SIAM J. NUMER. ANAL.
Vol. 61, No. 1, pp. 45-82

STATIONARY DENSITY ESTIMATION OF ITÔ DIFFUSIONS USING DEEP LEARNING[*]

YIQI GU[†], JOHN HARLIM[‡], SENWEI LIANG[§], AND HAIZHAO YANG[¶]

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 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–Planck 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 PDE regression solution obtained with neural-network models. We demonstrate the effectiveness of this method by numerical simulations of a two-dimensional Student t-distribution and a 20-dimensional Langevin dynamics.

Key words. stochastic differential equations, data-driven method, deep neural network, Fokker–Planck equation

MSC codes. 68T05, 68T07, 37N30, 65N12

DOI. 10.1137/21M1445363

1. Introduction.

Many phenomena subject to random perturbations can be modeled by stochastic differential equations (SDEs) driven by Brownian noises. Under some regularity assumptions, the time evolution of the probability measure can be characterized by the Fokker–Planck equation, a parabolic PDE that governs the time evolution of the density function of the underlying stochastic processes. Despite its wide applications in modeling physical or biological systems [6, 16, 21, 26, 54], 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

[*] Received by the editors September 13, 2021; accepted for publication (in revised form) September 9, 2022; published electronically February 3, 2023.
https://doi.org/10.1137/21M1445363

Funding: The work of the second author was partially supported under the NSF grants DMS-1854299, DMS-2207328, and DMS-2229435 and the Office of Naval Research grant N00014-22-1-2193. The work of the fourth author was partially supported by the US National Science Foundation under award DMS-2244988 and DMS-2206333 and the Office of Naval Research under award N00014-23-1-2007.

[†] Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Queenstown, 119076 Singapore (yiqigu1989@gmail.com).

[‡] 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).

[§] Corresponding author. Department of Mathematics, Purdue University, West Lafayette, IN 47907 USA (liang339@purdue.edu).

[¶] Department of Mathematics, University of Maryland, College Park, MD 20742 USA (hzyang@umd.edu).

from a discrete-time series of approximate solutions of the underlying SDEs without knowing the explicit drift and diffusion components.

Density estimation is a longstanding problem in computational statistics and machine learning. Among the existing approaches, it is widely accepted that the classical kernel density estimation [55] is not effective for problems with a dimension higher than three (see, e.g., [25, 39, 66]). Along this line, the kernel embedding (another class of linear estimator) also suffered from the curse of dimension [73]. Another class of popular parametric density estimators is the Gaussian mixture models (which is also known as the radial basis models in some literature) [25]. 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 nonconvex 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 [15, 24, 38, 47, 48, 49, 52, 59, 69]. Besides, it is shown that, with overparametrization and random initialization, the DNN-based least squares optimization achieves a global minimizer by gradient descent with a linear convergence rate in both the setting of regression [1, 8, 10, 11, 13, 28, 41, 44] and PDE solvers [35, 42]. 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 [51].

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 [33, 57, 61], can be employed to solve the Fokker–Planck equation, they are usually limited to low-dimensional problems. On the other hand, neural-network–based methods have been successfully used in solving high-dimensional PDEs [17, 19, 30, 31, 37, 53, 71, 72], including the recent application in solving the high-dimensional Fokker–Planck equation [36, 68, 72]. 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/depth 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 [74] 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,

$$dX_t = \boldsymbol{a}(X_t)\,dt + \boldsymbol{b}(X_t)\,dW_t, \tag{2.1}$$

with an initial condition randomly drawn from an arbitrary well-defined distribution, $X_0 \sim \pi_0$. The SDE in (2.1) 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 (2.1) 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 π as $t \to \infty$. When the probability measure π 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,

$$\mathcal{L}^* p := -\mathrm{div}(\boldsymbol{a}p) + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}((\boldsymbol{b}\boldsymbol{b}^\top)_{ij}p) = 0, \tag{2.2}$$

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 (2.1) 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 (2.1) that is assumed to possess an ergodic invariant measure, $\tilde{\pi}$, whose "distance" from π can be controlled by the numerical discretization time step δt. 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 (2.1) is geometrically ergodic, then the Markov chain $\{\boldsymbol{x}^n\}$ induced by the Euler–Maruyama (EM) discretization is also geometrically ergodic [43]. 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 (2.1) with a stochastic backward Euler discretization [43], 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,

$$\boldsymbol{x}^{n+1} - \boldsymbol{x}^n = \boldsymbol{a}(\boldsymbol{x}^n)\delta t + \boldsymbol{b}(\boldsymbol{x}^n)\sqrt{\delta t}\boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I}_m), \tag{2.3}$$

where δt 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 [64, 68, 72], we consider a simply compact hypercube $\Omega \subset \mathbb{R}^d$ large enough such that the density on $\mathbb{R}^d \backslash \Omega$ 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 Ω, 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 (2.1) 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 (2.1), we fit the drift $\boldsymbol{a}$ and diffusion coefficients $\boldsymbol{b}\boldsymbol{b}^\top$ in the SDE (2.1) by neural networks (NNs), denoted as $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$, respectively (see section 3.2). Define $\hat{\mathcal{L}}^*$ as the Fokker–Planck (FP) differential operator corresponding to the estimated networks $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$ that approximates the underlying (FP) operator $\mathcal{L}^*$ in (2.2). Our approach in estimating the stationary density p is to solve the homogeneous PDE $\hat{\mathcal{L}}^* \hat{p} = 0$, where $\hat{p}$ is a solution parameterized by an FNN. The PDE can be solved via the network-based least squares 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 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 $\boldsymbol{c} \in \mathbb{R}^{w_L \times 1}$; $\boldsymbol{h}_\ell(\boldsymbol{x}_\ell) := \sigma(\boldsymbol{W}_\ell \boldsymbol{x}_\ell + \boldsymbol{g}_\ell)$ 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 σ is applied entrywise to a vector $\boldsymbol{x}$ to obtain another vector of the same size. w_ℓ is the width of the ℓth 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 ϕ_{NN} to determine the underlying neural network.

Besides FNN, in our numerical simulations, we will also consider the ResNet [20]. Using similar notations above, ResNet can be defined recursively as follows:

$$\begin{aligned}
\boldsymbol{h}_0 &= x, \boldsymbol{h}_{-1} = \boldsymbol{0}, \\
\mathbf{v}_\ell &= \sigma\left(\boldsymbol{W}_\ell \boldsymbol{h}_{\ell-1} + \boldsymbol{g}_\ell\right), \quad \ell = 1, 2, \ldots, L, \\
\boldsymbol{h}_\ell &= \mathrm{pad}(\boldsymbol{h}_{\ell-2}) + \mathbf{v}_\ell, \quad \ell = 1, 2, \ldots, L, \\
\phi_{\mathrm{NN}}(x; \boldsymbol{\theta}) &= \boldsymbol{c}^\top \boldsymbol{h}_L.
\end{aligned} \tag{3.1}$$

Here, the function pad$(\cdot)$ is used to pad zeros to the vector such that two vectors in the summation (3.1) are of the 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 $\sigma(x)=x\mathrm{Tanh}(\log(1+e^x))$ [46]. We use $\mathcal{F}_{L,W,\sigma}$ to denote the class of FNNs with depth L, width W for all layers and activation σ.

3.2. Regression of drift and diffusion coefficients. Taking the expectation of (2.3) with respect to $\boldsymbol{\xi}_n$, one can see that

$$\mathbb{E}[\boldsymbol{x}^{n+1}-\boldsymbol{x}^n-\boldsymbol{a}(\boldsymbol{x}^n)\delta t]=0. \tag{3.2}$$

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 (3.2), we define $\boldsymbol{\theta}_i^{\mathrm{a}}$ as follows:

$$\boldsymbol{\theta}_i^{\mathrm{a}}:=\arg\min_{\boldsymbol{\theta}}\frac{1}{N}\sum_{n=0}^{N-1}\left|y_i^n-a_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta})\right|^2 \tag{3.3}$$

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

$$\boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^{\mathrm{a}}):=[a_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}_1^{\mathrm{a}}),\dots,a_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}_d^{\mathrm{a}})]^\top \tag{3.4}$$

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 labeled training datasets, $\{\boldsymbol{x}^n,\boldsymbol{y}^n\}_{n=0}^{N-1}$. To simplify the analysis in the next section, we assume that $\boldsymbol{x}^i$ is independent and identically distributed (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 subsampling from the Markov chain $\{\boldsymbol{x}^n\}_{n\geq 0}$ such that their temporal correlation is negligible. For convenience of the following discussion, we denote $\mathcal{X}:=\{\boldsymbol{x}^0,\dots,\boldsymbol{x}^{N-1}\}$ and $\mathcal{Y}:=\{\boldsymbol{y}^0,\dots,\boldsymbol{y}^{N-1}\}$. In (3.3), the parameter $\boldsymbol{\theta}_i^{\mathrm{a}}$ is a global minimizer of the empirical loss function. Practically, since stochastic gradient descent or the Adam method [32] 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 that $\mathbb{E}[\boldsymbol{\xi}_n\boldsymbol{\xi}_n^\top]=\boldsymbol{I}_n$ and (2.3) we have

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

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

$$\boldsymbol{\theta}_{ij}^{\mathrm{b}}:=\arg\min_{\boldsymbol{\theta}}\frac{1}{N}\sum_{n=0}^{N-1}\left|(y_i^n-a_{\mathrm{NN}}(\boldsymbol{x}^n,\boldsymbol{\theta}_i^{\mathrm{a}}))(y_j^n-a_{\mathrm{NN}}(\boldsymbol{x}^n,\boldsymbol{\theta}_j^{\mathrm{a}}))-\frac{1}{\delta t}B_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta})\right|^2 \tag{3.5}$$

for $1\leq i,j\leq d$. 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 $\boldsymbol{B}_{\mathrm{NN}}(\boldsymbol{x}):=[B_{\mathrm{NN}}(\boldsymbol{x},\boldsymbol{\theta}_{ij}^{\mathrm{b}})]_{i,j=1,\dots,d}\approx\boldsymbol{b}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})^\top$.

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 (3.5) by deep learning. In such a case, $\boldsymbol{B}_{\rm NN}$ is specified as a matrix, and we will empirically estimate $\boldsymbol{b}\boldsymbol{b}^\top$ using the residual from the drift estimator $\boldsymbol{a}_{\rm NN}(\cdot)$. Particularly,

$$\boldsymbol{B}_{\rm NN} := \frac{\delta t}{N}\sum_{n=1}^{N}\left(\boldsymbol{y}^n - \boldsymbol{a}_{\rm NN}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\rm a})\right)\left(\boldsymbol{y}^n - \boldsymbol{a}_{\rm NN}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\rm a})\right)^\top, \tag{3.6}$$

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

3.3. Estimation of the stationary density. Given the approximate drift $\boldsymbol{a}_{\rm NN}\approx\boldsymbol{a}$ and diffusion coefficient, $\boldsymbol{B}_{\rm NN}\approx\boldsymbol{b}\boldsymbol{b}^\top$, we define the estimated FP operator,

$$\hat{\mathcal{L}}^*p := -\mathrm{div}(\boldsymbol{a}_{\rm NN}p) + \frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}(B_{\rm NN}^{ij}p), \tag{3.7}$$

where $B_{\rm NN}^{ij}$ is the (i,j)-entry of $B_{\rm NN}$. Subsequently, the stationary density is estimated by solving the approximate stationary FP equation,

$$\hat{\mathcal{L}}^*\hat{p} = 0 \quad \text{in } \Omega, \tag{3.8}$$

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

$$\int_\Omega \hat{p}(\boldsymbol{x})d\boldsymbol{x} = 1. \tag{3.9}$$

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

We solve (3.8) with the condition (3.9) by the popular network-based least squares method [12, 34]. Specifically, we use a neural network $\hat{p}_{\rm NN}(\boldsymbol{x};\boldsymbol{\theta})$ with a parameter set $\boldsymbol{\theta}$ determined by solving the following minimization problem:

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

where

$$J[q] := \|\hat{\mathcal{L}}^*q\|^2_{L^2(\Omega)} + \lambda_1\left|\int_\Omega q(\boldsymbol{x})d\boldsymbol{x} - 1\right|^2 + \lambda_2\|q\|^2_{L^2(\partial\Omega)} \quad \forall q:\Omega\to\mathbb{R}. \tag{3.10}$$

Here, λ_1 is a regularization constant corresponding to the normalization factor in (3.9) to ensure a nontrivial solution; λ_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 (3.10) is usually computed via a Monte Carlo integration. For example, if the data $\{\boldsymbol{x}_{\rm I}^n\}_{n=1}^{N_1}$ are uniformly distributed points in Ω, then

$$\|\hat{\mathcal{L}}^*q\|^2_{L^2(\Omega)} \approx \frac{|\Omega|}{N_1}\sum_{n=1}^{N_1}\left|\hat{\mathcal{L}}^*q(\boldsymbol{x}_{\rm I}^n)\right|^2, \tag{3.11}$$

where $|\Omega|$ denotes the volume of the domain Ω. Similarly, as for the second term in (3.10), a Monte Carlo integral is formulated as

$$\int_{\Omega} q(\boldsymbol{x})d\boldsymbol{x} \approx \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} q(\boldsymbol{x}_{\mathrm{II}}^n), \tag{3.12}$$

where $\{\boldsymbol{x}_{\mathrm{II}}^n\}_{n=1}^{N_2}$ are uniformly distributed sampled points in Ω. For the third term in (3.10), we approximate,

$$\|q\|_{L^2(\partial\Omega)} \approx \frac{|\partial\Omega|}{N_3}\sum_{n=1}^{N_3} |q(\boldsymbol{x}_{\mathrm{III}}^n)|^2, \tag{3.13}$$

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

Combining (3.11), (3.12), and (3.13), the training procedure is to minimize the following empirical loss function:

$$J_S[q] := \frac{|\Omega|}{N_1}\sum_{n=1}^{N_1}\left|\hat{\mathcal{L}}^* q(\boldsymbol{x}_{\mathrm{I}}^n)\right|^2 + \lambda_1\left|\frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} q(\boldsymbol{x}_{\mathrm{II}}^n) - 1\right|^2 + \lambda_2\frac{|\partial\Omega|}{N_3}\sum_{n=1}^{N_3}|q(\boldsymbol{x}_{\mathrm{III}}^n)|^2 . \tag{3.14}$$

Let

$$\boldsymbol{\theta}^S = \arg\min_{\boldsymbol{\theta}} J_S[\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})]; \tag{3.15}$$

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, using the regular L^2 norm in the first component of the loss function (3.10) is empirically challenging if d is moderately large. In our method, one needs to intuitively set up the enclosing domain Ω. When d is large, one can either select a complicated domain that covers data points very tightly (which is difficult to implement) or select a standard domain (e.g., a d-dimensional box), in which most of the uniformly sampled points are outside the support of the density. One approach to overcome this issue is using the available time series from the Markov chain in (2.3) directly as the Monte Carlo integration points. Since the time series $\{\boldsymbol{x}^n\}_{n=1}^N$ are distributed in accordance to $\tilde{\pi}$, using them as integration points leads to the first component in (3.10) with a weighted norm, $L^2(\Omega,\tilde{\pi})$. Accordingly, we adjust the Monte Carlo sum in the first component in the empirical loss function in (3.14). This approach is adopted in the numerical examples in section 5.

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 (3.10) 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 (3.10) 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 (3.14). 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 $\mathcal{F}_{L,W,\sigma}$. For the simplicity of analysis, we choose special classes of FNNs as the hypothesis spaces of the optimization. Note the FP operator $\hat{\mathcal{L}}^* p$ involves the first derivative of $\boldsymbol{a}_{\mathrm{NN}}$ and the second derivative of p; it suffices to ensure the regularity that $\boldsymbol{a}_{\mathrm{NN}} \in C^1(\Omega)$ and $\hat{p}_{\mathrm{NN}} \in C^2(\Omega)$ so that the loss function (3.14) is well defined. In practice, the regularity can be weaker: it suffices to hold almost every where in Ω, because we only do computation on a finite number of sample points.

On the one hand, we consider using deep ReLU FNNs with uniform bounds in the minimization (3.3) for the regression of true drift $\boldsymbol{a}(\boldsymbol{x})$. Specifically, for any $P>0$, we denote

$$\mathcal{F}^P_{L,W,\mathrm{ReLU}} = \{\phi \in \mathcal{F}_{L,W,\mathrm{ReLU}} : \ |\phi(\boldsymbol{x})| \le P \ \forall \boldsymbol{x} \in \Omega\} \tag{4.1}$$

as the class of ReLU FNNs with depth L, width W, and a uniform bound P in Ω. It is clear that all functions in $\mathcal{F}^P_{L,W,\mathrm{ReLU}}$ are C^1 smooth almost everywhere in Ω.

On the other hand, we consider using two-layer ReLU^3 FNNs with parameter bounds in the minimization (3.15) for the approximation of the true density $p(\boldsymbol{x})$. More precisely, for any $Q>0$, we explicitly specify

$$\mathcal{F}_{2,M,\dot{\sigma},Q} = \left\{ \phi : \Omega \to \mathbb{R}, \ \phi(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} c_m \dot{\sigma}(\boldsymbol{w}_m^\top \boldsymbol{x}), |c_m|, \|\boldsymbol{w}_m\|_1 \le Q \right\}, \tag{4.2}$$

where M is the width and $\dot{\sigma} = \max(0, x^3/6)$ denotes the ReLU^3 activation function widely used in network-based methods for second-order PDEs. It is clear that $\mathcal{F}_{2,M,\dot{\sigma},Q}$ are C^2 smooth in Ω. 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 [74], 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 and 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 (2.1) given an initial condition. The global Lipschitz assumption also implies the existence of 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 π. See, e.g., Assumptions 2.2–2.3 in [74] 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

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

where $\mathcal{L}$ is the $L^2(\mathbb{R}^d)$ adjoint of the FP operator $\mathcal{L}^*$ defined in (2.2).

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

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

Together with the previous two assumptions, there exists $\delta_0 > 0$ such that, $\forall \delta t \in (0, \delta_0)$, the discrete Markov chain induced by the EM algorithm in (2.3) is geometrically ergodic with the invariant measure, $\tilde{\pi}$, and that

$$\sup_{f \in \mathcal{G}_\ell} |\pi(f) - \tilde{\pi}(f)| \leq K_1 (\delta t)^\nu \pi(V)$$

for some $K_1 = K_1(\ell)$ and $\nu \in (0, 1/2)$. In the equation above, $\pi(f) := \int_{\mathbb{R}^d} f(\boldsymbol{x}) \pi(d\boldsymbol{x})$ and $\tilde{\pi}(f) := \int_{\mathbb{R}^d} f(\boldsymbol{x}) \tilde{\pi}(d\boldsymbol{x})$ denote the expectation of f under the invariant measures π and $\tilde{\pi}$, respectively. Also, the supremum is defined over a set of locally Lipschitz functions bounded above by V,

$$\begin{aligned} \mathcal{G}_\ell := \big\{ f(\boldsymbol{x}) \leq V(\boldsymbol{x}) \forall \boldsymbol{x} \in \mathbb{R}^d \text{ and } \big| \, & |f(\boldsymbol{x}) - f(\boldsymbol{y})| \\ & \leq C_\ell \left(1 + \|\boldsymbol{x}\|^{2\ell - 1} + \|\boldsymbol{y}\|^{2\ell-1}\right) \|\boldsymbol{x} - \boldsymbol{y}\| \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d \big\}. \end{aligned} \tag{4.3}$$

Lemma 4.1. *Under Assumption* 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* ϵ *and is a consistent estimator in the following sense:*

$$\|\boldsymbol{a}(\boldsymbol{x}) - \hat{\boldsymbol{a}}(\boldsymbol{x})\|^2 \leq K_2 (1 + \|\boldsymbol{x}\|^2) \epsilon^2 \quad \forall \boldsymbol{x} \in \mathbb{R}^d, \tag{4.4}$$

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

$$d\hat{X}_t = \hat{\boldsymbol{a}}(\hat{X}_t)\, dt + \hat{\boldsymbol{b}}\, dW_t, \quad \hat{X}_0 = \boldsymbol{x}, \tag{4.5}$$

with $\hat{\boldsymbol{b}}\hat{\boldsymbol{b}}^\top := \hat{\boldsymbol{B}}$ *defined as*

$$\hat{\boldsymbol{B}} := \frac{\delta t}{N} \sum_{n=1}^{N} \left(\boldsymbol{y}^n - \hat{\boldsymbol{a}}(\boldsymbol{x}^n)\right) \left(\boldsymbol{y}^n - \hat{\boldsymbol{a}}(\boldsymbol{x}^n)\right)^\top.$$

For any $\boldsymbol{x} \in \mathbb{R}^d$*, there exist* $0 < \rho < 1$ *and* $K_3 > 0$ *such that*

$$\sup_{f \in \mathcal{G}_\ell} |\pi(f) - \mathbb{E}^{\boldsymbol{x}}[f(\hat{X}_n)]| \leq K_3 \left[\left(\rho^n + \frac{1 - \rho^n}{1 - \rho} \epsilon \right) V(\boldsymbol{x}) \right] \quad \forall n \geq 0, \tag{4.6}$$

where the set $\mathcal{G}_\ell$ *is defined in* (4.3)*. If the process* $\hat{X}_t$ *associated to* (4.5) *has an invariant measure* $\hat{\pi}$*, then there exist* $0 < \alpha < 1$*,* $0 < \beta < \infty$*, and* $0 < \gamma < 1 - \alpha$ *such that* $\hat{\pi}(V) \leq \frac{\beta}{1 - \alpha - \gamma}$*, where* $\hat{\pi}(f) = \int_{\mathbb{R}^d} f(\boldsymbol{x}) \hat{\pi}(d\boldsymbol{x})$*.*

The result above holds $\forall \boldsymbol{x} \in \mathbb{R}^d$ by requiring the condition in (4.4) and that underlying process X_t is ergodic in $\mathbb{R}^d$ with a unique invariant measure π. A similar conclusion was reported in [23] under a much stronger uniform convergence in place of (4.4). One of the key issues in applying this result directly to the learning configuration is that the assumption in (4.4) can be difficult to achieve unless one considers learning with a loss function defined with the topology that is used to deduce the error bound in (4.6), 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 (4.6)), one can at best expect to construct an estimator with convergence guaranteed under an $L^2(\Omega, \tilde{\pi})$ error on a compact domain $\Omega \supset \mathcal{X}$ that contains all the training data. In the numerical section, we will empirically show 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. Define X as a random variable corresponding to the invariant measure π. Let $\Omega \subset \mathbb{R}^d$ be a simply connected compact domain such that $P(X \notin \Omega) \leq \epsilon_0$ for some $0 < \epsilon_0 \ll 1$. For example, let $\Omega := B(0,R) = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq R\}$ be a closed Euclidean ball of radius $R > 1$, and suppose that X has mean zero (centered) and is a subexponentially distributed random variable, $SE(\nu^2, \alpha)$, with $\nu, \alpha > 0$; then by concentration inequality for subexponential distribution, one obtains

$$\mathbb{P}(\|X\| \geq R) \leq 2e^{-\frac{R}{2\alpha}} \quad \forall R > \nu^2\alpha^{-1}. \tag{4.7}$$

Let $\tilde{X}$ be a random variable corresponding to the stationary distribution induced by the EM discretization in (2.3). Using the Markov inequality and strong error bound of EM scheme, one can deduce that $\mathbb{P}[\|X - \tilde{X}\| \geq (\delta t)^{1/4}] \leq (\delta t)^{-1/4}\mathbb{E}[\|X - \tilde{X}\|] \leq C(\delta t)^{1/4}$, which means that

$$\begin{aligned}
&\mathbb{P}\left[\|\tilde{X}\| \geq R + (\delta t)^{1/4}\right] \\
&\quad \leq \mathbb{P}\left[\left\{\|\tilde{X}\| \geq R + (\delta t)^{1/4}\right\} \cap \left\{\|X\| \geq R\right\}\right] + \mathbb{P}\left[\left\{\|\tilde{X}\| \geq R + (\delta t)^{1/4}\right\} \cap \left\{\|X\| < R\right\}\right] \\
&\quad \leq \mathbb{P}\left[\|X\| \geq R\right] + \mathbb{P}\left[\|X - \tilde{X}\| \geq (\delta t)^{1/4}\right] \leq 2e^{-\frac{R}{2\alpha}} + C(\delta t)^{1/4} := \epsilon_0.
\end{aligned} \tag{4.8}$$

Even if X (resp., $\tilde{X}$) is defined on $\mathbb{R}^d$, one can almost surely realize $\|X\| \leq R$ (resp., $\|\tilde{X}\| \leq R + \delta t^{1/4}$) for large enough $R > 0$. This assumption effectively means that the process X satisfies Assumption 4.2 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 now understood $\pi(f) := \int_\Omega f(\boldsymbol{x})\pi(d\boldsymbol{x})$ in (4.6) as an integral with respect to a computational domain Ω. In the convergence theory below, without loss of generality, we will assume that $\Omega = [0,1]^d$. For a general box Ω, similar results can be derived easily by rescaling Ω to $[0,1]^d$ with an isomorphic map. In this case, the concentration inequality (4.7) is still valid for $\|\cdot\|_\infty$ norm since $\|\boldsymbol{x}\|_\infty \geq \|\boldsymbol{x}\|/\sqrt{d} \geq R/\sqrt{d}$ for any $\boldsymbol{x}$ in a box of radius $R/\sqrt{d}$, inscribed in the d-dimensional ball of radius R.

With the above assumption, we only need to restrict our attention to a compact domain Ω, and hence, the assumption that $\hat{\boldsymbol{a}}$ is globally Lipschitz (on a compact hypercube Ω) with Lipschitz constant independent of ϵ can be justified as follows. Particularly, 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 [18, 22], as long as $\boldsymbol{a} \in C^s(\mathbb{R}^d)$ with $s > 1$, there exists a ReLU network $\hat{\boldsymbol{a}}$ approximating $\boldsymbol{a}$ in the Sobolev norm of $W^{1,\infty}(\Omega)$.

For a convex hypercube Ω, this also means that the Lipschitz constant of $\hat{\boldsymbol{a}}$ can be controlled by the Lipschitz constant of $\boldsymbol{a}$ plus a sufficiently large constant. Hence, there exists an estimator $\hat{\boldsymbol{a}}$ such that the Lipschitz constant of $\hat{\boldsymbol{a}}$ can be independent of ϵ. However, how to identify $\hat{\boldsymbol{a}}$ satisfying these assumptions is a problem of the optimization algorithm.

For the rest of this paper, we use the notations $\tilde{\pi}(f) = \int_\Omega f(\boldsymbol{x})\tilde{\pi}(d\boldsymbol{x})$ and $\hat{\pi}(f) = \int_\Omega f(\boldsymbol{x})\hat{\pi}(d\boldsymbol{x})$ for integrals over Ω. 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 (3.8) is defined with the estimated coefficients, namely, $\boldsymbol{a}_{\mathrm{NN}} : \Omega \to \mathbb{R}^d$ as defined in (3.4) and $\boldsymbol{B}_{\mathrm{NN}} \in \mathbb{R}^{d\times d}$ as defined in (3.6), the error analysis below will need to account for the errors induced by these estimations. Recall that $\boldsymbol{b}$ is a constant matrix and that $\boldsymbol{a}_{\mathrm{NN}}$ 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 $\{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n=1}^N$, where $\boldsymbol{x}^n \in \mathcal{X}$ and $\boldsymbol{y}^n := \frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{\delta t} = \boldsymbol{a}(\boldsymbol{x}^n) + \boldsymbol{\eta}^n$, $\boldsymbol{\eta}^n \sim \mathcal{N}(\boldsymbol{0}, (\delta t)^{-1}\boldsymbol{b}\boldsymbol{b}^\top)$. We can now quantify the error of the diffusion estimator in terms of the L^2 error of the drift estimator.

LEMMA 4.2. *Let Assumption* 4.2 *be valid. Define*

$$\epsilon := \delta t\, \mathbb{E}_{\tilde{\pi}}\left[\|(\boldsymbol{a}(X) - \boldsymbol{a}_{\mathrm{NN}}(X; \boldsymbol{\theta}^{\mathrm{a}})\|^2\right] > 0, \tag{4.9}$$

where $\tilde{\pi}$ is the invariant measure corresponding to i.i.d. samples $\{\boldsymbol{x}^n\}_{n=1}^N$ for a fixed $\delta t > 0$. Then there exist some $\beta > 0$ that can depend on the Lipschitz constant of $\boldsymbol{a}_{\mathrm{NN}}$ and δt such that

$$\|\boldsymbol{b}\boldsymbol{b}^\top - \boldsymbol{B}_{\mathrm{NN}}\|_2 \leq 2\epsilon$$

with probability higher than $1 - 2de^{-\frac{\epsilon^2}{2\beta^2 N^{-1} + \frac{4}{3}\beta N^{-1}\epsilon}}$.

We should point out that the i.i.d. assumption is only for the convenience of the theoretical analysis below. While i.i.d. samples can be attained by subsampling from the realization $\{\boldsymbol{z}_n\}_{n\geq 0}$ of a Markov chain generated by the EM scheme in (2.3) to reduce the correlation, we used the correlated samples in our numerical simulations. For the reader's convenience, we quote the following matrix concentration bound that is used for proving the result above.

LEMMA 4.3 (Theorem 1.6.2 in [62] adopted to our notation). *Let $D_1, \ldots, D_N \in \mathbb{R}^{d\times d}$ be independent, symmetric, centered random matrices and $\|D_n\|_2 \leq L$ $\forall n = 1, \ldots, N$. Here, $\|\cdot\|_2$ denotes the spectral norm of a matrix. Let $D = \sum_{n=1}^N D_n$ and $v(D) = \|\mathbb{E}[D^2]\|_2$. Then for any $\epsilon > 0$,*

$$\mathbb{P}[\|D\|_2 \geq \epsilon] \leq 2d \exp\left(-\frac{\epsilon^2}{2v(D) + \frac{2}{3}L\epsilon}\right).$$

Proof of Lemma 4.2. To quantify the error of the diffusion estimator, one can subtract $\boldsymbol{b}\boldsymbol{b}^\top$ from the empirical estimator defined in (3.6) and derive the following upper bound:

$$\|\boldsymbol{b}\boldsymbol{b}^\top - \boldsymbol{B}_{\mathrm{NN}}\|_2 \leq \left\|\sum_{n=1}^N D_n\right\|_2 + \delta t\, \mathbb{E}_{\tilde{\pi}}\left[\|(\boldsymbol{a}(X) - \boldsymbol{a}_{\mathrm{NN}}(X; \boldsymbol{\theta}^{\mathrm{a}}))\|^2\right], \tag{4.10}$$

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

$$
\begin{aligned}
(4.11)\qquad D_n := {}& \frac{\delta t}{N}\big(\boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}})\big)\big(\boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}})\big)^\top \\
&- \frac{1}{N}\Big(\delta t \mathbb{E}_{\tilde{\pi}}[(\boldsymbol{a}(X) - \boldsymbol{a}_{\mathrm{NN}}(X;\boldsymbol{\theta}^{\mathrm{a}}))(\boldsymbol{a}(X) - \boldsymbol{a}_{\mathrm{NN}}(X;\boldsymbol{\theta}^{\mathrm{a}}))^\top] + \boldsymbol{b}\boldsymbol{b}^\top\Big)
\end{aligned}
$$

is an independent, random, symmetric matrix of mean zero. To simplify the notation, we define

$$\boldsymbol{z}_n := (\delta t)^{1/2}(\boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}}))$$

such that we can rewrite

$$D_n = \frac{1}{N}(\boldsymbol{z}_n\boldsymbol{z}_n^\top - A),$$

where $A = \mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top]$ with expectation taken jointly with respect to $\tilde{\pi}$ and the Gaussian random variable $\boldsymbol{\eta}$. By Assumption 4.2, $\boldsymbol{x}^n$ is bounded almost surely on $\Omega = B(0,R)$. Since both $\boldsymbol{a}$ and $\boldsymbol{a}_{\mathrm{NN}}$ are Lipschitz, we have

$$
\begin{aligned}
\|\boldsymbol{z}_n\| &= \|(\delta t)^{1/2}(\boldsymbol{y}^n - \boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}}))\| \\
&\le \left\| \frac{\boldsymbol{x}^{n+1} - \boldsymbol{x}^n}{(\delta t)^{1/2}} - (\delta t)^{1/2}\boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}}))\right\| \\
&\le \Big((\delta t)^{-1/2}\Big(\|\boldsymbol{x}^{n+1}\| + \|\boldsymbol{x}^n\|\Big) + (\delta t)^{1/2}\|\boldsymbol{a}_{\mathrm{NN}}(\boldsymbol{x}^n;\boldsymbol{\theta}^{\mathrm{a}})\|\Big) \le \sqrt{\beta}
\end{aligned}
$$

for some $\beta > 0$ that depends on the Lipschitz constant of $\boldsymbol{a}_{\mathrm{NN}}$ and δt. Therefore,

$$\|D_n\|_2 = \frac{1}{N}\|\boldsymbol{z}_n\boldsymbol{z}_n^\top - A\|_2 \le \frac{1}{N}(\|\boldsymbol{z}_n\boldsymbol{z}_n^\top\|_2 + \|A\|_2) \le \frac{2\beta}{N},$$

where the first term is immediate from the boundedness of $\boldsymbol{z}_i$ and the second term follows from

$$\|A\|_2 = \|\mathbb{E}(\boldsymbol{z}_n\boldsymbol{z}_n^\top)\|_2 \le \mathbb{E}\|\boldsymbol{z}_n\boldsymbol{z}_n^\top\|_2 \le \beta.$$

Here, we have used the Jensen inequality in the above and the boundedness of $\boldsymbol{z}_i$. Next, we compute

$$
\begin{aligned}
\mathbb{E}[D_n^2] &= \frac{1}{N^2}\mathbb{E}\Big[(\boldsymbol{z}_n\boldsymbol{z}_n^\top - A)^2\Big] = \frac{1}{N^2}\mathbb{E}\Big[\|\boldsymbol{z}_n\|^2\boldsymbol{z}_n\boldsymbol{z}_n^\top - A\boldsymbol{z}_n\boldsymbol{z}_n^\top - \boldsymbol{z}_n\boldsymbol{z}_n^\top A + A^2\Big] \\
(4.12)\qquad &\preceq \frac{1}{N^2}\beta\mathbb{E}[\boldsymbol{z}_n\boldsymbol{z}_n^\top] - A^2 \preceq \frac{\beta}{N^2}A,
\end{aligned}
$$

where $A \preceq B$ means that $B - A$ is positive semidefinite. Since $\{D_n\}$ are i.i.d., by (4.12), we have

$$v(D) = \|\mathbb{E}[D^2]\|_2 = \left\|\sum_{n=1}^N \mathbb{E}[D_n^2]\right\|_2 \le \frac{\beta}{N}\|A\|_2 \le \frac{\beta^2}{N}.$$

With these bounds, the conclusion follows directly from Lemma 4.3. □

This means that with high probability the first term in (4.11) can be bounded by $\epsilon > 0$ with large enough sample size, $N \ge 2\beta^2\epsilon^{-2}\log 2d$. Based on this result, we let ϵ be the generalization error rate as defined in (4.9).

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 (4.5). Suppose that $\hat{X}_t$ is generated with $\hat{\boldsymbol{a}} = \boldsymbol{a}_{\rm NN}$ and that $\hat{\boldsymbol{b}}\hat{\boldsymbol{b}}^\top = \boldsymbol{B}_{\rm NN}$ has an invariant measure $\hat{\pi}$ on Ω. Integrating (4.6) with respect to $\hat{\pi}$, we obtain

$$\left|\pi(f) - \hat{\pi}(f)\right| = \left|\pi(f) - \int_\Omega f(\boldsymbol{x})\hat{\pi}(d\boldsymbol{x})\right| = \left|\pi(f) - \int_\Omega \mathbb{E}^{\boldsymbol{x}}[f(\hat{X}_n)]\hat{\pi}(d\boldsymbol{x})\right| \leq K_3\hat{\pi}(V)\epsilon \tag{4.13}$$

as $n \to \infty$. To obtain (4.13), we have used (4.6). With this background, the error bound for $\hat{p}_{\rm NN}(\boldsymbol{x};\boldsymbol{\theta}^S)$ can be deduced by accounting for the regression error of a and the error from the proposed PDE solver,

$$\begin{aligned}(4.14)\quad &\left|\pi(f) - \int_\Omega f(\boldsymbol{x})\hat{p}_{\rm NN}(\boldsymbol{x};\boldsymbol{\theta}^S)\,d\boldsymbol{x}\right| \\ &\leq \left|\pi(f) - \int_\Omega f(\boldsymbol{x})\hat{p}(\boldsymbol{x})d\boldsymbol{x}\right| + \left|\int_\Omega f(\boldsymbol{x})\big(\hat{p}(\boldsymbol{x}) - \hat{p}_{\rm NN}(\boldsymbol{x};\boldsymbol{\theta}^S)\big)\,d\boldsymbol{x}\right| \\ &\leq K_3\hat{\pi}(V)\delta t\underbrace{\mathbb{E}_{\tilde{\pi}}\left[\|(\boldsymbol{a}(X) - \boldsymbol{a}_{\rm NN}(X;\boldsymbol{\theta}^{\rm a})\|^2\right]}_{(I)} + \|f\|_{L^2(\Omega)}\underbrace{\|\hat{p} - \hat{p}_{\rm NN}(\cdot;\boldsymbol{\theta}^S)\|_{L^2(\Omega)}}_{(II)},\end{aligned}$$

where we have used (4.13), (4.9), and the Cauchy–Schwarz inequality. In the next two subsections, we will bound the terms (I) and (II) in (4.14).

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 (3.3). We will derive the L^2 error with respect to $\tilde{\pi}$ between the estimator $\boldsymbol{a}_{\rm NN}(\boldsymbol{x};\boldsymbol{\theta}^{\rm a})$ and the true drift function $\boldsymbol{a}(\boldsymbol{x})$. For this purpose, given a class $\mathcal{F}$ of functions, $\Omega \to \mathbb{R}$, we denote its pseudodimension by ${\rm Pdim}(\mathcal{F})$, which is the largest integer m for which there is some $(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m,y_1,\ldots,y_m) \in \Omega^m \times \mathbb{R}^m$ such that, for any $(b_1,\ldots,b_m) \in \{0,1\}^m$, there exists $f \in \mathcal{F}$ satisfying $f(\boldsymbol{x}_i) > y_i \Leftrightarrow b_i = 1 \forall i$. Note the pseudodimension is a generalization of the Vapnik–Chervonenkis dimension to real-valued function classes [65]. If the class consists of binary-valued functions, the two concepts are equivalent. Thus the only extra feature of pseudodimension is the possibility of introducing the "off-set" vector $(y_1,\ldots,y_m) \in \mathbb{R}^m$. Based on [5], one can estimate the pseudodimension of the FNN class with ${\rm Pdim}(\mathcal{F}_{L,W,\sigma}) = O(L^2W^2\log(LW^2))$ if σ is a piecewise linear activation function. For sigmoid activation functions, an upper bound of the pseudodimension can be found in [2].

The prediction error analysis of FNNs has been studied in several papers, e.g., [7, 14, 29, 40, 42, 45, 50, 56]. In particular, we introduce the following lemma concerning the prediction error of the FNN-based least squares regression, which is studied in [29].

LEMMA 4.4 [29, 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 $|f_0(x) - f_0(y)| \leq \lambda\|\boldsymbol{x} - \boldsymbol{y}\|^\alpha$ $\forall \boldsymbol{x},\boldsymbol{y} \in [0,1]^d$. Suppose $\|f_0\|_{L^\infty([0,1]^d)} \leq P$ for some $P \geq 1$. Let ν be a probability measure that is absolutely continuous with respect to the Lebesgue measure and a random variable $\boldsymbol{x} \sim \nu$. Let η be a random variable with mean 0 and finite variance. Let $\{\boldsymbol{x}^n\}_{n=1}^N$ be N i.i.d. samples of $\boldsymbol{x}$, and $y^n = f_0(\boldsymbol{x}^n) + \eta$ is the response with noise η for each n. For any $I_1, I_2 \in \mathbb{N}^+$, let*

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

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

$$\begin{aligned}(4.15)\qquad \mathbb{E}_\nu&\left[|f_{\mathrm{NN}}(\cdot,\boldsymbol{\theta}^{f_0}) - f_0|^2\right]\\ &\le C\left[P^2WL(d+WL)\log(Wd+W^2L)(\log N)^3N^{-1} + \lambda^2 d(I_1I_2)^{-4\alpha/d}\right]\end{aligned}$$

for $N \ge \mathrm{Pdim}(\mathcal{F}^P_{L,W,\mathrm{ReLU}})$*, where* C *is a constant that does not depend on* d*,* N*,* L*,* W*,* λ*,* α*,* I_1*,* I_2*,* P*.*

In Lemma 4.4, the exponent of the error bound in (4.15) can be improved to be dimension-independent if we assume f_0 is in Barron-type spaces, which are first studied in [4] and further developed in [9, 15, 40, 60, 67]. Here we follow the Barron space with respect to two-layer ReLU networks proposed in [67]. Suppose $f:\Omega\to\mathbb{R}$ is a function of the form

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

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

$$(4.16)\qquad \|f\|_{\mathcal{B}_{\mathrm{ReLU}}} = \inf_{\rho\in P_f}(\mathbb{E}_\rho|c|\|\boldsymbol{w}\|_1),$$

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

LEMMA 4.5. *Let* $f_0 : [0,1]^d \to \mathbb{R}$ *such that* $\|f_0\|_{\mathcal{B}_{\mathrm{ReLU}}} \le P$ *and* $\|f_0\|_{L^\infty([0,1]^d)} \le P$ *for some* $P \ge 1$*. For the least squares regression proposed in Lemma* 4.4*, we let* $f_{\mathrm{NN}} \in \mathcal{F}^P_{2,W,\mathrm{ReLU}}$ *for some* $W\in\mathbb{N}^+$*. Then the prediction error is given by*

$$\begin{aligned}(4.17)\qquad \mathbb{E}_\nu&\left[|f_{\mathrm{NN}}(\cdot,\boldsymbol{\theta}^{f_0}) - f_0|^2\right]\\ &\le C\left[P^2W(d+W)\log(Wd+W^2)(\log N)^3N^{-1} + \|f_0\|^2_{\mathcal{B}_{\mathrm{ReLU}}}dW^{-1}\right]\end{aligned}$$

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

Proof. See Appendix A. ◻

In our case, we set in the hypothesis of Lemma 4.4 that $L = O(I_2)$ and $W = O(I_1)$ are both large integers. Combining with Lemma 4.5, the error estimation for the minimization problem (3.3) can be directly obtained.

LEMMA 4.6. *In addition to 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* $\boldsymbol{a}_{\mathrm{NN}}$ *defined in* (3.4) *with components* $a_{\mathrm{NN}} \in \mathcal{F}^{P_{\boldsymbol{a}}}_{L,W,\mathrm{ReLU}}$ *satisfies*

$$(4.18)\qquad \mathbb{E}_{\tilde{\pi}}\left[|\boldsymbol{a}_{\mathrm{NN}} - \boldsymbol{a}|^2\right] \le C_{\boldsymbol{a}}\left(d^2WLN^{-1} + d(WL)^2N^{-1} + d^2(WL)^{-4/d}\right)$$

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

2. *Suppose that all components of* $\boldsymbol{a}$ *are in* $\mathcal{B}_{\text{ReLU}}$ *with Barron norms no greater than* $P_{\boldsymbol{a}}$*. Let* $W \in \mathbb{N}^+$*; then the estimator* $\boldsymbol{a}_{\text{NN}}$ *defined in* (3.4) *with components* $a_{\text{NN}} \in \mathcal{F}^{P_{\boldsymbol{a}}}_{2,W,\text{ReLU}}$ *satisfies*

$$\mathbb{E}_{\tilde{\pi}}\left[|\boldsymbol{a}_{\text{NN}} - \boldsymbol{a}|^2\right] \le C_{\boldsymbol{a}}\left(d^2WN^{-1} + dW^2N^{-1} + d^2W^{-1}\right) \tag{4.19}$$

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

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 4.6, 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 also depends on the dimension at most quadratically.

4.3. Solution error for the approximate FP equation.

Now let us consider the error between $\hat{p}_{\text{NN}}(\cdot;\boldsymbol{\theta}^S)$ and the true solution $\hat{p}$ of the approximate stationary FP equation (3.8). In this section, we only consider the case that $\{\boldsymbol{x}^n_{\text{MC}}\}_{n=1}^{N_1}$ in (3.12) are uniformly distributed in Ω. Similar results apply to other measures with smooth densities supported on Ω.

We rewrite the approximate stationary FP equation (3.7) in the following divergence form:

$$-\hat{\mathcal{L}}^*\hat{p} = -\sum_{i,j=1}^{d}\left(\frac{1}{2}B^{ij}_{\text{NN}}\hat{p}_{x_j}\right)_{x_i} + \sum_{i=1}^{d} a^i_{\text{NN}}\hat{p}_{x_i} + \left(\sum_{i=1}^{d}\frac{\partial a^i_{\text{NN}}}{\partial x_i}\right)\hat{p} = 0 \quad \text{in } \Omega. \tag{4.20}$$

The error analysis is valid only when (4.20) is well-posed. So we need to set up specific assumptions on the coefficients of (4.20). First, note that $\boldsymbol{B}_{\text{NN}}$ is positive semidefinite and that (4.20) is elliptic, so we assume further that (4.20) is nondegenerate by specifying the smallest eigenvalue of $\boldsymbol{B}_{\text{NN}}$ 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 $\boldsymbol{B}_{\text{NN}}$, denoted as Λ, is positive. Besides, $|B^{ij}_{\text{NN}}| < 2B_1$, $|a^i_{\text{NN}}(\boldsymbol{x})| < B_1$, $|\sum_{i=1}^d \partial a^i_{\text{NN}}(\boldsymbol{x})/\partial x_i| < B_1$, $\forall i, j$ and $\forall \boldsymbol{x} \in \Omega$, for some $B_1 > 0$.

Next, considering (4.20) is defined in a compact domain, we cannot 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.

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

Under Assumption 4.4, one can show that (4.20) with any Dirichlet condition admits a unique solution by the Fredholm alternative and Lax–Milgram theorem. However, we cannot 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 Assumptions 4.4 and 4.5, it can be shown that any two solutions of (4.20) are close to each other with 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 4.7. *Assume $\hat{p}$ is a classical solution of* (3.8) *with the condition* (3.9). *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*

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

where C only depends on Ω, Λ, B_1, B_2, λ_1, λ_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 [42]. 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 of the 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 ρ on $\mathbb{R} \times \mathbb{R}^d$; then its ReLU^3 Barron norm is defined by

$$\|f\|_{\mathcal{B}_{\dot{\sigma}}} = \inf_{\rho \in P_f} (\mathbb{E}_\rho |c| \|\boldsymbol{w}\|_1^3), \tag{4.21}$$

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 $\mathcal{B}_{\dot{\sigma}} = \{f \in C^0 : \|f\|_{\mathcal{B}_{\dot{\sigma}}} < \infty\}$. Now let us derive the uniform approximation of FNNs in $\mathcal{F}_{2,M,\dot{\sigma},Q}$ for Barron functions.

LEMMA 4.8. *Given $f \in \mathcal{B}_{\dot{\sigma}}$, there exist some $p_{\mathrm{NN}} \in \mathcal{F}_{2,M,\dot{\sigma},\max\{\|f\|_{\mathcal{B}_{\dot{\sigma}}}/M,1\}}$ such that*

$$\begin{aligned} \sup_{\boldsymbol{x}\in\Omega} \left|\hat{\mathcal{L}}^* p_{\mathrm{NN}}(\boldsymbol{x}) - \hat{\mathcal{L}}^* f(\boldsymbol{x})\right| + \sup_{\boldsymbol{x}\in\Omega} |p_{\mathrm{NN}}(\boldsymbol{x}) - f(\boldsymbol{x})| + \sup_{\boldsymbol{x}\in\partial\Omega} |p_{\mathrm{NN}}(\boldsymbol{x}) - f(\boldsymbol{x})| \\ \le (4B_1 + 2)\, \|f\|_{\mathcal{B}_{\dot{\sigma}}} \sqrt{d/M} \end{aligned} \tag{4.22}$$

Proof. See Appendix A. ◻

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

Lemma 4.9. *Given a compact domain Ω, 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 Ω. Then for any $\delta\in(0,1)$, with probability at least $1-\delta$ over the choice of $\boldsymbol{x}_n$,*

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

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

Lemma 4.10. *Under Assumptions 4.3–4.5, we further assume $\hat{p}\in\mathcal{B}_{\dot\sigma}$. Let $\boldsymbol{\theta}^S=\operatorname{argmin}_{\boldsymbol{\theta}}J_S[\hat{p}_{\mathrm{NN}}(\cdot,\boldsymbol{\theta})]$ with $\hat{p}_{\mathrm{NN}}\in\mathcal{F}_{2,M,\dot\sigma,Q}$. Also, suppose $\{\boldsymbol{x}_{\mathrm{I}}^n\}_{n=1}^{N_1}\subset\Omega$, $\{\boldsymbol{x}_{\mathrm{II}}^n\}_{n=1}^{N_2}\subset\Omega$, $\{\boldsymbol{x}_{\mathrm{III}}^n\}_{n=1}^{N_3}\subset\partial\Omega$ in (3.14) are uniformly distributed. Then for any $\delta\in(0,1)$, with probability of at least $1-\delta$ over the choice of these points,*

(4.23)

$$\begin{aligned}&\|\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)-\hat{p}\|_{L^2(\Omega)}^2\\&\quad\le C\left(J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)]+d(MQ^4d^{\frac12}+\epsilon_{\hat p})J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)]^{\frac12}+d(MQ^4d^{\frac12}+\epsilon_{\hat p})\epsilon_{\hat p}\right),\end{aligned}$$

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

$$\begin{aligned}I_1&=(Q^8+1)\left(d^2\sqrt{\log(d)}+\log(Q^4+1)+\sqrt{\log(1/\delta)}\right)M^2(1/\sqrt{N_1}+1/\sqrt{N_3}),\\I_2&=MQ^4\sqrt{\log(6/\delta)/N_2}\left(MQ^4(\sqrt{\log(6/\delta)/N_2}+1)+1\right),\\I_3&=\|\hat{p}\|_{\mathcal{B}_{\dot\sigma}}^2d/M+\|\hat{p}\|_\infty^2\log(6/\delta)/N_2+\epsilon_{\hat p}^2,\end{aligned}$$

where C only depends on Ω, Λ, B_1, λ_1, and λ_2. Especially, suppose that $J[\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)]\le 1$ and $N_{\mathrm{p}}:=\min\{N_1,N_2,N_3\}$. Take $Q\le O(M^{-\frac14}d^{-\frac18})$ and $N_{\mathrm{p}}\ge O(\log(1/\delta))$; then

$$\|\hat{p}_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)-\hat{p}\|_{L^2(\Omega)}^2\le O\left(d^2(\log(d))^{\frac14}MN_{\mathrm{p}}^{-\frac14}+d^{\frac32}M^{-\frac12}+dN_{\mathrm{p}}^{-\frac12}+d\epsilon_{\hat p}\right),$$

with an order constant depending on Ω, Λ, B_1, λ_1, λ_2, δ, and $\hat{p}$.

Proof. See Appendix A. □

The result in Lemma 4.10 implies that, when $N_{\mathrm{p}}\sim O(M^s)$ with $s>4$, the error reduces to $O(d\epsilon_{\hat p})$ as $M,N_{\mathrm{p}}\to\infty$. In practice, as an approximation to the original density p that vanishes as $\|\boldsymbol{x}\|\to\infty$, the solution $\hat{p}$ could have a similar decaying behavior as p. Hence $\epsilon_{\hat p}$ is small enough if Ω is moderately large. And this also leads to a small solution error $\|\hat{p}_{\mathrm{NN}}-\hat{p}\|_L^2(\Omega)$.

4.4. The main error estimation. Inserting the two error bounds in Lemmas 4.6 and 4.10 into the inequality in (4.14) and collecting all the assumptions, we can show the following main theorem for the error estimation of the proposed algorithm.

Theorem 4.1. *Let π be the invariant measure of a Markov process X_t that satisfies Assumptions 4.1 and 4.2. Let $P_{\boldsymbol{a}}\ge1$ 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 (3.4) with components $a_{\mathrm{NN}}\in\mathcal{F}_{L,W,\mathrm{ReLU}}^{P_{\boldsymbol{a}}}$ is a consistent estimator in the sense of (4.4) $\forall\boldsymbol{x}\in\Omega=$*

$[0,1]^d$. *Suppose also that* $N \geq \mathrm{Pdim}(\mathcal{F}_{L,W,\mathrm{ReLU}})$. *Let the assumptions in Lemma* 4.10 *be valid, namely, Assumptions* 4.3–4.5. *Suppose that* $\hat{p} \in \mathcal{B}_{\hat{\sigma}}$ *is estimated by* $\hat{p}_{\mathrm{NN}}(\cdot, \boldsymbol{\theta}^S) \in \mathcal{F}_{2,M,\hat{\sigma},Q}$ *with* $Q \leq O(M^{-\frac{1}{4}} d^{-\frac{1}{8}})$, *where* $\boldsymbol{\theta}_S$ *is the global minimizer of the empirical loss function* (3.14). *Then,* $\forall f \in \mathcal{G}_\ell$ *as defined in* (4.3) *and for any* $\delta \in (0,1)$, *with probability of at least* $1-\delta$ *over the choice of* $\{\boldsymbol{x}_{\mathrm{I}}^n\}_{n=1}^{N_1}$, $\{\boldsymbol{x}_{\mathrm{II}}^n\}_{n=1}^{N_2}$, *and* $\{\boldsymbol{x}_{\mathrm{III}}^n\}_{n=1}^{N_3}$,

$$
\begin{aligned}
&\sup_{f \in \mathcal{G}_\ell} \left| \pi(f) - \int_\Omega f(\boldsymbol{x}) \hat{p}_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{\theta}^S)\, d\boldsymbol{x} \right| \\
&\quad \leq K_3 \hat{\pi}(V) \delta t C_{\boldsymbol{a}} \left(d^2 W L N^{-1} + d(WL)^2 N^{-1} + d^2 (WL)^{-4/d} \right) \\
&\qquad + C_{\hat{p}} \left(d^2 (\log(d))^{\frac{1}{4}} M N_{\mathrm{p}}^{-\frac{1}{4}} + d^{\frac{3}{2}} M^{-\frac{1}{2}} + d N_{\mathrm{p}}^{-\frac{1}{2}} + d \epsilon_{\hat{p}} \right) \\
&\qquad \sim O\left(d^2 \left(W^2 L^2 N^{-1} + W^{-4/d} L^{-4/d} + M N_{\mathrm{p}}^{-1/4} + M^{-1/2} + \epsilon_{\hat{p}} \right) \right),
\end{aligned}
\tag{4.24}
$$

where $N_{\mathrm{p}} := \min\{N_1, N_2, N_3\}$ *that satisfies* $N_{\mathrm{p}} \geq O(\log(1/\delta))$. *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* Ω, δ, $\hat{p}$, $\|f\|_{L^2(\Omega)}$, *the regularization weights* λ_1, λ_2, *and the upper bounds constants* Λ, B_1 *defined in Assumption* 4.3.

The error bound that only depends on the dimension quadratically is given as follows.

Theorem 4.2. *Under the hypothesis of Theorem* 4.1, *we further assume that all components of* $\boldsymbol{a}$ *are in* $\mathcal{B}_{\mathrm{ReLU}}$ *with Barron norms no greater than* $P_{\boldsymbol{a}}$, *and let* $a_{\mathrm{NN}} \in \mathcal{F}_{2,W,\mathrm{ReLU}}^{P_{\boldsymbol{a}}}$; *then the error bound term* $d^2 W L N^{-1} + d(WL)^2 N^{-1} + d^2 (WL)^{-4/d}$ *in* (4.24) *can be improved to be* $d^2 W N^{-1} + d W^2 N^{-1} + d^2 W^{-1}$, *and the entire bound is of* $O(d^2(W^2 N^{-1} + W^{-1} + M N_{\mathrm{p}}^{-1/4} + M^{-1/2} + \epsilon_{\hat{p}}))$.

We should point out that, while the results are valid for a global minimizer $\boldsymbol{\theta}_i^{\mathrm{a}}$ in (3.3) and $\boldsymbol{\theta}_S$ in (3.15), we do not specify the condition for which such global minimizers are attainable. We implicitly assume that a minimizer is found and do not consider the error from the optimization algorithms. In practice, one cannot ensure that a global minimizer can be necessarily found by usual optimizers like gradient descent.

Moreover, throughout the convergence analysis, we consider using special FNN class (4.1) with uniform bounds or (4.2) with parameter bounds as the hypothesis space and derive corresponding approximation errors. However, in practical deep learning, one usually uses the general FNN class $\mathcal{F}_{L,W,\sigma}$ 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 Lennard–Jones potential with the Gibbs invariant measure.

In our examples, we directly use the available dataset $\mathcal{X} := \{\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{N-1}\}$ as the Monte Carlo integration points. Hence in the mathematical sense, we replace the norm in the first term in (3.10) with a weighted $L^2(\Omega, \tilde{\pi})$, recalling that $\tilde{\pi}$ denotes the stationary measure of the discrete Markov chain induced by (2.3).

Empirically, we approximate the first term of (3.10) via the following Monte Carlo average:

$$\|\hat{\mathcal{L}}q\|^2_{L^2(\Omega,\tilde{\pi})} \approx \frac{1}{|\mathcal{X}\cap\Omega|}\sum_{n=0}^{N-1}\left|\hat{\mathcal{L}}q(\boldsymbol{x}^n)\right|^2 \mathbb{1}_\Omega(\boldsymbol{x}^n), \tag{5.1}$$

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

5.1. Student's t-distribution.

Consider a two-dimensional SDE (2.1) for Student's t-distribution [3] 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 $\phi(x_1,x_2) = 1 + \frac{2}{15}(4x_1^2 - x_1x_2 + x_2^2)$. Our consideration for testing the proposed method on this system of SDEs is based on the following motivations: Since this system of SDEs has a nontrivial nonconstant diffusion term, it is a reasonable testbed to verify the numerical performance of the proposed approach when a neural-network training that involves solving (3.5) is required. Furthermore, since the stationary density of this system is explicitly given by

$$p(x_1,x_2) = \frac{2}{\pi\sqrt{15}}\left(\phi(x_1,x_2)\right)^{-3}, \tag{5.2}$$

one can validate the accuracy of the numerical estimate.

5.1.1. Data generation and implementation details.

The time series dataset $\{\boldsymbol{x}^i\}_{i=0}^N$ is generated by EM scheme (2.3) with $\delta t = 0.05$ and $N = 2\times 10^7$. The bounded domain Ω is set as $[-4,4]\times[-6,6]$ such that over 98% points are in Ω.

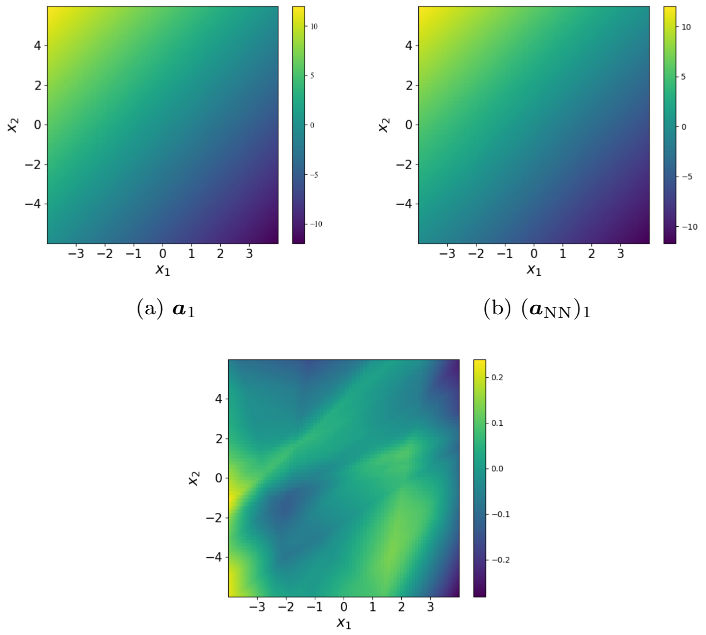
In our implementation, we use 6-hidden-layer ResNets (discussed in section 3.1) with the same width 50 per hidden layer. We employ the networks with ReLU, Mish [46], and ReLU^3 activations to learn $\boldsymbol{a}$, $\boldsymbol{b}\boldsymbol{b}^\top$, and p, respectively. Notice that Mish is C^∞ smooth, so the loss (3.14) is still well defined. To learn $\boldsymbol{a}_{\mathrm{NN}}$ and B_{NN}, the Adam algorithm is applied to optimize the loss (3.3) and (3.5) with batch size 10,000 for $T = 20{,}000$ iterations. We use an initial learning rate of 10^{-4}. The learning rate follows a cosine decay with the increasing training iterations; i.e., the learning rate decays by a multiplication factor $0.5(cos(\frac{\pi t}{T}) + 1)$, where t is the current iteration. To solve the PDE (3.7), we optimize the loss (3.10) with regularization parameters $\lambda_1 = 1, \lambda_2 = 500$. We remark that, although λ_1 and λ_2 can be tuned carefully, we only take any feasible choice because empirical experiences suggest that varying these parameters in a moderate range only changes the result slightly. In Adam, we use the batch size 10,000 for the first term in (3.10) and 4,000 for the boundary term, while the second term is approximated by 300^2 Gaussian quadrature points. The learning rate is initialized at 10^{-3} and follows the cosine decay prescribed above.

5.1.2. Identification of the drift and diffusion coefficients..

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

$$\frac{\|f - \hat{f}\|_{L^2(\Omega)}}{\|f\|_{L^2(\Omega)}}, \tag{5.3}$$

where f and $\hat{f}$ represent the true and approximate functions, respectively. Numerically, we approximate the integral over $10{,}000$ Gaussian quadrature points in Ω.



(a) $\boldsymbol{a}_1$ (b) $(\boldsymbol{a}_{\rm NN})_1$

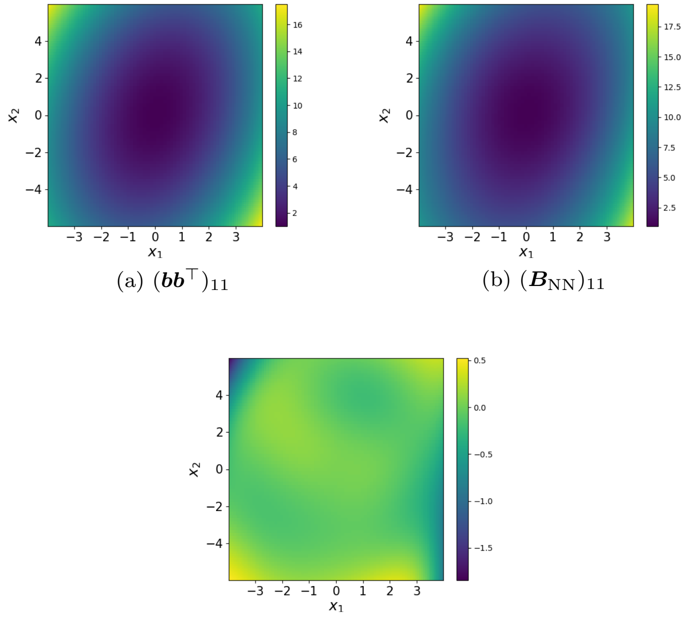
(c) $\boldsymbol{a}_1$-$(\boldsymbol{a}_{\rm NN})_1$

FIG. 5.1. *The comparison of the first component of drift term.* (a) $\boldsymbol{a}_1$, (b) $(\boldsymbol{a}_{NN})_1$, *and* (c) *their difference.*

The relative L_2 error between $\boldsymbol{a}_{\rm NN}$ and $\boldsymbol{a}$ is 5.63×10^{-2}. Figure 5.1 displays the spatial profile of the first components of $\boldsymbol{a}$ and $\boldsymbol{a}_{\rm NN}$ and their difference on the computational domain Ω. The relative L_2 error between $\boldsymbol{B}_{\rm NN}$ and $\boldsymbol{b}\boldsymbol{b}^\top$ is 3.63×10^{-2}. To check the pointwise accuracy of the estimates, we plot the first diagonal components of $\boldsymbol{b}\boldsymbol{b}^\top$, $\boldsymbol{B}_{\rm NN}$ on the computational domain Ω and their difference in Figure 5.2. We can see our method works well on fitting the drift and diffusion terms.

Given the approximate drift and diffusion coefficients, we now empirically validate the result in Lemma 4.1 on the computational domain Ω. Particularly, we want to check whether the Markov chain generated by the corresponding SDE in (4.5) (with $\hat{\boldsymbol{a}} = \boldsymbol{a}_{\rm NN}$ and $\hat{b}\hat{b}^\top = \boldsymbol{B}_{\rm NN}$) can reproduce the stationary mean and covariance statistics of the underlying invariant measure, π. In Table 5.1, we listed the true mean and covariance statistics corresponding to the underlying distribution π and the approximate distribution $\tilde{\pi}$ corresponding to discrete Markov chain generated by EM discretization in (2.3) with the time step $\delta t = 0.05$ that can be empirically estimated using the Monte Carlo average over the discrete samples. To emphasize that these statistics are subjected to EM error, we denote $\tilde{\pi} := \pi^{EM}$. Since the SDE with coefficients $\boldsymbol{a}_{\rm NN}$ and $\boldsymbol{B}_{\rm NN}$ is not analytically solvable, the statistics defined with respect to the corresponding stationary distribution $\hat{\pi}$, whose density solves (3.7), are not computable. To validate the statistical consistency of the approximate SDE, we compute the empirical mean and covariance by averaging over a Markov chain corresponding to the following EM discretization:

$$\boldsymbol{x}^{n+1} - \boldsymbol{x}^n = \boldsymbol{a}_{\rm NN}(\boldsymbol{x}^n)\delta t + \boldsymbol{U}(\boldsymbol{x}^n)\boldsymbol{S}(\boldsymbol{x}^n)^{\frac{1}{2}}\boldsymbol{U}(\boldsymbol{x}^n)^\top\sqrt{\delta t}\boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I}_2), \tag{5.4}$$



(a) $(\boldsymbol{b}\boldsymbol{b}^\top)_{11}$

(b) $(\boldsymbol{B}_{\mathrm{NN}})_{11}$

(c) $(\boldsymbol{b}\boldsymbol{b}^\top)_{11} - (\boldsymbol{B}_{\mathrm{NN}})_{11}$

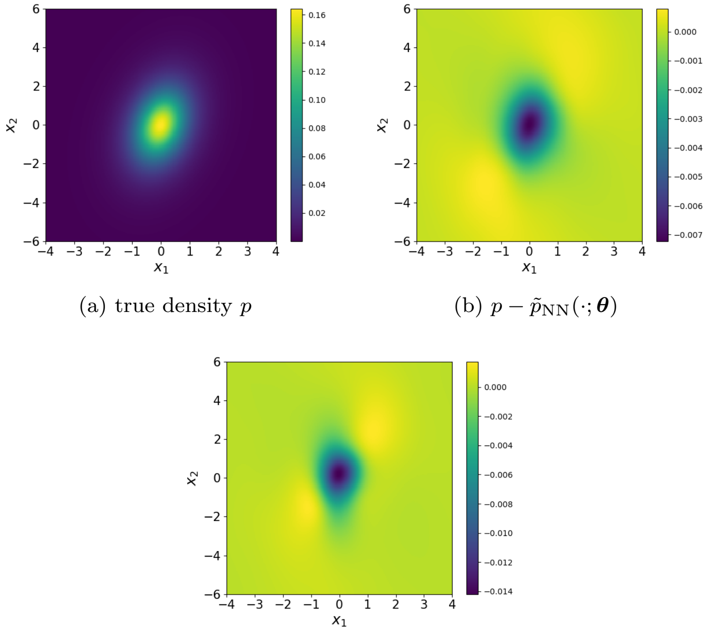
FIG. 5.2. *The comparison of first component of* $\boldsymbol{b}\boldsymbol{b}^\top$*.* (a) $(\boldsymbol{b}\boldsymbol{b}^\top)_{11}$*,* (b) $(\boldsymbol{B}_{NN})_{11}$*, and* (c) *their difference.*

TABLE 5.1
Comparison of mean and covariance statistics corresponding to the ground truth distribution π*, the discrete Markov chain induced by EM scheme in* (2.3)*,* $\tilde{\pi} := \pi^{EM}$*, and the discrete Markov chain generated by* (5.4) *for various* δt *whose invariant distribution is denoted as* $\hat{\pi}^{EM}$*. "N/A" means "not applicable".*

Distribution	π	$\tilde{\pi} := \pi^{\mathrm{EM}}$	$\hat{\pi}^{\mathrm{EM}}$	
δt	N/A	0.05	0.05	0.01
Mean	$[0.000 \quad 0.000]$	$[-0.002 \quad 0.000]$	$[0.000 \quad 0.004]$	$[-0.006 \quad -0.005]$
Covariance	$\begin{bmatrix}1.000 & 0.500\\0.500 & 4.000\end{bmatrix}$	$\begin{bmatrix}1.127 & 0.499\\0.499 & 4.398\end{bmatrix}$	$\begin{bmatrix}1.139 & 0.493\\0.493 & 4.389\end{bmatrix}$	$\begin{bmatrix}1.012 & 0.493\\0.493 & 4.061\end{bmatrix}$

where $\boldsymbol{U}(\boldsymbol{x}^n)\boldsymbol{S}(\boldsymbol{x}^n)\boldsymbol{U}(\boldsymbol{x}^n)^\top$ is the eigendecomposition of $\boldsymbol{B}_{\mathrm{NN}}(\boldsymbol{x}^n)$. We denote these empirical statistics to be defined with respect to the distribution $\hat{\pi}^{EM}$ that approximates $\hat{\pi}$. Compared to the ground truth statistics, the statistics of $\hat{\pi}^{EM}$ are subjected to errors from the estimation of $\boldsymbol{a}$, $\boldsymbol{b}\boldsymbol{b}^\top$ and from the EM integration. In Table 5.1, we note that, when $\delta t = 0.05$, the covariance estimate with $\boldsymbol{a}_{\mathrm{NN}}, \boldsymbol{B}_{\mathrm{NN}}$ is comparable to the error of $\boldsymbol{a}, \boldsymbol{b}\boldsymbol{b}^\top$. When $\delta t = 0.01$, the covariance estimate with $\boldsymbol{a}_{\mathrm{NN}}, \boldsymbol{B}_{\mathrm{NN}}$ becomes much closer to the ground truth.

5.1.3. Computation of the density function. We optimize the loss (3.10) with $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$ and obtain the solution $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$. The relative L_2 error between $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ and the true density (5.2) is 6.62×10^{-2}. To quantify the error induced by the regression alone, we replace $\boldsymbol{a}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$ of $\hat{\mathcal{L}}^*$ in (3.10) with the underlying coefficients, $\boldsymbol{a}$ and $\boldsymbol{b}\boldsymbol{b}^\top$, and optimize (3.10) with differential operator $\mathcal{L}^*$ in the first term. We denote the corresponding solution by $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$. The relative L_2 error between $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ and the true density (5.2) is 4.21×10^{-2}. We can see that $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$ achieves the error of same magnitude as $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$. Figure 5.3 shows the



(a) true density p

(b) $p - \tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$

(c) $p - \hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$

Fig. 5.3. *The comparison of solutions.* (a) *True density p,* (b) *difference between p and $\tilde{p}_{NN}(\cdot;\boldsymbol{\theta})$, and* (c) *difference between p and $\hat{p}_{NN}(\cdot;\boldsymbol{\theta})$. Here $\hat{p}_{NN}(\cdot;\boldsymbol{\theta})$ is obtained by optimizing* (3.10) *with $\boldsymbol{a}_{NN}$ and $\boldsymbol{B}_{NN}$, while $\tilde{p}_{NN}(\cdot;\boldsymbol{\theta})$ is obtained by optimizing* (3.10) *with $\boldsymbol{a}$ and $\boldsymbol{b}\boldsymbol{b}^\top$.*

true density and the differences between the true density and the network solutions $\hat{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$, $\tilde{p}_{\mathrm{NN}}(\cdot;\boldsymbol{\theta})$, plotted as functions of the computational domain Ω. Notice that the errors are more prominent when the coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$ are estimated, as expected.

We remark that the obtained $O(10^{-2})$ accuracy of this method is acceptable compared with other recent works using NNs to solve PDEs. Recent empirical results in solving deterministic PDEs using deep learning methods reported relative ℓ^2 errors between $O(10^{-4})$ and $O(10^{-2})$ (e.g., [17, 53, 70, 71]). In comparison, we solve the FP equation whose drift and diffusion terms are estimated from randomly generated data, which is more difficult than the purely deterministic problems.

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 mth particle is ma_0. Denote r_m as the displacement of the mth particle from its equilibrium position, and denote v_m as its velocity. The Langevin dynamics of this model is described as follows:

$$\begin{aligned} \dot{\boldsymbol{v}} &= -\nabla_{\boldsymbol{r}} U(\boldsymbol{r}) - \gamma \boldsymbol{v} + \sqrt{2k_BT\gamma}\,\dot{\boldsymbol{W}}_t, \\ \dot{\boldsymbol{r}} &= \boldsymbol{v}, \end{aligned} \tag{5.5}$$

where $\boldsymbol{v} = [v_1, \ldots, v_M]^\top$ and $\boldsymbol{r} = [r_1, \ldots, r_M]^\top$ are the velocities and displacement of all particles; $\boldsymbol{W}_t = [W_t^{(1)}, \ldots, W_t^{(M)}]^\top$ is an M-dimensional Wiener process; U is some potential function; γ is the friction constant; k_BT is the temperature. The mass of particles is set to be unity in (5.5). The equilibrium distribution of (5.5) is given by

$$p(\boldsymbol{v},\boldsymbol{r}) \propto \exp\left[-\frac{1}{k_BT}\left(U(\boldsymbol{r})+\frac{1}{2}|\boldsymbol{v}|^2\right)\right]. \tag{5.6}$$

In the numerical simulation, we take the Lennard–Jones potential [27], which is given by

$$U(\boldsymbol{r})=\sum_{i=1}^{M}\sum_{j=i-2}^{i-1}\psi(r_i-r_j+(i-j)a_0), \quad r_0:=r_M, r_{-1}:=r_{M-1} \tag{5.7}$$

with

$$\psi(r)=|r|^{-12}-|r|^{-6}. \tag{5.8}$$

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

5.2.1. Data generation. We generate the data by EM discretization, namely,

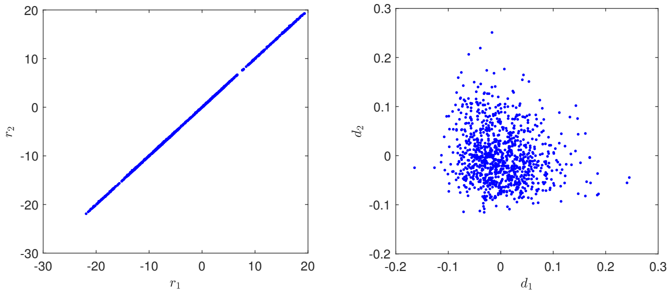
$$\begin{aligned}&\boldsymbol{v}^{n+1}=\boldsymbol{v}^n-(\nabla_{\boldsymbol{r}^n}U(\boldsymbol{r}^n)+\gamma\boldsymbol{v}^n)\delta t+\sqrt{2k_BT\gamma\delta t}\boldsymbol{\xi}, \quad \boldsymbol{\xi}\sim(\mathcal{N}(0,1))^M,\\&\boldsymbol{r}^{n+1}=\boldsymbol{r}^n+\boldsymbol{v}^n\delta t\end{aligned} \tag{5.9}$$

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

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

In this example, we set $\delta t = 0.0005$ and $N = 10^7$. Following the notation in section 3.2, we denote $\mathcal{X} := \{\boldsymbol{v}^n,\boldsymbol{r}^n\}_{n=0}^N$ as the original dataset. If we visualize the distribution of $\mathcal{X}$ by projecting it onto the (r_1,r_2)-plane (Figure 5.4), it is observed that displacement components are distributed near a straight line. To simplify the computation and visualization, we consider a coordinate transformation that maps $\mathcal{X}$ to a distribution that can be enclosed by a hyperrectangle. Specifically, we introduce the following coordinate transformation:

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



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

FIG. 5.4. *The distribution of the original dataset $\mathcal{X}$ in the (r_1,r_2)-plane and the distribution of the transformed dataset $\hat{\mathcal{X}}$ in the (d_1,d_2)-plane.*

Table 5.2

Comparison of mean and covariance statistics ($\boldsymbol{v}_1$ and $\boldsymbol{d}_1$) corresponding to the ground truth distribution π, the discrete Markov chain induced by EM scheme in (2.3)*, $\tilde{\pi}$, and the discrete Markov chain generated by* (5.10) *for $\delta t = 0.0005$ whose invariant distribution is denoted as $\hat{\pi}^{EM}$. "N/A" means "not applicable".*

Distribution	π	$\tilde{\pi}$	$\hat{\pi}^{\mathrm{EM}}$
δt	N/A	0.0005	0.0005
Mean	$[0 \quad 0]$	$[-0.00363 \quad -0.00013]$	$[-0.00153 \quad -0.00003]$
Covariance	$\begin{bmatrix} 0.40229 & -0.01749 \\ -0.01749 & 0.00245 \end{bmatrix}$	$\begin{bmatrix} 0.37816 & 0.00008 \\ 0.00008 & 0.00292 \end{bmatrix}$	$\begin{bmatrix} 0.40916 & 0.00041 \\ 0.00041 & 0.00314 \end{bmatrix}$

where $\boldsymbol{d}$ is called the relative displacement. Note that the map $\mathcal{T}$ implies $r_1 - r_M = -\sum_{m=1}^{M-1} d_m$. If we define the transformed dateset $\hat{\mathcal{X}} := \{\boldsymbol{v}^n, \boldsymbol{d}^n\}_{n=0}^N$ with $\boldsymbol{d}^n = \mathcal{T}(\boldsymbol{r}^n)$ and project it onto the (d_1, d_2)-plane (Figure 5.4), then it is observed that most points in $\hat{\mathcal{X}}$ are located near the origin and form a circular region. Consequently, we apply the proposed method to the transformed dataset $\hat{\mathcal{X}}$ in the practical computation.

5.2.2. Identification of the drift and diffusion coefficients. Now we aim to identify the drift term $\boldsymbol{a}(\boldsymbol{v}, \boldsymbol{r})$ and the diffusion $\boldsymbol{b}\boldsymbol{b}^\top$ of the underlying dynamics. Due to the transformation $\mathcal{T}$, we define $\hat{\boldsymbol{a}}(\boldsymbol{v}, \boldsymbol{d}) := \boldsymbol{a}(\boldsymbol{v}, \boldsymbol{r})$ and aim to identify $\hat{\boldsymbol{a}}$ by the optimization (3.3) using the dataset $\hat{\mathcal{X}}$. Note $\hat{\boldsymbol{a}}(\boldsymbol{v}, \boldsymbol{d})$ 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 $\hat{\boldsymbol{a}}(\boldsymbol{v}, \boldsymbol{d})$, solving the regression problem in (3.3). In this application, this is a regression over training dataset $(\hat{\mathcal{X}}, \mathcal{Y})$, where $\mathcal{Y} := \{\frac{\boldsymbol{v}^{n+1} - \boldsymbol{v}^n}{\delta t}, \frac{\boldsymbol{r}^{n+1} - \boldsymbol{r}^n}{\delta t}\}_{n=0}^{N-1}$

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

Next, we consider the approximation $\boldsymbol{B}_{NN}$ to the constant matrix $\boldsymbol{b}\boldsymbol{b}^\top$ using the formula in (3.6). In this example, since $\boldsymbol{b}\boldsymbol{b}^\top$ is a diagonal matrix, we also set $\boldsymbol{B}_{\mathrm{NN}}$ to be diagonal with components $(b_{11}, \ldots, b_{2M,2M})$. The errors $|b_{kk} - (\boldsymbol{b}\boldsymbol{b}^\top)_{kk}|$ for the first M components are observed to be between 6.32×10^{-6} and 2.67×10^{-6}, and the errors for the next M components are between 8.07×10^{-13} and 3.63×10^{-12}.

Similar to the previous example, we simulate the Markov chain of the estimated $\hat{\boldsymbol{a}}_{\mathrm{NN}}$ and $\boldsymbol{B}_{\mathrm{NN}}$,

$$\begin{aligned} &\boldsymbol{v}^{n+1} - \boldsymbol{v}^n = (\hat{\boldsymbol{a}}_{\mathrm{NN}})_{1:M}(\boldsymbol{v}^n, \boldsymbol{d}^n)\delta t + (\boldsymbol{B}_{\mathrm{NN}})^{\frac{1}{2}} \sqrt{\delta t} \boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I}_M), \\ &\boldsymbol{r}^{n+1} - \boldsymbol{r}^n = (\hat{\boldsymbol{a}}_{\mathrm{NN}})_{M+1:2M}(\boldsymbol{v}^n, \boldsymbol{d}^n)\delta t, \end{aligned} \tag{5.10}$$

and compare its statistics with those of the ground truth. For the covariance of π, Monte Carlo integration with 10^8 points is used. For the statistics of $\tilde{\pi}$ and $\hat{\pi}^{\mathrm{EM}}$, we generate a sequence of 10^7 points. The information is shown in Table 5.2 for the components $\boldsymbol{v}_1$ and $\boldsymbol{d}_1$. Notice that, in this case, the statistical error for estimating $\hat{\pi}^{EM}$ is not much worse than the Monte Carlo error of $\tilde{\pi}$.

5.2.3. Computation of the density function. In this section, we aim to recover the equilibrium density function based on the obtained $\{\hat{a}_{\mathrm{NN}}(\boldsymbol{v}, \boldsymbol{d}; \boldsymbol{\theta}_k)\}$ and

$\boldsymbol{B}_{\mathrm{NN}}$. We let $p(\boldsymbol{v},\boldsymbol{r})$ be the original density function in $(\boldsymbol{v},\boldsymbol{r})$-coordinates and define $\hat{p}(\boldsymbol{v},\boldsymbol{d}) := p(\boldsymbol{v},\boldsymbol{r})$ to be the density function under transformation $\mathcal{T}$. Since $p(\boldsymbol{v},\boldsymbol{r})$ satisfies (3.8), we can derive the PDE for $\hat{p}(\boldsymbol{v},\boldsymbol{d})$, which is given by

$$\begin{aligned}(5.11)\quad &-\sum_{k=1}^{M}\frac{\partial}{\partial v_k}(\hat{p}\hat{a}_k)-\sum_{k=M+1}^{2M-1}\frac{\partial}{\partial d_{k-M}}\left(\hat{p}(\hat{a}_{k+1}-\hat{a}_k)\right)\\ &+\frac{1}{2}\sum_{k=1}^{M}(\boldsymbol{b}\boldsymbol{b}^\top)_{kk}\frac{\partial^2}{\partial v_k^2}\hat{p}+\frac{1}{2}(\boldsymbol{b}\boldsymbol{b}^\top)_{M+1,M+1}\frac{\partial^2}{\partial d_1^2}\hat{p}\\ &+\frac{1}{2}\sum_{k=2}^{M-1}(\boldsymbol{b}\boldsymbol{b}^\top)_{k+M,k+M}\left(\frac{\partial}{\partial d_k}-\frac{\partial}{\partial d_{k-1}}\right)^2\hat{p}+\frac{1}{2}(\boldsymbol{b}\boldsymbol{b}^\top)_{2M,2M}\frac{\partial^2}{\partial d_{M-1}^2}\hat{p}=0,\end{aligned}$$

where $\hat{a}_k$ denotes the kth component of $\hat{\boldsymbol{a}}$.

Once the drift and diffusion coefficients are estimated, we substitute $\hat{a}_k$ with the kth FNN estimate, denoted as $\hat{a}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}_k)$, and $(\boldsymbol{b}\boldsymbol{b}^\top)_{k,k}$ with the diagonal components of the estimated diffusion matrix, $b_{kk} := (\boldsymbol{B}_{\mathrm{NN}})_{kk}$, such that (5.11) becomes

$$\begin{aligned}(5.12)\quad &\\ &-\sum_{k=1}^{M}\frac{\partial}{\partial v_k}(\hat{p}\hat{a}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}_k))-\sum_{k=M+1}^{2M-1}\frac{\partial}{\partial d_{k-M}}\left(\hat{p}(\hat{a}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}_{k+1})-\hat{a}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}_k))\right)\\ &+\frac{1}{2}\left(\sum_{k=1}^{M}b_{kk}\frac{\partial^2\hat{p}}{\partial v_k^2}+b_{M+1,M+1}\frac{\partial^2\hat{p}}{\partial d_1^2}\right.\\ &\qquad\left.+\sum_{k=2}^{M-1}b_{k+M,k+M}\left(\frac{\partial}{\partial d_k}-\frac{\partial}{\partial d_{k-1}}\right)^2\hat{p}+b_{2M,2M}\frac{\partial^2\hat{p}}{\partial d_{M-1}^2}\right)=0.\end{aligned}$$

Next, we select a bounded domain in which the PDE (5.12) 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 Ω to also be densely covered by the points in $\hat{\mathcal{X}}$. By this principle, we set c_k as the componentwise mean of the points in $\hat{\mathcal{X}}$, namely,

$$(5.13)\qquad c_k=\begin{cases}\frac{1}{N}\sum_{n=1}^{N}v_k & \text{for } k=1,\dots,M,\\ \frac{1}{N}\sum_{n=1}^{N}d_k & \text{for } k=M+1,\dots,2M-1,\end{cases}$$

and set s_k empirically as follows:

$$(5.14)\qquad s_k=\begin{cases}1.0 & \text{for } k=1,\dots,M,\\ 0.1 & \text{for } k=M+1,\dots,2M-1.\end{cases}$$

For clarity, we display the projections of $\hat{\mathcal{X}}$ and Ω onto coordinate planes in Figure 5.5.

We take a neural network $\hat{p}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta})$ to approximate $\hat{p}(\boldsymbol{v},\boldsymbol{d})$. Then we solve the PDE (5.12) with the least squares method introduced in section 3.3 to determine $\hat{p}_{\mathrm{NN}}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta})$. Specifically, we solve the least squares problem in (3.14) with $\lambda_1 = 1$ and $\lambda_2 = 0$, ignoring the artificial boundary constraint since the function values at the boundary $\partial\Omega$ are small: they range from 7×10^{-7} to 4×10^{-6}. Meanwhile, 90% of the points in $\hat{\mathcal{X}}\cap\Omega$ are selected as the training set, denoted as $\hat{D}_{\mathrm{T}}$,

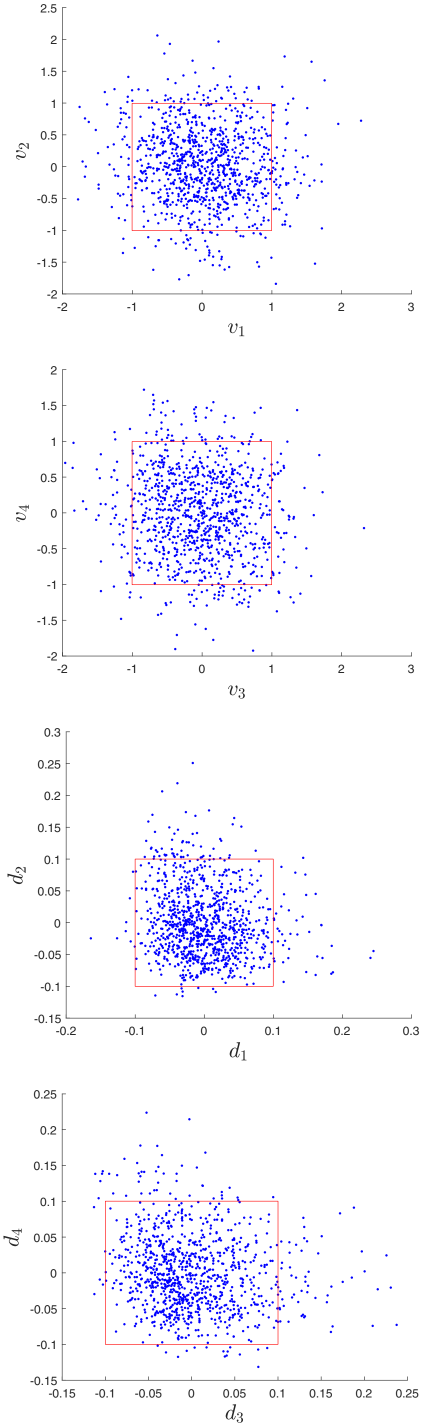


FIG. 5.5. *The projections of the dataset $\mathcal{X}$ (blue points) and the enclosing region Ω (red boxes) onto (v_1, v_2)-, (v_3, v_3)-, (d_1, d_2)-, (d_3, d_4)-planes.*

and the other 10% are chosen as the testing set, denoted as $\hat{D}_\mathrm{S}$, for the evaluation of the solution error. In practice. we set $\hat{p}_\mathrm{NN}$ to be a fully connected network having 3 layers and 100 neurons in each layer with ReLU^3 activation. The Adam optimizer is used to solve the optimization with 1,000 epochs, and the learning rates are set to decay from 10^{-4} to 10^{-5}. Once $\hat{p}_\mathrm{NN}$ is obtained from the minimization of (3.14), the integral $\int_{\mathbb{R}^{2M-1}} \hat{p}_\mathrm{NN}$ may not be quite close to 1, so we next perform an additional normalization to the estimated $\hat{p}_\mathrm{NN}$. Specifically, we approximate $I = \int_{\mathbb{R}^{2M-1}} \hat{p}_\mathrm{NN}$ by a Monte Carlo integration with a vast number of sample points. We repeat doubling the sample points to refine the numerical integral until it converges with stopping threshold 10^{-6}, i.e., $|I(N) - I(N/2)| \le 10^{-6}$, where $I(N)$ denotes the numerical integral with N sample points. Then $\hat{p}_\mathrm{NN}$ is normalized by the estimated I.

Next, 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 (5.6), we directly have the expression of $\hat{p}(\boldsymbol{v},\boldsymbol{d})$, namely,

$$\hat{p}(\boldsymbol{v},\boldsymbol{d}) = c \cdot \exp\left[-\frac{1}{k_B T}\left(\hat{U}(\boldsymbol{d}) + \frac{1}{2}|\boldsymbol{v}|^2\right)\right] \tag{5.15}$$

with

(5.16)

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

where c is a normalization constant such that

$$\int_{\mathbb{R}^{2M-1}} \hat{p}(\boldsymbol{v},\boldsymbol{d}) = 1. \tag{5.17}$$

Therefore c can be computed as

$$c = \left(\int_{\mathbb{R}^{2M-1}} \exp\left[-\frac{1}{k_B T}\left(\hat{U}(\boldsymbol{d}) + \frac{1}{2}|\boldsymbol{v}|^2\right)\right]\right)^{-1}. \tag{5.18}$$

Since there is no closed form for the integral in (5.18), we approximate c numerically by the Monte Carlo method.

Subsequently, the relative ℓ^2 error between $\hat{p}_\mathrm{NN}(\boldsymbol{v},\boldsymbol{d})$ and $\hat{p}(\boldsymbol{v},\boldsymbol{d})$ is computed according to (5.3) with $L^2(\Omega)$ replaced by $L^2(\hat{D}_S)$, where the integral is replaced by an average over the testing dataset $\hat{D}_S$. In this numerical result, we found that the relative ℓ^2 error of the computed density function $\hat{p}_\mathrm{NN}$ is 5.402×10^{-2}. In Figure 5.6, we also show the marginal densities of $\hat{p}_\mathrm{NN}$

$$\begin{aligned}\hat{p}^{\mathrm{marginal}}_{\mathrm{NN},k}(v_k) &:= \int_{(\boldsymbol{v},\boldsymbol{d})\setminus v_k \in \mathbb{R}^{2M-2}} \hat{p}_\mathrm{NN}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}) \quad \text{for } k = 1,\dots,M,\\ \hat{p}^{\mathrm{marginal}}_{\mathrm{NN},k}(d_k) &:= \int_{(\boldsymbol{v},\boldsymbol{d})\setminus d_k \in \mathbb{R}^{2M-2}} \hat{p}_\mathrm{NN}(\boldsymbol{v},\boldsymbol{d};\boldsymbol{\theta}) \quad \text{for } k = M+1,\dots,2M-1\end{aligned} \tag{5.19}$$

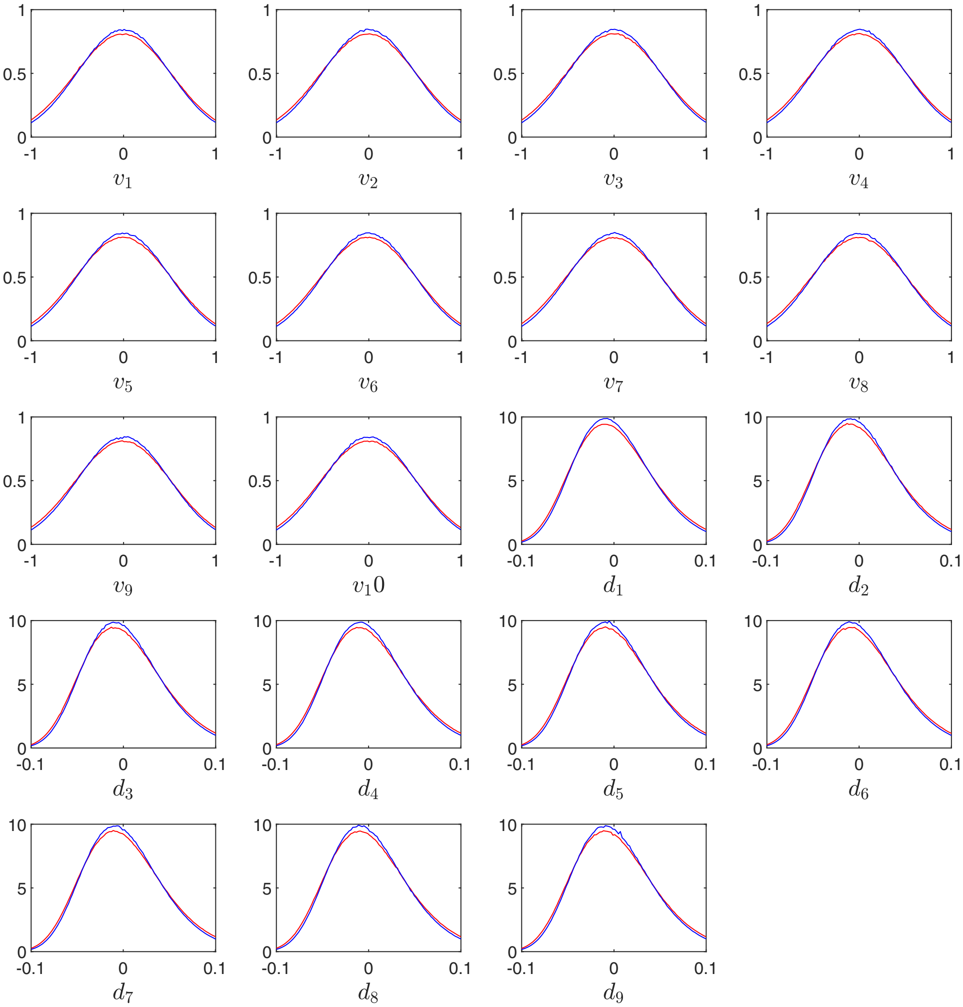


FIG. 5.6. *Marginal densities of the computed density function $\hat{p}_{NN}$ (red curves) and the true density function $\hat{p}$ (blue curves) for all components.*

compared with the following true marginal densities:

$$\hat{p}_k^{\text{marginal}}(v_k) := \int_{(\boldsymbol{v},\boldsymbol{d})\setminus v_k \in \mathbb{R}^{2M-2}} \hat{p}(\boldsymbol{v},\boldsymbol{d}) \quad \text{for } k = 1,\dots,M,$$

(5.20)

$$\hat{p}_k^{\text{marginal}}(d_k) := \int_{(\boldsymbol{v},\boldsymbol{d})\setminus d_k \in \mathbb{R}^{2M-2}} \hat{p}(\boldsymbol{v},\boldsymbol{d}) \quad \text{for } k = M+1,\dots,2M-1,$$

where the integrals in (5.17), (5.19), and (5.20) 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 nonsymmetric.

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 EM 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 squares regressions corresponding to the available dataset, and the second step is solving the steady-state FP 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 a constant diffusion matrix, accounting errors from 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 FP 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 [74], generalization errors in approximating Lipschitz continuous functions in [29], and in solving PDEs in [42].

From this theoretical study, we observe two difficult aspects that warrant careful treatment in future studies. The first issue concerns the incompatibility of the topologies that characterize the perturbation theory and machine learning generalization theory. Since the bound in (4.6) is stronger than an L^2 error bound in generalization theory, one requires a tacit assumption of consistency in the sense of (4.4), which is not easily verified in practice. The second issue concerns the incompatibility of the computational and physical domains, which is admitted under Assumption 4.2. Particularly, while the underlying stochastic process is defined on $\mathbb{R}^d$, the error estimation that accounts for finite samples and 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. Finally, recall that, in the analysis of regression error for the drift estimator $\boldsymbol{a}$, we derived an error estimate for deep networks of any width and depth. On the other hand, in the error analysis for the FP solution $\hat{p}$, only results with two-layer shallow networks were derived in the current work. It is promising to extend this result to deep networks in future work.

Numerically, we verified the effectiveness of the proposed method on two examples: a two-dimensional Student 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 (3.14) requires samples $\boldsymbol{x}_I^n, \boldsymbol{x}_{II}^n$, 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 (3.14) 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 (3.14) is unavoidable. In the future, we plan to consider different penalties such as the one proposed in [72] which requires no additional samples other than the available time series.

Appendix A. Proofs for section 4.

Proof of Lemma 4.5. Since $f_0 \in \mathcal{B}_{\mathrm{ReLU}}$, by [67, Theorem 12], there exists a two-layer ReLU FNN f^* with width W such that $\|f^*\|_{L^\infty([0,1]^d)} \leq \|f_0\|_{\mathcal{B}_{\mathrm{ReLU}}}$ and

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

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

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

Also, [29, Lemma 3.2] implies that

$$\begin{aligned}&\mathbb{E}_\nu\left[|f_{\mathrm{NN}}(\cdot,\boldsymbol{\theta}^{f_0})-f_0|^2\right]\\&\quad\leq C\left[P^2W(d+W)\log(Wd+W^2)(\log N)^3N^{-1}+\inf_{f\in\mathcal{F}^P_{2,W,\mathrm{ReLU}}}\mathbb{E}_\nu\left[|f-f_0|^2\right]\right],\end{aligned}\tag{A.2}$$

where C is a constant that does not depend on d, N, W, f_0, P. Combining (A.1) and (A.2) completes the proof. □

Proof of Lemma 4.7. Denote $\hat{e}:=q-\hat{p}$. On the one hand, using integration by parts,

$$\begin{aligned}\int_\Omega\hat{\mathcal{L}}^*\hat{e}\cdot\hat{e}\,\mathrm{d}\boldsymbol{x}\geq&\int_\Omega\sum_{i,j=1}^d\frac{1}{2}B^{ij}_{\mathrm{NN}}\hat{e}_{x_i}\hat{e}_{x_j}\mathrm{d}\boldsymbol{x}-\int_{\partial\Omega}\left(\sum_{i,j=1}^d\frac{1}{2}B^{ij}_{\mathrm{NN}}|\hat{e}_{x_i}|\cdot|\mathrm{n}_j|\right)|\hat{e}|\mathrm{d}s\\&+\int_\Omega\sum_{i=1}^d a^i_{\mathrm{NN}}\hat{e}_{x_i}\cdot\hat{e}+\left(\sum_{i=1}^d\frac{\partial a^i_{\mathrm{NN}}}{\partial x_i}\right)\hat{e}^2\mathrm{d}\boldsymbol{x}\\\geq&\frac{1}{2}\Lambda\int_\Omega\|\nabla\hat{e}\|^2\mathrm{d}x-\frac{1}{2}dB_1\int_{\partial\Omega}\|\nabla\hat{e}\|\cdot|\hat{e}|\mathrm{d}s\\\geq&\frac{1}{2}\Lambda\|\nabla\hat{e}\|^2_{L^2(\Omega)}-\frac{1}{2}dB_1\left(\|\nabla q\|_{L^2(\partial\Omega)}+\|\nabla\hat{p}\|_{L^2(\partial\Omega)}\right)\|\hat{e}\|_{L^2(\partial\Omega)}\\\geq&\frac{1}{2}\Lambda\|\nabla\hat{e}\|^2_{L^2(\Omega)}-\frac{1}{2}dB_1\left(B_2+\epsilon_{\hat{p}}\right)\|\hat{e}\|_{L^2(\partial\Omega)},\end{aligned}\tag{A.3}$$

where n_j is the jth component of the outward unit normal vector.

On the other hand,

$$\int_\Omega\hat{\mathcal{L}}^*\hat{e}\cdot\hat{e}\mathrm{d}x=\int_\Omega\hat{\mathcal{L}}^*q\cdot\hat{e}\mathrm{d}x\leq\|\hat{\mathcal{L}}^*q\|_{L^2(\Omega)}\cdot\|\hat{e}\|_{L^2(\Omega)}.\tag{A.4}$$

Combining (A.3) and (A.4) leads to

$$\|\nabla\hat{e}\|^2_{L^2(\Omega)}\leq2\Lambda^{-1}\|\hat{\mathcal{L}}^*q\|_{L^2(\Omega)}\cdot\|\hat{e}\|_{L^2(\Omega)}+\Lambda^{-1}dB_1(B_2+\epsilon_{\hat{p}})\|\hat{e}\|_{L^2(\partial\Omega)}.\tag{A.5}$$

Next, by Poincaré inequality, there exist some $C_1>0$ that only depend on Ω 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 (A.5) and the fact $\int_\Omega\hat{p}\mathrm{d}\boldsymbol{x}=1$,

$$\begin{aligned}\|\hat{e}\|^2_{L^2(\Omega)}&\leq C_3\left[\left|\int_\Omega\hat{e}\mathrm{d}\boldsymbol{x}\right|^2+\|\nabla\hat{e}\|^2_{L^2(\Omega)}\right]\\&\leq C_3\left[\left|\int_\Omega q\mathrm{d}x-1\right|^2+2\Lambda^{-1}\|\hat{\mathcal{L}}^*q\|_{L^2(\Omega)}\cdot\|\hat{e}\|_{L^2(\Omega)}+\Lambda^{-1}dB_1(B_2+\epsilon_{\hat{p}})\|\hat{e}\|_{L^2(\partial\Omega)}\right],\end{aligned}\tag{A.6}$$

where $C_3 = 2C_2^2$. Using the Young inequality $2C_3\Lambda^{-1}\|\hat{\boldsymbol{\mathcal{L}}}^* q\|_{L^2(\Omega)} \cdot \|\hat e\|_{L^2(\Omega)} \le \frac{4C_3^2\Lambda^{-2}\|\hat{\boldsymbol{\mathcal{L}}}^* q\|^2_{L^2(\Omega)}+\|\hat e\|^2_{L^2(\Omega)}}{2}$, it follows from (A.6) that

$$\frac{1}{2}\|\hat e\|^2_{L^2(\Omega)} \le C_3 \left|\int_\Omega q \mathrm{d}x - 1\right|^2 + 2C_3^2\Lambda^{-2}\|\hat{\boldsymbol{\mathcal{L}}}^* q\|^2_{L^2(\Omega)} + C_3\Lambda^{-1} dB_1(B_2+\epsilon_{\hat p})\|\hat e\|_{L^2(\partial\Omega)}. \tag{A.7}$$

Note $\|\hat e\|_{L^2(\partial\Omega)} \le \|\hat p\|_{L^2(\partial\Omega)} + \|q\|_{L^2(\partial\Omega)} \le \epsilon_{\hat p} + \|q\|_{L^2(\partial\Omega)}$; it follows from (A.7) that

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

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

Proof of Lemma 4.8. Let $f = \mathbb{E}_{(c,\boldsymbol w)\sim\rho}[c\dot\sigma(\boldsymbol w^\top \boldsymbol x)]$ for some ρ taking the infimum in (4.21). Then $\hat{\boldsymbol{\mathcal{L}}}^* f = \mathbb{E}_{(c,\boldsymbol w)\sim\rho}[\hat{\boldsymbol{\mathcal{L}}}^*(c\dot\sigma(\boldsymbol w^\top\boldsymbol x))]$. Using the homogeneity of the neuron $c\dot\sigma(\boldsymbol w^\top\boldsymbol x)$, we may assume that $\|\boldsymbol w\|_1 = 1$ and $|c| = \|f\|_{\mathcal{B}_{\dot\sigma}}$ ρ-almost everywhere. Indeed, denote p_0 as the density of ρ; we define the probability measure ρ^* with the density

$$p_0^*(\hat c, \hat{\boldsymbol w}) = \begin{cases}\int_{c\|\boldsymbol w\|_1^3=\hat c} p_0(c,\boldsymbol w)\mathrm{d}c\mathrm{d}\boldsymbol w & \text{if } \|\hat{\boldsymbol w}\|_1 = 1,\\ 0, \quad \text{otherwise;}\end{cases} \tag{A.8}$$

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 $\mathrm{supp}(p_0^*) \subset \mathbb{R}\times\{\|\hat{\boldsymbol w}\|_1 = 1\}$. Moreover, we define the probability measure ρ^{**} with the density

$$p_0^{**}(\tilde c, \tilde{\boldsymbol w}) = \begin{cases}\|f\|^{-1}_{\mathcal{B}_{\dot\sigma}}\int_0^{+\infty}|\hat c|p_0^*(\hat c,\hat{\boldsymbol w})\mathrm{d}\hat c & \text{if } \tilde c = \|f\|_{\mathcal{B}_{\dot\sigma}}, \|\tilde{\boldsymbol w}\|_1 = 1,\\ \|f\|^{-1}_{\mathcal{B}_{\dot\sigma}}\int_{-\infty}^{0}|\hat c|p_0^*(\hat c,\hat{\boldsymbol w})\mathrm{d}\hat c & \text{if } \tilde c = -\|f\|_{\mathcal{B}_{\dot\sigma}}, \|\tilde{\boldsymbol w}\|_1 = 1,\\ 0, \quad \text{otherwise;}\end{cases} \tag{A.9}$$

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 $\mathrm{supp}(p_0^{**}) \subset \{\tilde c = \pm\|f\|_{\mathcal{B}_{\dot\sigma}}\}\times\{\|\tilde{\boldsymbol w}\|_1 = 1\}$.

Let $\{(c_m, \boldsymbol w_m)\}$ be M i.i.d. samples with ρ. By [58, Lemma 26.2],

$$\begin{aligned}&\mathbb{E}_{\{(c_m,\boldsymbol w_m)\}\sim\rho^M}\left[\sup_{\boldsymbol x\in\Omega}\hat{\boldsymbol{\mathcal{L}}}^*\left(\frac{1}{M}\sum_{m=1}^M c_m\dot\sigma(\boldsymbol w_m^\top\boldsymbol x)\right) - \hat{\boldsymbol{\mathcal{L}}}^* f(\boldsymbol x)\right]\\ &\quad = \mathbb{E}_{\{(c_m,\boldsymbol w_m)\}\sim\rho^M}\left[\sup_{\boldsymbol x\in\Omega}\left(\frac{1}{M}\sum_{m=1}^M\hat{\boldsymbol{\mathcal{L}}}^* c_m\dot\sigma(\boldsymbol w_m^\top\boldsymbol x) - \mathbb{E}_{(c,\boldsymbol w)\sim\rho}[\hat{\boldsymbol{\mathcal{L}}}^*(c\dot\sigma(\boldsymbol w^\top\boldsymbol x))]\right)\right]\\ &\quad \le 2\mathbb{E}_{\{(c_m,\boldsymbol w_m)\}\sim\rho^M}\mathbb{E}_\tau\left[\sup_{\boldsymbol x\in\Omega}\frac{1}{M}\sum_{m=1}^M\tau_m\hat{\boldsymbol{\mathcal{L}}}^*(c_m\dot\sigma(\boldsymbol w_m^\top\boldsymbol x))\right],\end{aligned} \tag{A.10}$$

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

Note that

$$\begin{aligned}
\text{(A.11)}\qquad &\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m\hat{\mathcal{L}}^*(c_m\dot{\sigma}(\boldsymbol{w}_m^\top\boldsymbol{x}))\right]\\
&=\mathbb{E}_\tau\Bigg[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m c_m\left(\frac{1}{2}\boldsymbol{w}_m^\top\boldsymbol{B}_{\mathrm{NN}}\boldsymbol{w}_m\dot{\sigma}''(\boldsymbol{w}_m^\top\boldsymbol{x})\right.\\
&\qquad\left.+\boldsymbol{a}_{\mathrm{NN}}^\top\boldsymbol{w}_m\dot{\sigma}'(\boldsymbol{w}_m^\top\boldsymbol{x})+\left(\sum_{i=1}^{d}\frac{\partial a_{\mathrm{NN}}^i}{\partial x_i}\right)\dot{\sigma}(\boldsymbol{w}_m^\top\boldsymbol{x})\right)\Bigg]\\
&\le\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\tau_m c_m\boldsymbol{w}_m^\top\boldsymbol{B}_{\mathrm{NN}}\boldsymbol{w}_m\dot{\sigma}''(\boldsymbol{w}_m^\top\boldsymbol{x})\right]\\
&\quad+\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m c_m\boldsymbol{a}_{\mathrm{NN}}^\top\boldsymbol{w}_m\dot{\sigma}'(\boldsymbol{w}_m^\top\boldsymbol{x})\right]\\
&\quad+\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m c_m\left(\sum_{i=1}^{d}\frac{\partial a_{\mathrm{NN}}^i}{\partial x_i}\right)\dot{\sigma}(\boldsymbol{w}_m^\top\boldsymbol{x})\right].
\end{aligned}$$

For the first term in (A.11), by the contraction lemma for Rademacher complexities [58, Lemma 26.9], we have

$$\begin{aligned}
\text{(A.12)}\qquad &\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\tau_m c_m\boldsymbol{w}_m^\top\boldsymbol{B}_{\mathrm{NN}}\boldsymbol{w}_m\dot{\sigma}''(\boldsymbol{w}_m^\top\boldsymbol{x})\right]\\
&=\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m\dot{\sigma}''\left(\frac{1}{2}c_m\boldsymbol{w}_m^\top\boldsymbol{B}_{\mathrm{NN}}\boldsymbol{w}_m\cdot\boldsymbol{w}_m^\top\boldsymbol{x}\right)\right]\\
&\le\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m B_1|c_m|\|\boldsymbol{w}_m\|_1^2\cdot\boldsymbol{w}_m^\top\boldsymbol{x}\right]\\
&=\frac{B_1}{M}\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\boldsymbol{x}^\top\sum_{m=1}^{M}\tau_m|c_m|\|\boldsymbol{w}_m\|_1^2\cdot\boldsymbol{w}_m\right]\\
&\le B_1\mathbb{E}_\tau\left\|\frac{1}{M}\sum_{m=1}^{M}\tau_m|c_m|\|\boldsymbol{w}_m\|_1^2\cdot\boldsymbol{w}_m\right\|_1.
\end{aligned}$$

Similarly, we can derive

(A.13)
$$\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m c_m\boldsymbol{a}_{\mathrm{NN}}^\top\boldsymbol{w}_m\dot{\sigma}'(\boldsymbol{w}_m^\top\boldsymbol{x})\right]\le B_1\mathbb{E}_\tau\left\|\frac{1}{2M}\sum_{m=1}^{M}\tau_m|c_m|\|\boldsymbol{w}_m\|_1\cdot\boldsymbol{w}_m\right\|_1$$

and

(A.14)
$$\mathbb{E}_\tau\left[\sup_{\boldsymbol{x}\in\Omega}\frac{1}{M}\sum_{m=1}^{M}\tau_m c_m\left(\sum_{i=1}^{d}\frac{\partial a_{\mathrm{NN}}^i}{\partial x_i}\right)\dot{\sigma}(\boldsymbol{w}_m^\top\boldsymbol{x})\right]\le B_1\mathbb{E}_\tau\left\|\frac{1}{2M}\sum_{m=1}^{M}\tau_m|c_m|\boldsymbol{w}_m\right\|_1.$$

Denote $\boldsymbol{u}_m := c_m\boldsymbol{w}_m$; then $\|\boldsymbol{u}_m\|_1 = \|f\|_{\mathcal{B}_{\dot\sigma}}$. We combine (A.10)–(A.14) and obtain

$$\begin{aligned}
\text{(A.15)}\qquad &\mathbb{E}_{\{(c_m,\boldsymbol{w}_m)\}\sim\rho^M}\left[\sup_{\boldsymbol{x}\in\Omega}\hat{\mathcal{L}}\left(\frac{1}{M}\sum_{m=1}^{M}c_m\dot\sigma(\boldsymbol{w}_m^\top\boldsymbol{x})\right)-\hat{\mathcal{L}}f(\boldsymbol{x})\right]\\
&\le 2\sup_{\|\boldsymbol{u}_m\|_1\le\|f\|_{\mathcal{B}_{\dot\sigma}}} 2B_1\mathbb{E}_\tau\|\frac{1}{M}\sum_{m=1}^{M}\tau_m\boldsymbol{u}_m\|_1\\
&\le 2\|f\|_{\mathcal{B}_{\dot\sigma}}\sup_{\|\boldsymbol{u}_m\|_1\le 1} 2B_1\mathbb{E}_\tau\|\frac{1}{M}\sum_{m=1}^{M}\tau_m\boldsymbol{u}_m\|_1\\
&\le 2\sqrt{d}\|f\|_{\mathcal{B}_{\dot\sigma}}\sup_{\|\boldsymbol{u}_m\|_2\le 1} 2B_1\mathbb{E}_\tau\|\frac{1}{M}\sum_{m=1}^{M}\tau_m\boldsymbol{u}_m\|_2\\
&\le 4B_1\|f\|_{\mathcal{B}_{\dot\sigma}}\sqrt{d/M}
\end{aligned}$$

by using the Rademacher complexity of the unit ball [58, Lemma 26.10]. Applying the same argument to $-(\hat{\mathcal{L}}(\frac{1}{M}\sum_{m=1}^{M}c_m\dot\sigma(\boldsymbol{w}_m^\top\boldsymbol{x}))-\hat{\mathcal{L}}f(\boldsymbol{x}))$ leads to

(A.16)

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

By a similar argument, we can derive

$$\begin{aligned}
\text{(A.17)}\qquad &\mathbb{E}_{(c,\boldsymbol{w})\sim\rho}\left[\sup_{\boldsymbol{x}\in\Omega}\left|\frac{1}{M}\sum_{m=1}^{M}c_m\dot\sigma(\boldsymbol{w}_m^\top\boldsymbol{x})-f(\boldsymbol{x})\right|\right],\\
&\mathbb{E}_{(c,\boldsymbol{w})\sim\rho}\left[\sup_{\boldsymbol{x}\in\partial\Omega}\left|\frac{1}{M}\sum_{m=1}^{M}c_m\dot\sigma(\boldsymbol{w}_m^\top\boldsymbol{x})-f(\boldsymbol{x})\right|\right]\le\|f\|_{\mathcal{B}_{\dot\sigma}}\sqrt{d/M}.
\end{aligned}$$

Therefore we have

(A.18)

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

which implies there exists $\{(c_m,\boldsymbol{w}_m)\}_{m=1}^{M}$ such that the inequality holds. Then the FNN $\sum_{m=1}^{M}(c_m/M)\dot\sigma(\boldsymbol{w}_m^\top\boldsymbol{x})\in\mathcal{F}_{2,M,\dot\sigma,\max\{\|f\|_{\mathcal{B}_{\dot\sigma}}/M,1\}}$ satisfies (4.22). ∎

Proof of Lemma 4.10. Denote $\hat p^S_{\mathrm{NN}}(\boldsymbol{x})=\hat p_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}^S)$. Since $\hat p_{\mathrm{NN}}\in\mathcal{F}_{2,M,\dot\sigma,Q}$, using the expression in (4.2) we have $\|\nabla\hat p^S_{\mathrm{NN}}\|_{L^2(\partial\Omega)}\le\frac{1}{2}MQ^4|\partial\Omega|^{\frac{1}{2}}=\frac{1}{2}MQ^4(2d)^{\frac{1}{2}}$. Then the inequality (4.23) directly follows Lemma 4.7. For the rest, we use C to represent any constant which on depends on Ω, Λ, B_1, λ_1, and λ_2. On the one hand,

(A.19)

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

By virtue of [42, Theorem 3.2], with probability at least $1-\delta/3$,

(A.20)

$$\left| \|\mathcal{L}\hat{p}^S_{\mathrm{NN}}\|^2_{L^2(\Omega)} - \frac{|\Omega|}{N_1}\sum_{n=1}^{N_1} |\mathcal{L}\hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{I}})|^2 \right| + \lambda_2 \left| \|\hat{p}^S_{\mathrm{NN}}\|^2_{L^2(\partial\Omega)} - \frac{|\partial\Omega|}{N_3}\sum_{n=1}^{N_3} |\hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{III}})|^2 \right| \leq CI_1.$$

Similarly, by the fact $|\hat{p}^S_{\mathrm{NN}}(\boldsymbol{x})| \leq MQ^4/6 \ \forall \boldsymbol{x}$ and Lemma 4.9, we have, with probability at least $1-\delta/3$,

$$\left| \int_\Omega \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) \right| \leq CMQ^4\sqrt{\log(6/\delta)/N_2}, \tag{A.21}$$

and $\frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) \leq CMQ^4$. Then we have

(A.22)

$$\lambda_1 \left| \int_\Omega \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) \right| \cdot \left| \int_\Omega \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} \hat{p}^S_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) - 2 \right| \leq CI_2.$$

On the other hand, by Lemma 4.8 there exist some $p_{\mathrm{NN}} \in F_{2,M,\hat{\sigma},Q}$ such that

(A.23)

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

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

$$\begin{aligned}
&\left| \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) - 1 \right|^2 \\
&\qquad \leq 2\left(\left| \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} (p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}}) - \hat{p}(\boldsymbol{x}^n_{\mathrm{II}})) \right|^2 + \left| \frac{|\Omega|}{N_2}\sum_{n=1}^{N_2} \hat{p}(\boldsymbol{x}^n_{\mathrm{II}}) - \int_\Omega \hat{p}\mathrm{d}\boldsymbol{x} \right|^2 \right)
\end{aligned} \tag{A.24}$$

and

$$|p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{III}})|^2 \leq 2\left(|p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{III}}) - \hat{p}(\boldsymbol{x}^n_{\mathrm{III}})|^2 + \epsilon^2_{\hat{p}} \right), \tag{A.25}$$

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

Then it follows (A.23)–(A.25) and Lemma 4.9 that with probability at least $1-\delta/3$

$$\begin{aligned}J_S[\hat{p}^S_{\mathrm{NN}}] \le J_S[p_{\mathrm{NN}}] \le \frac{1}{N_1}\sum_{n=1}^{N_1}|\mathcal{L}p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{I}})|^2 + 2\lambda_1\left|\frac{|\Omega|}{N_2}\sum_{n=1}^{N_2}(p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{II}})-\hat{p}(\boldsymbol{x}^n_{\mathrm{II}}))\right|^2 \\ + 2\lambda_1\left|\frac{|\Omega|}{N_2}\sum_{n=1}^{N_2}\hat{p}(\boldsymbol{x}^n_{\mathrm{II}}) - \int_\Omega \hat{p}\right|^2 \\ + 2\lambda_2\frac{|\partial\Omega|}{N_3}\sum_{n=1}^{N_3}\left(|p_{\mathrm{NN}}(\boldsymbol{x}^n_{\mathrm{III}})-\hat{p}(\boldsymbol{x}^n_{\mathrm{III}})|^2+\epsilon^2_{\hat{p}}\right) \le CI_3.\end{aligned} \tag{A.26}$$

Finally, the proof can be completed by using (A.19), (A.20), (A.22), (A.26), and the fact $J[\hat{p}^S_{\mathrm{NN}}] \le |J[\hat{p}^S_{\mathrm{NN}}] - J_S[\hat{p}^S_{\mathrm{NN}}]| + J_S[\hat{p}^S_{\mathrm{NN}}]$. ∎

REFERENCES

[1] Z. A.-Zhu, Y. Li, and Z. Song, *A convergence theory for deep learning via overparameterization*, in Proceedings of the 36th International Conference on Machine Learning, Proc. Mach. Learn. Res. (PMLR) 97, K. Chaudhuri, and R. Salakhutdinov, eds., JMLR, Cambridge, MA, 2019, pp. 242–252.

[2] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 1999.

[3] T. A. Averina and S. S. Artemiev, *Numerical solution of systems of stochastic differential equations*, Russian J. Numer. Anal. Math. Modelling, 3 (1988), pp. 267–286.

[4] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945.

[5] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, *Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks*, J. Mach. Learn. Res., 20 (2019), pp. 2285–2301.

[6] P.-H. Chavanis, *Nonlinear mean field Fokker-Planck equations. Application to the chemotaxis of biological populations*, Eur. Phys. J. B, 62 (2008), pp. 179–208.

[7] M. Chen, H. Jiang, W. Liao, and T. Zhao, *Nonparametric regression on low-dimensional manifolds using deep ReLU networks: Function approximation and statistical recovery*, Inf. Inference J. IMA, 11 (2022), pp. 1203–1253.

[8] Z. Chen, Y. Cao, D. Zou, and Q. Gu, *How much over-parameterization is sufficient to learn deep ReLU networks?*, in Proceedings of the International Conference on Learning Representations, 2021.

[9] Z. Chen, J. Lu, and Y. Lu, *On the representation of solutions to elliptic PDEs in Barron spaces*, in Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Curran Associates, Red Hook, NY, 2021, pp. 6454–6465.

[10] P. Cheridito, A. Jentzen, and F. Rossmannek, *Non-convergence of stochastic gradient descent in the training of deep neural networks*, J. Complexity, 64 (2021), 101540.

[11] Z. Ding, S. Chen, Q. Li, and S. Wright, *Overparameterization of deep ResNet: Zero loss and mean-field analysis*, J. Mach. Learn. Res., 23 (2022), pp. 1–65.

[12] M. W. M. G. Dissanayake and N. Phan-Thien, *Neural-network-based approximations for solving partial differential equations*, Comm. Numer. Methods Eng., 10 (1994), pp. 195–201.

[13] S. S. Du, X. Zhai, B. Poczos, and A. Singh, *Gradient descent provably optimizes over-parameterized neural networks*, in Proceedings of the Seventh International Conference on Learning Representations, ICLR, 2019.

[14] C. Duan, Y. Jiao, Y. Lai, X. Lu, and Z. Yang, *Convergence rate analysis for deep Ritz method*, Commun. Comput. Phys., 31 (2022), pp. 1020–1048, https://doi.org/10.4208/cicp.OA-2021-0195.

[15] E. W., C. Ma, and L. Wu, *A priori estimates of the population risk for two-layer neural networks*, Commun. Math. Sci., 17 (2019), pp. 1407–1425.

[16] T. D. Frank, *Nonlinear Fokker-Planck Equations: Fundamentals and Applications*, Springer, Berlin, 2005.

[17] Y. Gu, H. Yang, and C. Zhou, *SelectNet: Self-paced learning for high-dimensional partial differential equations*, J. Comput. Phys., 441 (2021), 110444.

[18] I. Gühring and M. Raslan, *Approximation rates for neural networks with encodable weights in smoothness spaces*, Neural Netw., 134 (2021), pp. 107–130.

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

[20] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[21] S. Hess, *Fokker-Planck-equation approach to flow alignment in liquid crystals*, Z. Naturforsch. A, 31 (1976), pp. 1034–1037.

[22] S. Hon and H. Yang, *Simultaneous neural network approximations in sobolev spaces*, Neural Netw., 154 (2022), 152–164, https://doi.org/10.1016/j.neunet.2022.06.040.

[23] J. Huggins and J. Zou, *Quantifying the accuracy of approximate diffusions and Markov chains*, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, 2017, pp. 382–391.

[24] M. Hutzenthaler, A. Jentzen, T. 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*, SN Partial Differential Equations Appl., 1 (2020), 10, https://doi.org/10.1007/s42985-019-0006-9.

[25] J.-N. Hwang, S.-R. Lay, and A. Lippman, *Nonparametric multivariate density estimation: A comparative study*, IEEE Trans. Signal Process., 42 (1994), pp. 2795–2810.

[26] E. Iancu, A. Leonidov, and L. McLerran, *Nonlinear gluon evolution in the color glass condensate:* I, Nucl. Phys. A, 692 (2001), pp. 583–645.

[27] Y. Ishimori, *Solitons in a one-dimensional Lennard-Jones lattice*, Prog. Theor. Phys., 68 (1982), pp. 402–410.

[28] A. Jacot, F. Gabriel, and C. Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, in Advances in Neural Information Processing Systems 31 (NeurIPS 2021), Curran Associates, Red Hook, NY, 2018, pp. 8580–8589.

[29] Y. Jiao, G. Shen, Y. Lin, and J. Huang, *Deep Nonparametric Regression on Approximately Low-Dimensional Manifolds*, preprint, arXiv:2104.06708, 2021.

[30] S. Justin and S. Konstantinos, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys., 375 (2018), pp. 1339–1364.

[31] Y. Khoo, J. Lu, and L. Ying, *Solving parametric PDE problems with artificial neural networks*, European J. Appl. Math., 32 (2021), pp. 421–435.

[32] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, Y. Bengio, and Y. LeCun, eds., San Diego, CA, 2015.

[33] P. Kumar and S. Narayanan, *Solution of Fokker-Planck equation by finite element and finite difference methods for nonlinear systems*, Sadhana, 31 (2006), pp. 445–461.

[34] I. E. Lagaris, A. Likas, and D. I. Fotiadis, *Artificial neural networks for solving ordinary and partial differential equations*, IEEE Trans. Neural Netw., 9 (1998), pp. 987–1000.

[35] S. Liang, S. W. Jiang, J. Harlim, and H. Yang, *Solving PDEs on Unknown Manifolds With Machine Learning*, preprint, arxiv:2106.06682, 2021.

[36] S. Liu, W. Li, H. Zha, and H. Zhou, *Neural parametric Fokker–Planck equations*, SIAM J. Numer. Anal., 60 (2022), pp. 1385–1449, https://doi.org/10.1137/20M1344986.

[37] Z. Liu, W. Cai, and Z.-Q. J. Xu, *Multi-scale deep neural network (MscaleDNN) for solving Poisson-Boltzmann equation in complex domains*, Commun. Comput. Phys., 28 (2020), pp. 1970–2001.

[38] J. Lu, Z. Shen, H. Yang, and S. Zhang, *Deep network approximation for smooth functions*, SIAM J. Math. Anal., 53 (2021), pp. 5465–5506.

[39] L. Lu, H. Jiang, and W. H. Wong, *Multivariate density estimation by Bayesian sequential partitioning*, J. Amer. Statist. Assoc., 108 (2013), pp. 1402–1410.

[40] Y. Lu, J. Lu, and M. Wang, *A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations*, in Proceedings of Thirty Fourth Conference on Learning Theory, PMLR, 2021, pp. 3196–3241.

[41] Y. Lu, C. Ma, Y. Lu, J. Lu, and L. Ying, *A mean-field analysis of deep ResNet and beyond: Towards provable optimization via overparameterization from depth*, in Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020

[42] T. Luo and H. Yang, *Two-Layer Neural Networks for Partial Differential Equations: Optimization and Generalization Theory*, preprint, arXiv:2006.15733, 2020.

[43] J. C. Mattingly, A. M. Stuart, and D. J. Higham, *Ergodicity for SDEs and approximations: Locally Lipschitz vector fields and degenerate noise*, Stochastic Process. Appl., 101 (2002), pp. 185–232.

[44] S. Mei, A. Montanari, and P.-M. Nguyen, *A mean field view of the landscape of two-layer neural networks*, Proc. Natl. Acad. Sci. USA, 115 (2018), pp. E7665–E7671, https://doi.org/10.1073/pnas.1806579115.

[45] S. Mishra and R. Molinaro, *Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs*, IMA J. Numer. Anal., 42 (2021), pp. 981–1022.

[46] D. Misra, *Mish: A self regularized non-monotonic activation function*, in Proceedings of the 31st British Machine Vision Virtual Conference, British Machine Vision Association, 2020

[47] H. Montanelli and Q. Du, *New error bounds for deep ReLU networks using sparse grids*, SIAM J. Math. Data Sci., 1 (2019), pp. 78–92.

[48] H. Montanelli and H. Yang, *Error bounds for deep ReLU networks using the Kolmogorov-Arnold superposition theorem*, Neural Netw., 129 (2020), pp. 1–6.

[49] H. Montanelli, H. Yang, and Q. Du, *Deep ReLU networks overcome the curse of dimensionality for generalized bandlimited functions*, J. Comput. Math., 39 (2021), pp. 801–815.

[50] R. Nakada and M. Imaizumi, *Adaptive approximation and generalization of deep neural network with intrinsic dimensionality*, J. Mach. Learn. Res., 21 (2020), pp. 1–38.

[51] G. Papamakarios, T. Pavlakou, and I. Murray, *Masked autoregressive flow for density estimation*, in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, Curran Associates, Red Hook, NY, 2017, pp. 2338–2347.

[52] 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*, Int. J. Autom. Comput., 14 (2017), pp. 503–519.

[53] 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*, J. Comput. Phys., 378 (2019), pp. 686–707.

[54] H. Risken, *Fokker-Planck equation*, in The Fokker-Planck Equation, Springer, Berlin, 1996, pp. 63–95.

[55] M. Rosenblatt, *Remarks on some nonparametric estimates of a density function*, Ann. Math. Stat., 27 (1956), pp. 832–837.

[56] J. Schmidt-Hieber, *Nonparametric regression using deep neural networks with ReLU activation function*, Ann. Statist., 48 (2020), pp. 1875–1897.

[57] B. Sepehrian and M. K. Radpoor, *Numerical solution of non-linear Fokker-Planck equation using finite differences method and the cubic spline functions*, Appl. Math. Comput., 262 (2015), pp. 187–190.

[58] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, Cambridge, UK, 2014.

[59] Z. Shen, H. Yang, and S. Zhang, *Deep network with approximation error being reciprocal of width to power of square root of depth*, Neural Comput., 33 (2021), pp. 1005–1036.

[60] J. W. Siegel and J. Xu, *Approximation rates for neural networks with general activation functions*, Neural Netw., 128 (2020), pp. 313–321, https://doi.org/10.1016/j.neunet.2020.05.019.

[61] B. F. Spencer and L. A. Bergman, *On the numerical solution of the Fokker-Planck equation for nonlinear stochastic systems*, Nonlinear Dynam., 4 (1993), pp. 357–372.

[62] J. A. Tropp, *An Introduction to Matrix Concentration Inequalities*, Found. Trends Mach. Learn. 8, Now Publishers, Hanover, MA, 2015.

[63] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, *Neural autoregressive distribution estimation*, J. Mach. Learn. Res., 17 (2016), pp. 7184–7220.

[64] W. I. T. Uy and M. D. Grigoriu, *Neural network representation of the probability density function of diffusion processes*, Chaos, 30 (2020), 093118.

[65] M. Vidyasagar, *Learning and Generalisation: With Applications to Neural Networks*, 2nd ed., Springer, London, 2003.

[66] Z. Wang and D. W. Scott, *Nonparametric density estimation for high-dimensional data—Algorithms and applications*, Wiley Interdiscip. Rev. Comput. Stat., 11 (2019), e1461.

[67] E. Weinan, C. Ma, S. Wojtowytsch, and L. Wu, *Towards a mathematical understanding of neural network-based machine learning: What we know and what we don't*, CSIAM Trans. Appl. Math., 1 (2020), pp. 561–615, https://doi.org/10.4208/csiam-am.SO-2020-0002.

[68] Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, and J. Kurths, *Solving Fokker-Planck equation using deep learning*, Chaos, 30 (2020), 013133.

[69] D. Yarotsky and A. Zhevnerchuk, *The phase diagram of approximation rates for deep neural networks*, in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Curran Associates, Red Hook, NY, 2020, pp. 13005–13015.

[70] B. Yu, et al., *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat., 6 (2018), pp. 1–12.

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

[72] J. Zhai, M. Dobson, and Y. Li, *A deep learning method for solving Fokker-Planck equations*, in Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference, PMLR, 2021

[73] H. Zhang, J. Harlim, and X. Li, *Estimating linear response statistics using orthogonal polynomials: An RKHS formulation*, Found. Data Sci., 2 (2020), pp. 443–485.

[74] H. Zhang, J. Harlim, and X. Li, *Error bounds of the invariant statistics in machine learning of ergodic Itô diffusions*, Phys. D, 427 (2021), 133022.