M} b_{kk}\frac{\partial^2 \hat{p}}{\partial v_k^2} + b_{M+1,M+1}\frac{\partial^2 \hat{p}}{\partial d_1^2} + \sum_{k=2}^{M-1} b_{k+M,k+M}\left(\frac{\partial}{\partial d_k} - \frac{\partial}{\partial d_{k-1}}\right)^2\hat{p} + b_{2M,2M}\frac{\partial^2 \hat{p}}{\partial d_{M-1}^2}\right) = 0, \quad (52)$$

Next, we select a bounded domain in which the PDE (52) will be solved. Our choice is to use a hyperrectangle $\Omega = \prod_{k=1}^{2M-1}[c_k - s_k, c_k + s_k]$ to enclose most of the points in $\hat{\mathcal{X}}$. At the same time, we expect $\Omega$ to be also densely covered by the points in $\hat{\mathcal{X}}$. By this principle, we set $c_k$ as the component-wise mean of the points in $\hat{\mathcal{X}}$, namely,

$$c_k = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} v_k, & \text{for } k = 1, \cdots, M, \\ \frac{1}{N}\sum_{n=1}^{N} d_k, & \text{for } k = M+1, \cdots, 2M-1, \end{cases} \quad (53)$$

and set $s_k$ empirically as follows.

$$s_k = \begin{cases} 1.0, & \text{for } k = 1, \cdots, M, \\ 0.1, & \text{for } k = M+1, \cdots, 2M-1. \end{cases} \quad (54)$$

For clarity, we display the projections of $\hat{\mathcal{X}}$ and $\Omega$ onto coordinate planes in Figure 5.

We take a neural network $\hat{p}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta})$ to approximate $\hat{p}(\boldsymbol{v}, \boldsymbol{d})$. Then we solve the PDE (52) with the least square method introduced in Section 3.3 to determine $\hat{p}_{\text{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta})$. Specifically, we solve the least-square problem in (17) with $\gamma = 0$, ignoring the artificial boundary constraint since the function values at the boundary $\partial\Omega$ are small, they range from $7 \times 10^{-7}$ to $4 \times 10^{-6}$. Meanwhile, 90% of the points in $\hat{\mathcal{X}} \cap \Omega$ are selected as the training set, denoted as $\hat{D}_{\text{T}}$, and the other 10% are chosen as the testing set, denoted as $\hat{D}_{\text{S}}$, for the evaluation of the solution error. In practice. we set each $\hat{p}_{\text{NN}}$ to be a fully-connected network having 3 layers and 100 neurons in each layer with activation
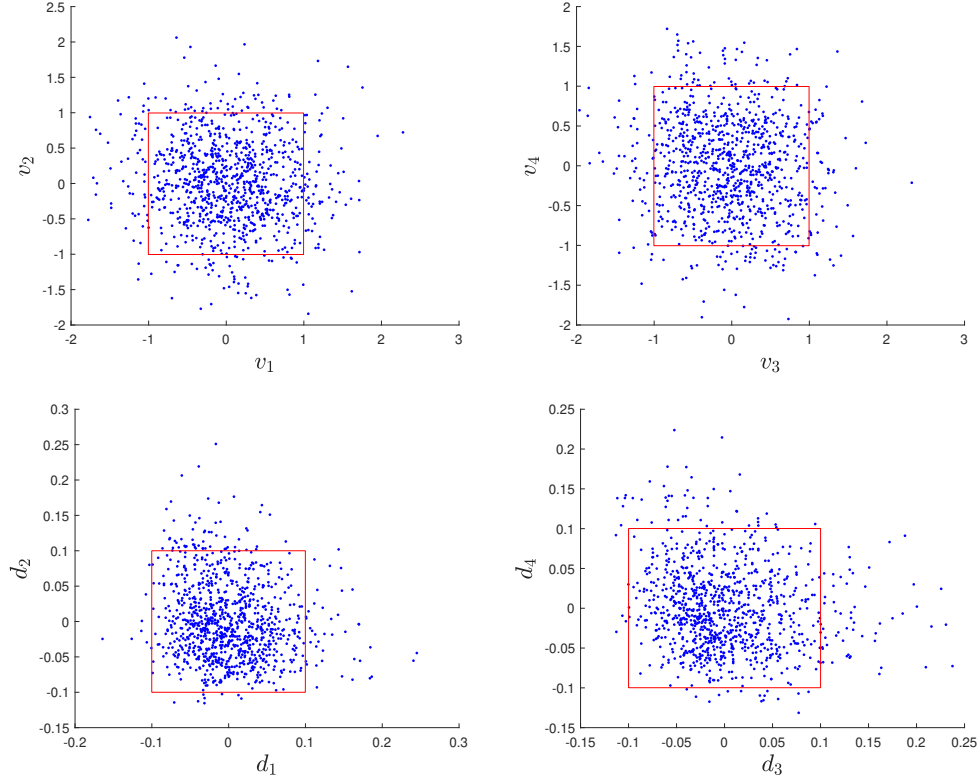
Figure 5: *The projections of the dataset $\mathcal{X}$ (blue points) and the enclosing region $\Omega$ (red boxes) onto $(v_1, v_2)$, $(v_3, v_3)$, $(d_1, d_2)$, $(d_3, d_4)$-planes*

function $\max\{x^3, 0\}$. Adams optimizer is used to solve the optimization with 1000 epochs, and the learning rates are set to decay from $10^{-4}$ to $10^{-5}$. Once $\hat{p}_{\mathrm{NN}}$ is obtained, we evaluate the result by computing the error between $\hat{p}_{\mathrm{NN}}(\boldsymbol{v}, \boldsymbol{d})$ and the true density function $\hat{p}(\boldsymbol{v}, \boldsymbol{d})$. From (46)-(47), we directly have the expression of $\hat{p}(\boldsymbol{v}, \boldsymbol{d})$, namely,

$$\hat{p}(\boldsymbol{v}, \boldsymbol{d}) = c \cdot \hat{p}_0(\boldsymbol{v}, \boldsymbol{d}) := c \cdot \exp\left[-\frac{1}{k_B T}\left(\hat{U}(\boldsymbol{d}) + \frac{1}{2}|\boldsymbol{v}|^2\right)\right] \tag{55}$$

with

$$\hat{U}(\boldsymbol{d}) = \psi\left(-\sum_{i=1}^{M-1} d_i + a_0\right) + \psi\left(-\sum_{i=1}^{M-2} d_i + 2a_0\right) + \psi(d_1 + a_0) + \psi\left(-\sum_{i=2}^{M-1} d_i + 2a_0\right)$$
$$+ \sum_{i=3}^{M} \psi(d_{i-1} + a_0) + \sum_{i=3}^{M} \psi(d_{i-1} + d_{i-2} + 2a_0), \tag{56}$$

where $c$ is determined by the uniform condition

$$c = \left(\int_{\mathbb{R}^{2M-1}} \hat{p}_0(\boldsymbol{v}, \boldsymbol{d})\right)^{-1}. \tag{57}$$

Then the relative $\ell^2$ error between $\hat{p}_{\mathrm{NN}}(\boldsymbol{v}, \boldsymbol{d})$ and $\hat{p}(\boldsymbol{v}, \boldsymbol{d})$ can be computed according to (43) with $L^2(\Omega)$ replaced by $L^2(\hat{D}_S)$, where the integral is replaced by an average over the testing data set $\hat{D}_S$. In this numerical result, we found thatthe relative $\ell^2$ error of the computed density function $\hat{p}_{\mathrm{NN}}$ is $5.402 \times 10^{-2}$. In Figure 6, we also show the marginal densities of $\hat{p}_{\mathrm{NN}}$

$$\hat{p}_{\mathrm{NN},k}^{\mathrm{marginal}}(v_k) := \int_{(\boldsymbol{v}, \boldsymbol{d}) \backslash v_k \in \mathbb{R}^{2M-2}} \hat{p}_{\mathrm{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}), \quad \text{for } k = 1, \cdots, M,$$

$$\hat{p}_{\mathrm{NN},k}^{\mathrm{marginal}}(d_k) := \int_{(\boldsymbol{v}, \boldsymbol{d}) \backslash d_k \in \mathbb{R}^{2M-2}} \hat{p}_{\mathrm{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}), \quad \text{for } k = M+1, \cdots, 2M-1, \tag{58}$$
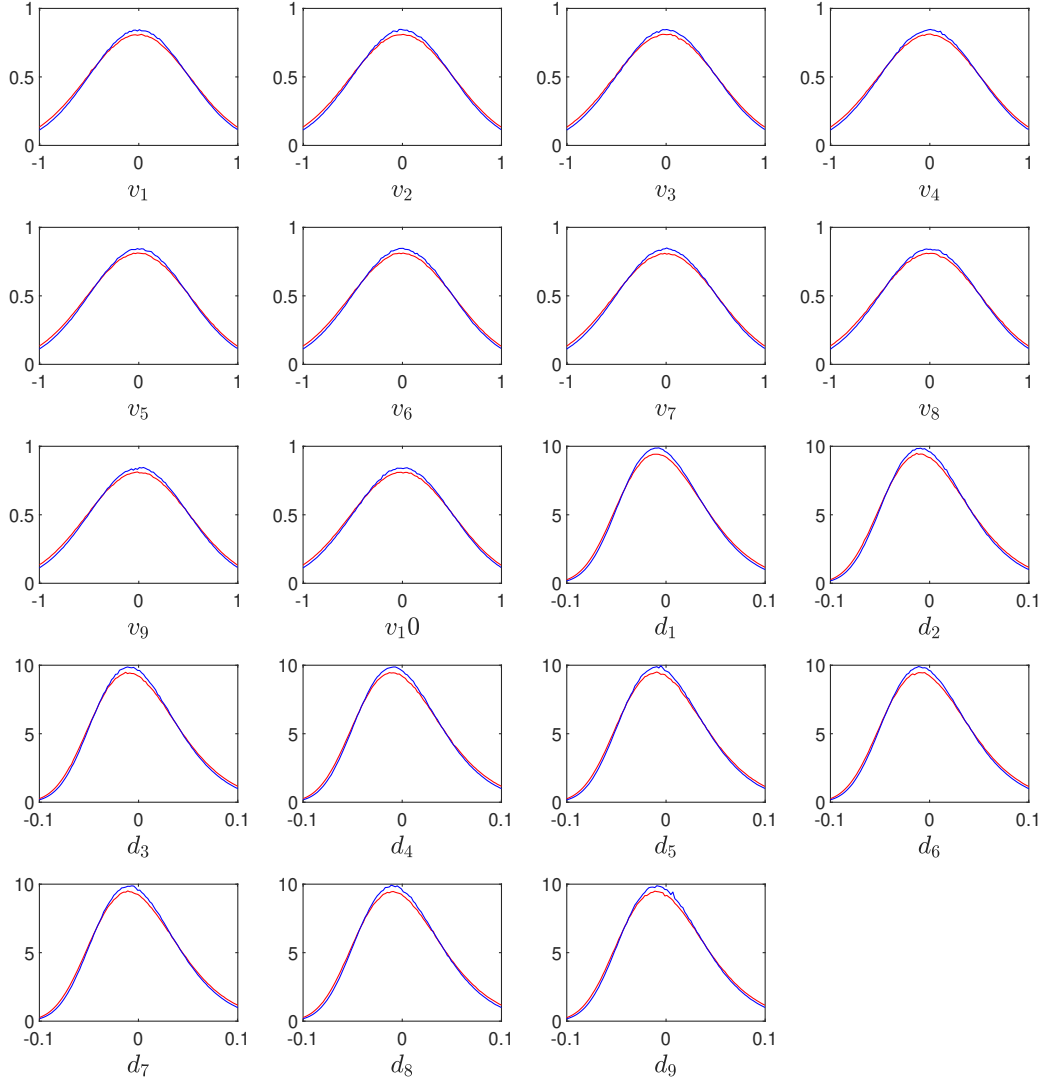
Figure 6: *Marginal densities of the computed density function $\hat{p}_{NN}$ (red curves) and the true density function $\hat{p}$ (blue curves) for all components.*

compared with the following true marginal densities

$$
\begin{aligned}
\hat{p}_k^{\text{marginal}}(v_k) &:= \int_{(\boldsymbol{v},\boldsymbol{d})\setminus v_k \in \mathbb{R}^{2M-2}} \hat{p}(\boldsymbol{v},\boldsymbol{d}), \quad \text{for } k = 1,\cdots,M, \\
\hat{p}_k^{\text{marginal}}(d_k) &:= \int_{(\boldsymbol{v},\boldsymbol{d})\setminus d_k \in \mathbb{R}^{2M-2}} \hat{p}(\boldsymbol{v},\boldsymbol{d}), \quad \text{for } k = M+1,\cdots,2M-1,
\end{aligned}
\tag{59}
$$

where the integrals in (57), (58) and (59) are computed by the Monte Carlo method. Notice the accurate estimation of the marginal densities of the velocity components that are Gaussian and the marginal densities of the relative displacement components that are non-symmetric.

## 6 Conclusion

In this paper, we developed a deep learning-based method to estimate the stationary density of an unknown Itô diffusion SDE from a time series induced by the Euler-Maruyama solver. Neural networks are employed to approximate the drift, diffusion, and stationary density of the underlying dynamics. In our method, the first step is learning the drift and diffusion coefficients by solving least square regressions corresponding to the available dataset, and the second step is solving the steady-state Fokker-Plank equation formed by the estimated drift and diffusion coefficients. Theoretically, we deduced an error bound for the proposed approach for an SDE with global Lipschitz drift coefficients and constant diffusion matrix, accounting errors contributed by the discretization of the SDE in the training data, the regression of the drift terms using fully-connected ReLU networks with arbitrary width and layers, and the regression solution to the Fokker-Planck PDE using a fully-connected two-layer neural network with the ReLU$^3$ activation function. This error bound is deduced under various assumptions that underpin the perturbation theory result in [72], generalization errors in approximating Lipschitz continuous functions in [28] and in solving PDEs in [41].

From this theoretical study, we observe two difficult aspects that warrant careful treatments in future studies. The first issue is concerning the incompatibility of the topologies that characterize the perturbation theory and machine learning generalization theory. Since the bound in (24) is stronger than an $L^2$-error bound in generalization theory, one requires a tacit assumption of consistency in the sense of (22), which is not easily verified in practice. The second issue is concerning the incompatibility of the computational and physical domains, which is admitted under the Assumption 4.2. Particularly, while the underlying stochastic process is defined on $\mathbb{R}^d$, the error estimation that accounts for finite samples the training for $\boldsymbol{a}$ and $\hat{p}$ is not easily guaranteed for the entire unbounded domain. Besides, it is also only feasible to employ the computation over a bounded domain.

In numerical simulations, we verified the effectiveness of the proposed method on two examples, a two-dimensional Student's t-distribution, and the 20-dimensional Langevin dynamics. Although the proposed data-driven methods show encouraging numerical results on the approximation of the invariant statistics and densities, the empirical loss function in (17) requires samples $\boldsymbol{x}_I, \boldsymbol{x}_{II}^n$ and $\boldsymbol{x}_{III}^n$. Such a requirement may not be viable when the geometry is more complicated than hypercubes. While sampling the first term in (17) is avoidable by a Monte-Carlo over the available time series as we have done in our numerical examples, generating samples for the second and third terms in the loss function in (17) is unavoidable. In the future, we plan to consider different penalties such as the one proposed in [70] which requires no additional samples other than the available time series.

## Acknowledgment

## References

[1] TA Averina and SS Artemiev. Numerical solution of systems of stochastic differential equations. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 3(4):267–286, 1988.

[2] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

[3] A. Caragea, P. Petersen, and F. Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *arXiv e-prints*, arXiv:2011.09363, 2020.

[4] Pierre-Henri Chavanis. Nonlinear mean field fokker-planck equations. application to the chemotaxis of biological populations. *The European Physical Journal B*, 62(2):179–208, 2008.

[5] Minshuo Chen, Haoming Jiang, Wenjing Liao, and T. Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks. *arXiv: Learning*, 2019.

[6] Ziang Chen, Jianfeng Lu, and Yulong Lu. On the representation of solutions to elliptic pdes in barron spaces. *arxiv:2106.07539*, 2021.

[7] Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep relu networks? *CoRR*, arXiv:1911.12360, 2019.

[8] Patrick Cheridito, Arnulf Jentzen, and Florian Rossmannek. Non-convergence of stochastic gradient descent in the training of deep neural networks. *Journal of Complexity*, 64:101540, 2021.

[9] Zhiyan Ding, Shi Chen, Qin Li, and Stephen Wright. Overparameterization of deep resnet: zero loss and mean-field analysis. *arxiv:2105.14417*, 2021.

[10] M. W. M. G. Dissanayake and N. Phan-Thien. Neural-network-based Approximations for Solving Partial Differential Equations. *Comm. Numer. Methods Engrg.*, 10:195–201, 1994.

[11] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv e-prints*, arXiv:1810.02054, 2018.

[12] Chenguang Duan, Yuling Jiao, Yanming Lai, Xiliang Lu, and Zhijian Yang. Convergence rate analysis for deep ritz method. *arxiv:2103.13330*, 2021.

[13] Weinan E, Chao Ma, Stephan Wojtowytsch, and Lei Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv e-prints*, arXiv:2009.10713, 2020.

[14] Weinan E, Chao Ma, and Lei Wu. A priori estimates of the population risk for two-layer neural networks. *Communications in Mathematical Sciences*, 17(5):1407 – 1425, 2019.

[15] Till Daniel Frank. *Nonlinear Fokker-Planck equations: fundamentals and applications*. Springer Science & Business Media, 2005.

[16] Y. Gu, H. Yang, and C. Zhou. SelectNet: Self-paced Learning for High-dimensional Partial Differential Equations. *Journal of Computational Physics*, 441:110444, 2021.

[17] Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

[18] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci.*, 115(34):8505–8510, 2018.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] Siegfried Hess. Fokker-planck-equation approach to flow alignment in liquid crystals. *Zeitschrift für Naturforschung A*, 31(9):1034–1037, 1976.

[21] Sean Hon and Haizhao Yang. Simultaneous neural network approximations in sobolev spaces. *arxiv:2109.00161*, 2021.

[22] Jonathan Huggins and James Zou. Quantifying the accuracy of approximate diffusions and markov chains. In *Artificial Intelligence and Statistics*, pages 382–391. PMLR, 2017.

[23] M. Hutzenthaler, A. Jentzen, Th. Kruse, and T. A. Nguyen. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. Technical Report 2019-10, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2019.

[24] Jenq-Neng Hwang, Shyh-Rong Lay, and Alan Lippman. Nonparametric multivariate density estimation: a comparative study. *IEEE Transactions on Signal Processing*, 42(10):2795–2810, 1994.

[25] Edmond Iancu, Andrei Leonidov, and Larry McLerran. Nonlinear gluon evolution in the color glass condensate: I. *Nuclear Physics A*, 692(3-4):583–645, 2001.

[26] Yuji Ishimori. Solitons in a one-dimensional lennard-jones lattice. *Progress of Theoretical Physics*, 68:402–410, 1982.

[27] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018.

[28] Y. Jiao, G. Shen, Y. Lin, and J. Huang. Deep nonparametric regression on approximately low-dimensional manifolds. *arXiv e-prints*, arXiv:2104.06708, 2021.

[29] S. Justin and S. Konstantinos. Dgm: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018.

[30] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving parametric pde problems with artificial neural networks. *arXiv: Numerical Analysis*, 2017.

[31] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. *arXiv e-prints*, arXiv:1412.6980, 2014.

[32] Pankaj Kumar and S Narayanan. Solution of fokker-planck equation by finite element and finite difference methods for nonlinear systems. *Sadhana*, 31(4):445–461, 2006.

[33] I.E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Trans. Neural Networks*, 9:987–1000, 1998.

[34] Senwei Liang, Shixiao W. Jiang, John Harlim, and Haizhao Yang. Solving pdes on unknown manifolds with machine learning. *arxiv:2106.06682*, 2021.

[35] Shu Liu, Wuchen Li, Hongyuan Zha, and Haomin Zhou. Neural parametric fokker-planck equations. *arXiv preprint arXiv:2002.11309*, 2020.

[36] Ziqi Liu, Wei Cai, and Zhi-Qin John Xu. Multi-scale deep neural network (mscalednn) for solving poisson-boltzmann equation in complex domains. *Communications in Computational Physics*, 28(5):1970–2001, Jun 2020.

[37] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep Network Approximation for Smooth Functions. *arXiv e-prints*, arXiv:2001.03040, 2020.

[38] Jianfeng Lu, Yulong Lu, and Min Wang. A priori generalization analysis of the deep ritz method for solving high dimensional elliptic equations. *arxiv:2101.01708*, 2021.

[39] Luo Lu, Hui Jiang, and Wing H. Wong. Multivariate density estimation by bayesian sequential partitioning. *Journal of the American Statistical Association*, 108(504):1402–1410, 2013.

[40] Yiping Lu, Chao Ma, Yulong Lu, Jianfeng Lu, and Lexing Ying. A mean-field analysis of deep resnet and beyond: Towards provable optimization via overparameterization from depth. *arxiv:2003.05508*, 2020.

[41] Tao Luo and Haizhao Yang. Two-Layer Neural Networks for Partial Differential Equations: Optimization and Generalization Theory. *arXiv e-prints*, arXiv:2006.15733, 2020.

[42] Jonathan C Mattingly, Andrew M Stuart, and Desmond J Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic processes and their applications*, 101(2):185–232, 2002.

[43] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[44] Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics informed neural networks (pinns) for approximating pdes. *arxiv:2006.16144*, 2020.

[45] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[46] H. Montanelli and Q. Du. New error bounds for deep networks using sparse grids. *arXiv e-prints*, arXiv:1712.08688, 2017.

[47] H. Montanelli and H. Yang. Error bounds for deep relu networks using the kolmogorov–arnold superposition theorem. *Neural Networks*, 129:1–6, 2020.

[48] Hadrien Montanelli, Haizhao Yang, and Qiang Du. Deep relu networks overcome the curse of dimensionality for bandlimited functions. *arXiv preprint arXiv:1903.00735*, 2019.

[49] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174):1–38, 2020.

[50] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[51] T. Poggio, H.N. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep—but not shallow—networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14:503–519, 2017.

[52] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 – 707, 2019.

[53] Hannes Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996.

[54] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837, 1956.

[55] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020.

[56] Behnam Sepehrian and Marzieh Karimi Radpoor. Numerical solution of non-linear fokker–planck equation using finite differences method and the cubic spline functions. *Applied mathematics and computation*, 262:187–190, 2015.

[57] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[58] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network with approximation error being reciprocal of width to power of square root of depth. *Neural Computation*, 2021.

[59] Jonathan W. Siegel and Jinchao Xu. Characterization of the variation spaces corresponding to shallow neural networks. *arxiv:2106.15002*, 2021.

[60] Jonathan W. Siegel and Jinchao Xu. Improved approximation properties of dictionaries and applications to neural networks. *arxiv:2101.12365*, 2021.

[61] BF Spencer and LA Bergman. On the numerical solution of the fokker-planck equation for nonlinear stochastic systems. *Nonlinear Dynamics*, 4(4):357–372, 1993.

[62] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[63] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.

[64] Wayne Isaac T Uy and Mircea D Grigoriu. Neural network representation of the probability density function of diffusion processes. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(9):093118, 2020.

[65] Zhipeng Wang and David W Scott. Nonparametric density estimation for high-dimensional data—algorithms and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(4):e1461, 2019.

[66] Yong Xu, Hao Zhang, Yongge Li, Kuang Zhou, Qi Liu, and Jürgen Kurths. Solving fokker-planck equation using deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1):013133, 2020.

[67] Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. *arXiv e-prints*, page arXiv:1906.09477, June 2019.

[68] Z. Song Z. A.-Zhu, Y. Li. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252, Long Beach, California, USA, 2019. PMLR.

[69] Y. Zang, G. Bao, X. Ye, and H. Zhou. Weak adversarial networks for high-dimensional partial differential equations. *J. Comput. Phys.*, 411:109409, 2020.

[70] Jiayu Zhai, Matthew Dobson, and Yao Li. A deep learning method for solving fokker-planck equations. *arXiv preprint arXiv:2012.10696*, 2020.

[71] He Zhang, John Harlim, and Xiantao Li. Estimating linear response statistics using orthogonal polynomials: An rkhs formulation. *Foundations of Data Science*, 2(4):443–485, 2020.

[72] He Zhang, John Harlim, and Xiantao Li. Error bounds of the invariant statistics in machine learning of ergodic itô diffusions. *Physica D (in press), arXiv preprint arXiv:2105.10102*, 2021.

## A Proofs for Section 4

*Proof of Lemma 4.3.* Since $f_0 \in \mathscr{B}_{\mathrm{ReLU}}$, by [13, Theorem 12], there exists a two-layer ReLU FNN $f^*$ with width $W$ such that $\|f^*\|_{L^\infty([0,1]^d)} \le \|f_0\|_{\mathscr{B}_{\mathrm{ReLU}}}$ and

$$\|f^* - f_0\|_{L^\infty([0,1]^d)} \le 4\|f_0\|_{\mathscr{B}_{\mathrm{ReLU}}}(d+1)^{\frac{1}{2}}W^{-\frac{1}{2}} \le 4\sqrt{2}\|f_0\|_{\mathscr{B}_{\mathrm{ReLU}}}d^{\frac{1}{2}}W^{-\frac{1}{2}}.$$

So $f^* \in \mathscr{F}_{2,W,\mathrm{ReLU}}^P$. Since $\nu$ is absolutely continuous with respect to the Lebesgue measure, it follows that

$$\|f^* - f_0\|_{L^2_\nu([0,1]^d)}^2 \le 32\|f_0\|_{\mathscr{B}_{\mathrm{ReLU}}}^2 dW^{-1}. \tag{60}$$

Also, [28, Lemma 3.2] implies that

$$\mathbb{E}_\nu\left[|f_{\mathrm{NN}}(\cdot,\boldsymbol{\theta}^{f_0}) - f_0|^2\right] \le C\left[P^2 W(d+W)\log(Wd+W^2)(\log N)^3 N^{-1} + \inf_{f \in \mathscr{F}_{2,W,\mathrm{ReLU}}^P} \mathbb{E}_\nu\left[|f - f_0|^2\right]\right], \tag{61}$$

where $C$ is a constant that does not depend on $d$, $N$, $W$, $f_0$, $P$. Combining (60) and (61) completes the proof. $\square$

*Proof of Lemma 4.5.* Denote $\hat{e} := q - \hat{p}$. On one hand, using integration by parts,

$$
\begin{aligned}
\int_\Omega \hat{\mathcal{L}}^* \hat{e} \cdot \hat{e}\,\mathrm{d}\boldsymbol{x} &\geq \int_\Omega \sum_{i,j=1}^d \frac{1}{2} B_{\mathrm{NN}}^{ij} \hat{e}_{x_i} \hat{e}_{x_j}\,\mathrm{d}x - \int_{\partial\Omega} \left( \sum_{i,j=1}^d \frac{1}{2} B_{\mathrm{NN}}^{ij} |\hat{e}_{x_i}| \cdot |\mathrm{n}_j| \right) |\hat{e}|\,\mathrm{d}s + \int_\Omega \sum_{i=1}^d a_{\mathrm{NN}}^i \hat{e}_{x_i} \cdot \hat{e} + \left( \sum_{i=1}^d \frac{\partial a_{\mathrm{NN}}^i}{\partial x_i} \right) \hat{e}^2\,\mathrm{d}\boldsymbol{x} \\
&\geq \frac{1}{2} \Lambda \int_\Omega \|\nabla \hat{e}\|^2\,\mathrm{d}x - \frac{1}{2} dB_1 \int_{\partial\Omega} \|\nabla \hat{e}\| \cdot |\hat{e}|\,\mathrm{d}s \\
&\geq \frac{1}{2} \Lambda \|\nabla \hat{e}\|_{L^2(\Omega)}^2 - \frac{1}{2} dB_1 \left( \|\nabla q\|_{L^2(\partial\Omega)} + \|\nabla \hat{p}\|_{L^2(\partial\Omega)} \right) \|\hat{e}\|_{L^2(\partial\Omega)} \\
&\geq \frac{1}{2} \Lambda \|\nabla \hat{e}\|_{L^2(\Omega)}^2 - \frac{1}{2} dB_1 \left( B_2 + \epsilon_{\hat{p}} \right) \|\hat{e}\|_{L^2(\partial\Omega)},
\end{aligned}
\tag{62}
$$

where $\mathrm{n}_j$ is the $j$-th component of the outward unit normal vector.

On the other hand,

$$
\int_\Omega \hat{\mathcal{L}}^* \hat{e} \cdot \hat{e}\,\mathrm{d}x = \int_\Omega \hat{\mathcal{L}}^* q \cdot \hat{e}\,\mathrm{d}x \leq \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)} \cdot \|\hat{e}\|_{L^2(\Omega)}.
\tag{63}
$$

Combining (62) and (63) leads to

$$
\|\nabla \hat{e}\|_{L^2(\Omega)}^2 \leq 2\Lambda^{-1} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)} \cdot \|\hat{e}\|_{L^2(\Omega)} + \Lambda^{-1} dB_1 (B_2 + \epsilon_{\hat{p}}) \|\hat{e}\|_{L^2(\partial\Omega)}.
\tag{64}
$$

Next, by Poincaré inequality, there exists some $C_1 > 0$ that only depends on $\Omega$ such that

$$
\left\| \hat{e} - |\Omega|^{-1} \int_\Omega \hat{e}\,\mathrm{d}x \right\|_{L^2(\Omega)} \leq C_1 \|\nabla \hat{e}\|_{L^2(\Omega)},
$$

which leads to

$$
\|\hat{e}\|_{L^2(\Omega)} \leq |\Omega|^{-1} \left| \int_\Omega \hat{e}\,\mathrm{d}x \right| \|1\|_{L^2(\Omega)} + C_1 \|\nabla \hat{e}\|_{L^2(\Omega)} \leq C_2 \left( \left| \int_\Omega \hat{e}\,\mathrm{d}x \right| + \|\nabla \hat{e}\|_{L^2(\Omega)} \right),
$$

where $C_2 = \max(C_1, |\Omega|^{-1/2})$. Therefore, by (64) and the fact $\int_\Omega \hat{p}\,\mathrm{d}\boldsymbol{x} = 1$,

$$
\begin{aligned}
\|\hat{e}\|_{L^2(\Omega)}^2 &\leq C_3 \left[ \left| \int_\Omega \hat{e}\,\mathrm{d}\boldsymbol{x} \right|^2 + \|\nabla \hat{e}\|_{L^2(\Omega)}^2 \right] \\
&\leq C_3 \left[ \left| \int_\Omega q\,\mathrm{d}x - 1 \right|^2 + 2\Lambda^{-1} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)} \cdot \|\hat{e}\|_{L^2(\Omega)} + \Lambda^{-1} dB_1 (B_2 + \epsilon_{\hat{p}}) \|\hat{e}\|_{L^2(\partial\Omega)} \right],
\end{aligned}
\tag{65}
$$

where $C_3 = 2C_2^2$. Using the Young's inequality $2C_3 \Lambda^{-1} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)} \cdot \|\hat{e}\|_{L^2(\Omega)} \leq \frac{4C_3^2 \Lambda^{-2} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)}^2 + \|\hat{e}\|_{L^2(\Omega)}^2}{2}$, it follows from (65) that

$$
\frac{1}{2} \|\hat{e}\|_{L^2(\Omega)}^2 \leq C_3 \left| \int_\Omega q\,\mathrm{d}x - 1 \right|^2 + 2C_3^2 \Lambda^{-2} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)}^2 + C_3 \Lambda^{-1} dB_1 (B_2 + \epsilon_{\hat{p}}) \|\hat{e}\|_{L^2(\partial\Omega)}.
\tag{66}
$$

Note $\|\hat{e}\|_{L^2(\partial\Omega)} \leq \|\hat{p}\|_{L^2(\partial\Omega)} + \|q\|_{L^2(\partial\Omega)} \leq \epsilon_{\hat{p}} + \|q\|_{L^2(\partial\Omega)}$, it follows from (66) that

$$
\begin{aligned}
\|\hat{e}\|_{L^2(\Omega)}^2 &\leq 2C_3 \left| \int_\Omega q\,\mathrm{d}x - 1 \right|^2 + 4C_3^2 \Lambda^{-2} \|\hat{\mathcal{L}}^* q\|_{L^2(\Omega)}^2 + 2C_3 \Lambda^{-1} dB_1 (B_2 + \epsilon_{\hat{p}}) \epsilon_{\hat{p}} \\
&\quad + 2C_3 \Lambda^{-1} dB_1 (B_2 + \epsilon_{\hat{p}}) \|q\|_{L^2(\partial\Omega)} \\
&\leq C \left( J[q] + d(B_2 + \epsilon_{\hat{p}}) J[q]^{\frac{1}{2}} + d(B_2 + \epsilon_{\hat{p}}) \epsilon_{\hat{p}} \right),
\end{aligned}
$$

where $C$ only depends on $\Omega, \Lambda, B_1, \lambda_1, \lambda_2$. $\qquad\square$

*Proof of Lemma 4.6.* Let $f = \mathbb{E}_{(c,\boldsymbol{w})\sim\rho} [c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x})]$ for some $\rho$ taking the infimum in (37). Then $\hat{\mathcal{L}}^* f = \mathbb{E}_{(c,\boldsymbol{w})\sim\rho} [\hat{\mathcal{L}}^* (c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x}))]$. Using the homogeneity of the neuron $c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x})$, we may assume that $\|\boldsymbol{w}\|_1 = 1$ and $|c| = \|f\|_{\mathscr{B}_{\dot{\sigma}}}$ $\rho$-almost everywhere. Indeed, denote $p_0$ as the density of $\rho$, we define the probability measure $\rho^*$ with the density

$$
p_0^*(\hat{c}, \hat{\boldsymbol{w}}) = \begin{cases} \int_{c\|\boldsymbol{w}\|_1^3 = \hat{c}} p_0(c, \boldsymbol{w})\,\mathrm{d}c\mathrm{d}\boldsymbol{w}, & \text{if } \|\hat{\boldsymbol{w}}\|_1 = 1, \\ 0, & \text{otherwise,} \end{cases}
\tag{67}
$$

then it can be verified that $\rho^* \in P_f$, $\mathbb{E}_\rho |c| \|\boldsymbol{w}\|_1^3 = \mathbb{E}_{\rho^*} |\hat{c}| \|\hat{\boldsymbol{w}}\|_1^3$ and $\text{supp}(p_0^*) \subset \mathbb{R} \times \{\|\hat{\boldsymbol{w}}\|_1 = 1\}$. Moreover, we define the probability measure $\rho^{**}$ with the density

$$
p_0^{**}(\tilde{c}, \tilde{\boldsymbol{w}}) = \begin{cases} \|f\|_{\mathscr{B}_{\dot{\sigma}}}^{-1} \int_0^{+\infty} |\hat{c}| p_0^*(\hat{c}, \hat{\boldsymbol{w}}) \mathrm{d}\hat{c}, & \text{if } \tilde{c} = \|f\|_{\mathscr{B}_{\dot{\sigma}}}, \|\tilde{\boldsymbol{w}}\|_1 = 1, \\ \|f\|_{\mathscr{B}_{\dot{\sigma}}}^{-1} \int_{-\infty}^0 |\hat{c}| p_0^*(\hat{c}, \hat{\boldsymbol{w}}) \mathrm{d}\hat{c}, & \text{if } \tilde{c} = -\|f\|_{\mathscr{B}_{\dot{\sigma}}}, \|\tilde{\boldsymbol{w}}\|_1 = 1, \\ 0, & \text{otherwise,} \end{cases}
\tag{68}
$$

then it can be verified that $\rho^{**} \in P_f$, $\mathbb{E}_{\rho^*} |\hat{c}| \|\hat{\boldsymbol{w}}\|_1^3 = \mathbb{E}_{\rho^{**}} |\tilde{c}| \|\tilde{\boldsymbol{w}}\|_1^3$ and $\text{supp}(p_0^{**}) \subset \{\tilde{c} = \pm\|f\|_{\mathscr{B}_{\dot{\sigma}}}\} \times \{\|\tilde{\boldsymbol{w}}\|_1 = 1\}$.

Let $\{(c_m, \boldsymbol{w}_m)\}$ be $M$ independent and identically distributed samples with $\rho$. By [57, Lemma 26.2],

$$
\mathbb{E}_{\{(c_m, \boldsymbol{w}_m)\} \sim \rho^M} \left[ \sup_{\boldsymbol{x} \in \Omega} \hat{\mathscr{L}}^* \left( \frac{1}{M} \sum_{m=1}^M c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) - \hat{\mathscr{L}}^* f(\boldsymbol{x}) \right]
$$

$$
= \mathbb{E}_{\{(c_m, \boldsymbol{w}_m)\} \sim \rho^M} \left[ \sup_{\boldsymbol{x} \in \Omega} \left( \frac{1}{M} \sum_{m=1}^M \hat{\mathscr{L}}^* c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}) - \mathbb{E}_{(c,\boldsymbol{w}) \sim \rho} [\hat{\mathscr{L}}^* (c\dot{\sigma}(\boldsymbol{w}^\top \boldsymbol{x}))] \right) \right]
$$

$$
\leq 2 \mathbb{E}_{\{(c_m, \boldsymbol{w}_m)\} \sim \rho^M} \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m \hat{\mathscr{L}}^* (c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x})) \right], \tag{69}
$$

where $\tau_m = \pm 1$ with probability $1/2$ are independent Rademacher variables.

Note that

$$
\mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m \hat{\mathscr{L}}^* (c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x})) \right]
$$

$$
= \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m c_m \left( \frac{1}{2} \boldsymbol{w}_m^\top \boldsymbol{B}_{\text{NN}} \boldsymbol{w}_m \dot{\sigma}''(\boldsymbol{w}_m^\top \boldsymbol{x}) + \boldsymbol{a}_{\text{NN}}^\top \boldsymbol{w}_m \dot{\sigma}'(\boldsymbol{w}_m^\top \boldsymbol{x}) + \left( \sum_{i=1}^d \frac{\partial a_{\text{NN}}^i}{\partial x_i} \right) \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) \right]
$$

$$
\leq \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \frac{1}{2} \tau_m c_m \boldsymbol{w}_m^\top \boldsymbol{B}_{\text{NN}} \boldsymbol{w}_m \dot{\sigma}''(\boldsymbol{w}_m^\top \boldsymbol{x}) \right] + \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m c_m \boldsymbol{a}_{\text{NN}}^\top \boldsymbol{w}_m \dot{\sigma}'(\boldsymbol{w}_m^\top \boldsymbol{x}) \right]
$$

$$
+ \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m c_m \left( \sum_{i=1}^d \frac{\partial a_{\text{NN}}^i}{\partial x_i} \right) \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}) \right] \tag{70}
$$

For the first term in (70), by the contraction lemma for Rademacher complexities [57, Lemma 26.9], we have

$$
\mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \frac{1}{2} \tau_m c_m \boldsymbol{w}_m^\top \boldsymbol{B}_{\text{NN}} \boldsymbol{w}_m \dot{\sigma}''(\boldsymbol{w}_m^\top \boldsymbol{x}) \right] = \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m \dot{\sigma}''(\frac{1}{2} c_m \boldsymbol{w}_m^\top \boldsymbol{B}_{\text{NN}} \boldsymbol{w}_m \cdot \boldsymbol{w}_m^\top \boldsymbol{x}) \right]
$$

$$
\leq \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m B_1 |c_m| \|\boldsymbol{w}_m\|_1^2 \cdot \boldsymbol{w}_m^\top \boldsymbol{x} \right] = \frac{B_1}{M} \mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \boldsymbol{x}^\top \sum_{m=1}^M \tau_m |c_m| \|\boldsymbol{w}_m\|_1^2 \cdot \boldsymbol{w}_m \right]
$$

$$
\leq B_1 \mathbb{E}_\tau \left\| \frac{1}{M} \sum_{m=1}^M \tau_m |c_m| \|\boldsymbol{w}_m\|_1^2 \cdot \boldsymbol{w}_m \right\|_1. \tag{71}
$$

Similarly, we can derive

$$
\mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m c_m \boldsymbol{a}_{\text{NN}}^\top \boldsymbol{w}_m \dot{\sigma}'(\boldsymbol{w}_m^\top \boldsymbol{x}) \right] \leq B_1 \mathbb{E}_\tau \left\| \frac{1}{2M} \sum_{m=1}^M \tau_m |c_m| \|\boldsymbol{w}_m\|_1 \cdot \boldsymbol{w}_m \right\|_1 \tag{72}
$$

and

$$
\mathbb{E}_\tau \left[ \sup_{\boldsymbol{x} \in \Omega} \frac{1}{M} \sum_{m=1}^M \tau_m c_m \left( \sum_{i=1}^d \frac{\partial a_{\text{NN}}^i}{\partial x_i} \right) \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}) \right] \leq B_1 \mathbb{E}_\tau \left\| \frac{1}{2M} \sum_{m=1}^M \tau_m |c_m| \boldsymbol{w}_m \right\|_1. \tag{73}
$$

Denote $\boldsymbol{u}_m := c_m \boldsymbol{w}_m$, then $\|\boldsymbol{u}_m\|_1 = \|f\|_{\mathscr{B}_{\dot\sigma}}$. We combine (69)-(73) and obtain

$$
\begin{aligned}
\mathbb{E}_{\{(c_m, \boldsymbol{w}_m)\} \sim \rho^M} \left[ \sup_{\boldsymbol{x} \in \Omega} \hat{\mathscr{L}} \left( \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) - \hat{\mathscr{L}} f(\boldsymbol{x}) \right] & \leq 2 \sup_{\|\boldsymbol{u}_m\|_1 \leq \|f\|_{\mathscr{B}_{\dot\sigma}}} 2 B_1 \mathbb{E}_\tau \| \frac{1}{M} \sum_{m=1}^{M} \tau_m \boldsymbol{u}_m \|_1 \\
& \leq 2 \|f\|_{\mathscr{B}_{\dot\sigma}} \sup_{\|\boldsymbol{u}_m\|_1 \leq 1} 2 B_1 \mathbb{E}_\tau \| \frac{1}{M} \sum_{m=1}^{M} \tau_m \boldsymbol{u}_m \|_1 \\
& \leq 2 \sqrt{d} \|f\|_{\mathscr{B}_{\dot\sigma}} \sup_{\|\boldsymbol{u}_m\|_2 \leq 1} 2 B_1 \mathbb{E}_\tau \| \frac{1}{M} \sum_{m=1}^{M} \tau_m \boldsymbol{u}_m \|_2 \\
& \leq 4 B_1 \|f\|_{\mathscr{B}_{\dot\sigma}} \sqrt{d/M} \qquad (74)
\end{aligned}
$$

by using the Rademacher complexity of the unit ball [57, Lemma 26.10]. Applying the same argument to $-\left( \hat{\mathscr{L}} \left( \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) - \hat{\mathscr{L}} f(\boldsymbol{x}) \right)$ leads to

$$
\mathbb{E}_{\{(c_m, \boldsymbol{w}_m)\} \sim \rho^M} \left[ \sup_{\boldsymbol{x} \in \Omega} \left| \hat{\mathscr{L}} \left( \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) - \hat{\mathscr{L}} f(\boldsymbol{x}) \right| \right] \leq 4 B_1 \|f\|_{\mathscr{B}_{\dot\sigma}} \sqrt{d/M}. \qquad (75)
$$

By similar argument, we can derive

$$
\mathbb{E}_{(c, \boldsymbol{w}) \sim \rho} \left[ \sup_{\boldsymbol{x} \in \Omega} \left| \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) - f(\boldsymbol{x}) \right| \right], \quad \mathbb{E}_{(c, \boldsymbol{w}) \sim \rho} \left[ \sup_{\boldsymbol{x} \in \partial\Omega} \left| \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) - f(\boldsymbol{x}) \right| \right] \leq \|f\|_{\mathscr{B}_{\dot\sigma}} \sqrt{d/M}. \qquad (76)
$$

Therefore we have

$$
\begin{aligned}
\mathbb{E}_{(c, \boldsymbol{w}) \sim \rho^M} \left[ \sup_{\boldsymbol{x} \in \Omega} \left| \hat{\mathscr{L}} \left( \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) \right) - \hat{\mathscr{L}} f(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x} \in \Omega} \left| \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) - f(\boldsymbol{x}) \right| \right. \\
\left. + \sup_{\boldsymbol{x} \in \partial\Omega} \left| \frac{1}{M} \sum_{m=1}^{M} c_m \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) - f(\boldsymbol{x}) \right| \right] \leq (4 B_1 + 2) \|f\|_{\mathscr{B}_{\dot\sigma}} \sqrt{d/M}, \quad (77)
\end{aligned}
$$

which implies there exists $\{(c_m, \boldsymbol{w}_m)\}_{m=1}^{M}$ such that the inequality holds. Then the FNN $\sum_{m=1}^{M} (c_m/M) \dot\sigma(\boldsymbol{w}_m^\top \boldsymbol{x}) \in \mathscr{F}_{2, M, \dot\sigma, \max\{\|f\|_{\mathscr{B}_{\dot\sigma}}/M, 1\}}$ satisfies (38). $\qquad\square$

*Proof of Lemma 4.8.* Denote $\hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) = \hat{p}_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta}^S)$. Since $\hat{p}_{\mathrm{NN}} \in \mathscr{F}_{2, M, \dot\sigma, Q}$, using the expression in (20) we have $\|\nabla \hat{p}_{\mathrm{NN}}^S\|_{L^2(\partial\Omega)} \leq \frac{1}{2} M Q^4 |\partial\Omega|^{\frac{1}{2}} = \frac{1}{2} M Q^4 (2d)^{\frac{1}{2}}$. Then the inequality (39) directly follows Lemma 4.5. For the rest, we use $C$ to represent any constant which on depends on $\Omega$, $\Lambda$, $B_1$, $\lambda_1$ and $\lambda_2$. On one hand,

$$
\begin{aligned}
|J[\hat{p}_{\mathrm{NN}}^S] - J_S[\hat{p}_{\mathrm{NN}}^S]| & \leq \left| \|\mathscr{L} \hat{p}_{\mathrm{NN}}^S\|_{L^2(\Omega)}^2 - \frac{|\Omega|}{N_1} \sum_{n=1}^{N_1} |\mathscr{L} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{I}}^n)|^2 \right| + \lambda_2 \left| \|\hat{p}_{\mathrm{NN}}^S\|_{L^2(\partial\Omega)}^2 - \frac{|\partial\Omega|}{N_3} \sum_{n=1}^{N_3} |\hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{III}}^n)|^2 \right| \\
& + \lambda_1 \left| \int_\Omega \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) \right| \cdot \left| \int_\Omega \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) - 2 \right|. \qquad (78)
\end{aligned}
$$

By virtue of [41, Theorem 3.2], with probability at least $1 - \delta/3$,

$$
\left| \|\mathscr{L} \hat{p}_{\mathrm{NN}}^S\|_{L^2(\Omega)}^2 - \frac{|\Omega|}{N_1} \sum_{n=1}^{N_1} |\mathscr{L} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{I}}^n)|^2 \right| + \lambda_2 \left| \|\hat{p}_{\mathrm{NN}}^S\|_{L^2(\partial\Omega)}^2 - \frac{|\partial\Omega|}{N_3} \sum_{n=1}^{N_3} |\hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{III}}^n)|^2 \right| \leq C I_1. \qquad (79)
$$

Similarly, by the fact $|\hat{p}_{\mathrm{NN}}^S(\boldsymbol{x})| \leq M Q^4/6$ for all $\boldsymbol{x}$ and Lemma 4.7, we have with probability at least $1 - \delta/3$,

$$
\left| \int_\Omega \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) \right| \leq C M Q^4 \sqrt{\log(6/\delta)/N_2}, \qquad (80)
$$

and $\frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) \leq C M Q^4$. Then we have

$$
\lambda_1 \left| \int_\Omega \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) \right| \cdot \left| \int_\Omega \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}_{\mathrm{NN}}^S(\boldsymbol{x}_{\mathrm{II}}^n) - 2 \right| \leq C I_2. \qquad (81)
$$

On the other hand, by Lemma 4.6 there exists some $p_{\mathrm{NN}} \in F_{2,M,\dot{\sigma},Q}$ such that

$$\sup_{\boldsymbol{x} \in \Omega} \left| \hat{\mathscr{L}} p_{\mathrm{NN}}(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x} \in \Omega} \left| p_{\mathrm{NN}}(\boldsymbol{x}) - \hat{p}(\boldsymbol{x}) \right| + \sup_{\boldsymbol{x} \in \partial\Omega} \left| p_{\mathrm{NN}}(\boldsymbol{x}) - \hat{p}(\boldsymbol{x}) \right| \le C \| \hat{p} \|_{\mathscr{B}_{\dot{\sigma}}} \sqrt{d/M}. \tag{82}$$

Note that $\int_{\Omega} \hat{p} \mathrm{d}\boldsymbol{x} = 1$, we have

$$\left| \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{II}}^n) - 1 \right|^2 \le 2 \left( \left| \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \left( p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{II}}^n) - \hat{p}(\boldsymbol{x}_{\mathrm{II}}^n) \right) \right|^2 + \left| \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}(\boldsymbol{x}_{\mathrm{II}}^n) - \int_{\Omega} \hat{p} \mathrm{d}\boldsymbol{x} \right|^2 \right). \tag{83}$$

and

$$\left| p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{III}}^n) \right|^2 \le 2 \left( \left| p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{III}}^n) - \hat{p}(\boldsymbol{x}_{\mathrm{III}}^n) \right|^2 + \epsilon_{\hat{p}}^2 \right), \tag{84}$$

using the fact that $|\hat{p}(\boldsymbol{x})| \le \epsilon_{\hat{p}}$ on $\partial\Omega$ in Assumption 4.5.

Then it follows (82)-(84) and Lemma 4.7 that with probability at least $1 - \delta/3$

$$J_S[\hat{p}_{\mathrm{NN}}^S] \le J_S[p_{\mathrm{NN}}] \quad \le \quad \frac{1}{N_1} \sum_{n=1}^{N_1} \left| \mathscr{L} p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{I}}^n) \right|^2 + 2\lambda_1 \left| \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \left( p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{II}}^n) - \hat{p}(\boldsymbol{x}_{\mathrm{II}}^n) \right) \right|^2 + 2\lambda_1 \left| \frac{|\Omega|}{N_2} \sum_{n=1}^{N_2} \hat{p}(\boldsymbol{x}_{\mathrm{II}}^n) - \int_{\Omega} \hat{p} \right|^2$$

$$+ 2\lambda_2 \frac{|\partial\Omega|}{N_3} \sum_{n=1}^{N_3} \left( \left| p_{\mathrm{NN}}(\boldsymbol{x}_{\mathrm{III}}^n) - \hat{p}(\boldsymbol{x}_{\mathrm{III}}^n) \right|^2 + \epsilon_{\hat{p}}^2 \right) \le C I_3. \tag{85}$$

Finally, the proof can be completed by using (78), (79), (81), (85) and the fact $J[\hat{p}_{\mathrm{NN}}^S] \le \left| J[\hat{p}_{\mathrm{NN}}^S] - J_S[\hat{p}_{\mathrm{NN}}^S] \right| + J_S[\hat{p}_{\mathrm{NN}}^S]$.

<p style="text-align:right">□</p>