Drift Estimation for Diffusion Processes Using Neural Networks Based on Discretely Observed Independent Paths

Yuzhen Zhao[1], **Yating Liu**[2], **Marc Hoffmann**[3]

[1]Sorbonne Université, and LEDa, Université Paris-Dauphine, PSL, Paris, France
[2]CEREMADE, CNRS, Université Paris-Dauphine, PSL, 75016 Paris, France
[3]CEREMADE, CNRS, Université Paris-Dauphine, PSL and Institut Universitaire de France, 75016 Paris, France
yuzhen.zhao@dauphine.eu, liu@ceremade.dauphine.fr, hoffmann@ceremade.dauphine.fr

Abstract

This paper addresses the nonparametric estimation of the drift function over a compact domain for a time-homogeneous diffusion process, based on high-frequency discrete observations from N independent trajectories. We propose a neural network-based estimator and derive a non-asymptotic convergence rate, decomposed into a training error, an approximation error, and a diffusion-related term scaling as $\log N/N$. For compositional drift functions, we establish an explicit rate. In the numerical experiments, we consider a drift function with local fluctuations generated by a double-layer compositional structure featuring local oscillations, and show that the empirical convergence rate becomes independent of the input dimension d. Compared to the B-spline method, the neural network estimator achieves better convergence rates and more effectively captures local features, particularly in higher-dimensional settings.

Code — https://github.com/yuzhen3001/nn-drift-estimation-diffusion-process

Extended version — https://arxiv.org/abs/2511.11161

1 Introduction

In this paper, we study the estimation of the drift function of a time-homogeneous diffusion process using neural networks. Specifically, we consider an $\mathbb{R}^d$-valued diffusion process $X = (X_t)_{t\in[0,T]}$ that solves the stochastic differential equation (SDE)

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}B_t, \tag{1.1}$$

where X_0 is a random vector, $B = (B_t)_{t\in[0,T]}$ is a d-dimensional standard Brownian motion, and $b : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^d \otimes \mathbb{R}^d$ are measurable functions. To estimate the drift, we assume that N independent and identically distributed (i.i.d.) sample paths $\bar{X}^{(n)} = (\bar{X}_t^{(n)})_{t\in[0,T]}$, $1 \le n \le N$, are available, each having the same distribution as X and independent of X. Moreover, each sample path is observed at high frequency, that is, at discrete time points on a fine grid:

$$\bar{X}^{(n)}_{t_0:t_M} := \big(\bar{X}^{(n)}_{t_0=0}, \bar{X}^{(n)}_{t_1}, \dots, \bar{X}^{(n)}_{t_M=T}\big),\ 1\le n\le N, \tag{1.2}$$

where $t_m = m\Delta$ with the time step $\Delta = \frac{T}{M} \to 0$ as $M \to \infty$.

Literature Review

Diffusion processes, as defined by (1.1), are fundamental stochastic models with various applications in physics, biology, and mathematical finance (see, e.g., Gardiner (2004); Bressloff (2014); Karatzas and Shreve (1998)). A central task is the estimation of the drift function, which has been extensively studied under two standard frameworks: the long-time horizon regime, often assuming ergodicity (see, e.g., Hoffmann (1999); Kutoyants (2004); Dalalyan (2005); Frishman and Ronceray (2020)), and the high-frequency regime over a fixed time interval (see, e.g., Comte and Genon-Catalot (2020); Denis, Dion-Blanc, and Martinez (2021); Yildiz et al. (2018)).

In recent years, neural networks have become widely used as effective tools for nonparametric function estimation in statistical learning (see, e.g., Yarotsky (2017); Schmidt-Hieber (2020)). They have also been successfully applied to drift estimation in diffusion processes under the long-time horizon setting with ergodicity assumptions (see, e.g., Oga and Koike (2024); Ren et al. (2024)). On the other hand, the neural network-based drift estimation in the high-frequency regime over a fixed time interval is beginning to be explored (see, e.g., (Gao, Barzel, and Yan 2024; Bae, Ha, and Jeong 2025)), and theoretical analyses remain limited. Our work contributes to this direction by providing convergence guarantees.

Contribution and Organization of This Paper

In this work, we address the problem of estimating the drift function b over a compact set $K \subset \mathbb{R}^d$, using neural networks defined further in (2.2), based on discrete observations of the process over a fixed and finite time horizon T. Importantly, our approach does not rely on the ergodicity of the diffusion process, in contrast to Oga and Koike (2024). Moreover, since the drift function can be estimated component-wise, we can focus on estimating each component separately and combine these estimators to obtain an estimator for b. Specifically, for a given $i \in \{1, \dots, d\}$, we consider the i-th component of b restricted to the compact domain $[0, 1]^d$. Our estimation target is therefore the function

$$f_0 := b^i \mathbf{1}_{[0,1]^d}, \tag{1.3}$$

where $\mathbf{1}_{[0,1]^d}$ denotes the indicator function of the domain $[0,1]^d$. It is worth noting that the assumption of a compact domain is standard in the literature (see, e.g., Hoffmann (1999); Comte, Genon-Catalot, and Rozenholc (2007); Oga and Koike (2024)). In this paper, the choice of $[0,1]^d$ is made for convenience in theoretical analysis and can be replaced by any compact set $K \subset \mathbb{R}^d$. In practice, the compact set K can be determined from the training data distribution (e.g., based on sample coverage or through a sample-splitting procedure).

The main contribution of this paper is Theorem 2.1, which establishes a non-asymptotic upper bound on the L^2 estimation risk in terms of the training error from neural network optimization, the approximation error over the neural network class, the number of training samples N, and the time step Δ of the observation grid. Moreover, under a composition structure assumption on f_0, as considered in Schmidt-Hieber (2020) and Oga and Koike (2024), we derive an explicit upper bound on the test error as a function of N.

The paper is organized as follows. In Section 2, we present the mathematical framework, along with the assumptions and main result (Theorem 2.1). Section 3 provides a simulation study illustrating the proposed method, where we compare the performance of our estimator with that of Denis, Dion-Blanc, and Martinez (2021), which uses a B-spline-based estimator and ridge estimation. The numerical results show that our estimator outperforms the reference method in Denis, Dion-Blanc, and Martinez (2021) in terms of convergence rate and the ability to capture the local fluctuations, especially in high-dimensional settings. Finally, a sketch of the proof of the main result is provided in Section 4.

Notations

For a vector or matrix W, let $|W|$ denote the Euclidean norm (if W is a vector) or the Frobenius norm (if W is a matrix). We write $|W|_\infty$ for the maximum-entry norm and $|W|_0$ for the number of nonzero entries. For $\beta \in \mathbb{R}$, $\lfloor \beta \rfloor$ denotes the largest integer *strictly* smaller than β. For two sequences (a_N) and (b_N), we write $a_N \lesssim b_N$ if there exists a constant C such that $a_N \leq C b_N$ for all N, and we write $a_N \asymp b_N$ if $a_N \lesssim b_N$ and $b_N \lesssim a_N$. Moreover, for a function $f : \mathbb{R}^d \to \mathbb{R}$, let $\|f\|_\infty := \sup_{x \in [0,1]^d} |f(x)|$. For a random variable X, we write $\mathcal{L}(X)$ for its law. The index $m = 0, \ldots, M-1$ corresponds to time steps on the discrete observation grid, while $n = 1, \ldots, N$ indexes the samples in the training set.

2 Mathematical Setting and Main Result

This section presents the mathematical setting and main result of the paper. We begin by introducing the neural network class used for estimation, followed by a description of the proposed estimator and the different loss functions considered. Finally, we present the assumptions and main result of the paper.

Neural Network Class for Estimation

We use feedforward neural networks to estimate the drift function component-wise. Specifically, let $\mathcal{F}_{L,\mathbf{p}}$ denote the class of functions defined by neural networks with L hidden layers, layer widths given by $\mathbf{p} = (p_0, p_1, \ldots, p_{L+1}) \in \mathbb{N}^{L+2}$, where $p_0 = d$ is the input dimension and $p_{L+1} = 1$ is the output dimension. Each $f \in \mathcal{F}_{L,\mathbf{p}}$ is a function from $\mathbb{R}^d$ to $\mathbb{R}$ of the form

$$f(x) = W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 x, \tag{2.1}$$

where W_i is a $p_{i+1} \times p_i$ weight matrix, $\sigma(x) = \max(x, 0)$ is the ReLU function applied component-wise, and for $\mathbf{v} = (v_1, ..., v_r) \in \mathbb{R}^r$, $\sigma_{\mathbf{v}} : \mathbb{R}^r \to \mathbb{R}^r$ denotes the shifted ReLU function with shift vector $\mathbf{v}$, defined by

$$\sigma_{\mathbf{v}}\big((y_1, \ldots, y_r)^\top \big) = \big(\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r) \big)^\top.$$

In this paper, we consider neural networks with sparsity parameter s, and we restrict the neural network functions to be uniformly bounded by a constant $F > 0$. Since the target function f_0 is supported on $[0,1]^d$ (see (1.3)), we define the class of neural network estimators $\mathcal{F}(L, \mathbf{p}, s, F)$ as follows:

$$\begin{aligned} \mathcal{F}(L, \mathbf{p}, s, F) := \Big\{ & f \mathbf{1}_{[0,1]^d} \,:\, f \in \mathcal{F}_{L,\mathbf{p}} \text{ such that} \\ & \max_{j=0,\ldots,L} \big(\|W_j\|_\infty \vee |\mathbf{v}_j|_\infty \big) \leq 1,\ \big\|f\big\|_\infty \leq F, \\ & \text{and } \sum_{j=0}^{L} \big(\|W_j\|_0 + |\mathbf{v}_j|_0 \big) \leq s \Big\}. \end{aligned} \tag{2.2}$$

When there is no ambiguity, we abbreviate $\mathcal{F}(L, \mathbf{p}, s, F)$ as $\mathcal{F}$ for simplicity, and we also use the notation $\mathcal{F}(L, \mathbf{p}, s) := \mathcal{F}(L, \mathbf{p}, s, \infty)$.

Estimator and Loss Functions

Recall that our training data $\mathcal{D}_N$ for this estimation problem consists of discrete observations from independent sample paths of solutions to (1.1), that is,

$$\mathcal{D}_N = \Big\{ \bar{X}^{(n)}_{t_0:t_M} = \big(\bar{X}^{(n)}_{t_0}, \ldots, \bar{X}^{(n)}_{t_M} \big),\ 1 \leq n \leq N \Big\}.$$

For a fixed n, let $\bar{X}^{(n),i}$ denote the i-th component of $\bar{X}^{(n)}$. The classical approach for estimating b^i considers the increment

$$Y^{(n)}_{t_m} := \frac{1}{\Delta} \left(\bar{X}^{(n),i}_{t_{m+1}} - \bar{X}^{(n),i}_{t_m} \right), \tag{2.3}$$

which is approximately equal to $b^i(\bar{X}^{(n)}_{t_m})$ plus a noise term (see Denis, Dion-Blanc, and Martinez (2021, Appendix C), Oga and Koike (2024, Equation (4.1)); see also (4.2)). Unlike in classical regression with neural networks (see Schmidt-Hieber (2020)), the errors in our setting are neither normally distributed nor independent of $b^i(\bar{X}^{(n)}_{t_m})$, which presents a key challenge in analyzing the estimator's performance.

Based on the above definition of $Y^{(n)}_{t_m}$, our neural network training procedure consists in minimizing

$$\mathcal{Q}_{\mathcal{D}_N}(f) := \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=0}^{M-1} \left(Y^{(n)}_{t_m} - f(\bar{X}^{(n)}_{t_m}) \right)^2 \tag{2.4}$$

over $f \in \mathcal{F}(L, \mathbf{p}, s, F)$. In practice, finding an exact minimizer of $\mathcal{Q}_{\mathcal{D}_N}(f)$ may be challenging. To quantify the discrepancy between our estimator $\hat{f} = \hat{f}_N$, which is the output of the neural network training procedure applied to the empirical loss $\mathcal{Q}_{\mathcal{D}_N}(f)$, and an exact minimizer of $\mathcal{Q}_{\mathcal{D}_N}(f)$ over $\mathcal{F}(L, \mathbf{p}, s, F)$, we define

$$\Psi^{\mathcal{F}}(\hat{f}) \coloneqq \mathbb{E}\left[\mathcal{Q}_{\mathcal{D}_N}(\hat{f}) - \inf_{f \in \mathcal{F}(L,\mathbf{p},s,F)} \mathcal{Q}_{\mathcal{D}_N}(f)\right]. \quad (2.5)$$

The performance of $\hat{f}$ is measured by the estimation risk

$$\mathcal{R}(\hat{f}, f_0) \coloneqq \mathbb{E}\left[\frac{1}{M}\sum_{m=0}^{M-1}\left(\widehat{f}(X_{t_m}) - f_0(X_{t_m})\right)^2\right]. \quad (2.6)$$

where $X = (X_t)_{t\in[0,T]}$ is a solution to (1.1) that is independent of the training set $\mathcal{D}_N$. In practice (see also the numerical experiments in Section 3), the estimation risk $\mathcal{R}(\hat{f}, f_0)$ is often estimated by the following empirical test error

$$\widetilde{\mathcal{R}}(\hat{f}, f_0) \coloneqq \frac{1}{MN'}\sum_{n=1}^{N'}\sum_{m=0}^{M-1}\left(\widehat{f}(\widetilde{X}_{t_m}^{(n)}) - f_0(\widetilde{X}_{t_m}^{(n)})\right)^2, \quad (2.7)$$

where $\widetilde{X}^{(n)} = (\widetilde{X}_t^{(n)})_{t\in[0,T]}$, $1 \le n \le N'$, are i.i.d. sample paths, independent of the training set $\mathcal{D}_N$.

Assumptions and Main Result

Throughout this paper, we impose the following two assumptions. Assumption 1 ensures the existence and uniqueness of the solution to the SDE (1.1), while Assumption 2 is a technical condition that guarantees the convergence result stated in Theorem 2.1.

Assumption 1. The initial random variable X_0 satisfies $\mathbb{E}[|X_0|^2] < +\infty$. The coefficient functions b and σ are globally Lipschitz continuous; that is, there exist constants $L_b, L_\sigma > 0$ such that for all $x, y \in \mathbb{R}^d$,

$$|b(x) - b(y)| \le L_b|x - y|, \quad |\sigma(x) - \sigma(y)| \le L_\sigma|x - y|.$$

Since b and σ are continuous, we define

$$C_b \coloneqq \sup_{x\in[0,1]^d}|b(x)| < \infty \text{ and } C_\sigma \coloneqq \sup_{x\in[0,1]^d}|\sigma(x)| < \infty.$$

It is obvious that $\|f_0\|_\infty \le C_b$.

Assumption 2. $F \ge \max(C_b, 1)$, $C_\sigma > 0$, $L \ge 1$, $s \ge 2$, $N \ge 2$ and $\Delta \le 1$.

Our main result is presented in the following theorem.

Theorem 2.1. *Suppose that Assumptions 1 and 2 hold. There exists a constant $\mathfrak{C}$, depending only on C_b, C_σ, L_b, L_σ, T, and a universal constant C (introduced later in Lemma 4.3), such that*

$$\mathcal{R}(\hat{f}, f_0) \le 4\Psi^{\mathcal{F}}(\hat{f}) + 6\inf_{f\in\mathcal{F}(L,\mathbf{p},s,F)}\mathcal{R}(f, f_0) + \mathfrak{C}F^2\left(\Delta + \frac{s(L\log s + \log d) + s\log 4F}{N} + s\frac{\log N}{N}\right).$$

Remark 2.2. Theorem 2.1 provides a decomposition of the upper bound on the estimation risk into three components. The first term reflects the error arising from the training procedure for finding a global minimizer with respect to the empirical loss $\mathcal{Q}_{\mathcal{D}_N}$. The second term can be bounded by the classical regression error for i.i.d. samples, as analyzed in Schmidt-Hieber (2020), and does not depend on the diffusion setting. Finally, the third term captures the error due to the temporal dependence inherent in the diffusion process.

Now let

$$\mathcal{C}_r^\beta(D, K) = \Big\{f : D \subset \mathbb{R}^r \to \mathbb{R} : \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta}\|\partial^{\boldsymbol{\alpha}} f\|_\infty + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor}\sup_{x,y\in D, x\ne y}\frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}} f(y)|}{|x-y|_\infty^{\beta-\lfloor\beta\rfloor}} \le K\Big\}.$$

Let $\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$ be the function space defined in Schmidt-Hieber (2020):

$$\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K) \coloneqq \big\{f = g_q \circ \dots g_0 : g_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}, g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, K), \text{ for some } |a_i|, |b_i| \le K\big\}.$$

with $\mathbf{d} \coloneqq (d_0, \dots, d_{q+1})$, $\mathbf{t} \coloneqq (t_0, \dots, t_q)$, $\boldsymbol{\beta} \coloneqq (\beta_0, \dots, \beta_q)$. Define $\beta_i^* \coloneqq \beta_i\prod_{l=i+1}^q(\beta_l \wedge 1)$ and

$$\phi_N \coloneqq \max_{0\le i\le q} N^{-\frac{2\beta_i^*}{2\beta_i^* + t_i}}. \quad (2.8)$$

Corollary 2.3. *Suppose that Assumptions 1 and 2 hold. Assume moreover that $f_0 \in \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$ and the neural network estimators set $\mathcal{F}(L, \mathbf{p}, s, F)$ satisfies*

(i) $F \ge \max(K, 1)$, $L \asymp \log_2 N$
(ii) $N\phi_N \lesssim \min_{i=1,\dots,L} p_i$, $s \asymp N\phi_N \log N$.

Then there exists a constant C depending on $q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, F, C_b, C_\sigma, L_b, L_\sigma, T$ such that if

$$\Delta \lesssim \phi_N \log^3 N \text{ and } \Psi^{\mathcal{F}}(\hat{f}) \le C\phi_N \log^3 N,$$

then $\mathcal{R}(\hat{f}, f_0) \le C\phi_N \log^3 N$.

3 Numerical Experiments

In this section, we evaluate the performance of our estimator using the following drift functions:

$$x \in \mathbb{R}^d \mapsto b(x) = -x + \phi\left(\tfrac{s(x)}{\theta}\right)\mathbf{1}_d, \quad (3.1)$$

where $\mathbf{1}_d = (1, \dots, 1)^\top$, $s(x) \coloneqq \sum_{i=1}^d x_i$, $\theta = 0.2$ and $\phi(z) \coloneqq 2z\exp(-z^2) - 8z\exp(-2z^2)$. In the following, Figure 3a *(True function)* illustrates $b(x)$ in one dimension. The initial condition X_0 is set as a random variable having standard normal distribution $\mathcal{N}(0, I_d)$, and the diffusion coefficient $\sigma(x)$ is set to the identity matrix.

The numerical study consists of three parts. First, we examine the convergence rate of our estimator with respect to N. Then, we compare its performance to the B-spline-based estimator proposed in Denis, Dion-Blanc, and Martinez (2021), focusing on two aspects: the convergence rate

and the ability to capture the local fluctuations induced by the function ϕ in the drift. Finally, we conclude this section with a summary of the comparative results, followed by additional remarks on the memory requirements and computational feasibility of our estimator compared to the B-spline-based approach.

Experimental Setup

Throughout this section, we fix the time horizon to $T = 1$ and the time step to $\Delta = 0.01$, and we estimate the first component of the drift coefficient $b_1(x) : \mathbb{R}^d \to \mathbb{R}$. We set the test set size $N' = 1000$, and the estimation risk is approximated using its empirical version defined in (2.7). The experiments are carried out for a range of dimensions $d \in \{1, 2, 10, 50\}$ and training size $N \in \{100, 200, 500, 1000, 2000, 5000\}$.

For the neural network estimator, we perform hyperparameter tuning for each combination of (N, d). We consider different hidden layer width

$$\mathbf{p} \in \{(d, 16, 16, 1), (d, 16, 32, 16, 1), (d, 16, 32, 32, 16, 1)\}.$$

The number of non-zero parameters s is set as a proportion s_{ratio} of the total number of parameters, with $s_{\text{ratio}} \in \{0.25, 0.5, 0.75\}$. For the training procedure, we use the Adam optimizer with a learning rate of 10^{-3}. The final number of training epochs is determined via early stopping triggered after 20 consecutive epochs.

Moreover, for each configuration, we repeat the experiment 50 times independently and compute the average error along with the corresponding 95% confidence interval. All implementations are performed using PyTorch.

Numerical Study of the Convergence Rate

We first examine how the estimation error evolves with the number of training trajectories N. In the setting of Corollary 2.3, we have $\phi_N = N^{-1}$ for the first component of the drift function b defined in (3.1). Hence, if $\Psi^{\mathcal{F}}(\hat{f}) \leq C\phi_N \log^3 N$, then the theoretical upper bound of $\mathcal{R}(\hat{f}, f_0)$ will be $CN^{-1} \log^3 N$ for some constant C.

Figure 1 shows the log–log plot of the estimation risk $\mathcal{R}(\hat{f}, f_0)$ as a function of N for $d \in \{1, 2, 10, 50\}$. The empirical convergence rate closely matches the theoretical upper bound for $d \in \{1, 2, 50\}$. For $d = 10$, however, the empirical error slightly exceeds the theoretical envelope for certain values of N. This deviation likely results from the restricted hyperparameter search space, which covered only a limited range of learning rates, training epochs, network widths and depths, and regularization parameters. However, the search grid was intentionally constrained to ensure comparability and to maintain a reasonable computational cost across dimensions.

Comparison with B-Spline-Base Estimator

We now compare our estimator with the B-spline-based estimator from Denis, Dion-Blanc, and Martinez (2021), using the same training and test datasets. The implementation of the B-spline estimator follows the procedure described in Denis, Dion-Blanc, and Martinez (2021), with an adaption to the multivariate setting using tensor-product B-splines basis (see e.g. Györfi et al. (2002)), and the number of knots selected based on validation performance.

Figure 2 demonstrates that our method achieves a better convergence rate of the estimation risk in N. Furthermore, as illustrated in Figure 3, our estimator ($\mathbf{p} = (d, 16, 32, 16, 1), s_{\text{ratio}} = 0.75$) more accurately captures the local variations of the drift function, while the B-spline-based method fails to recover these fine-scale features.

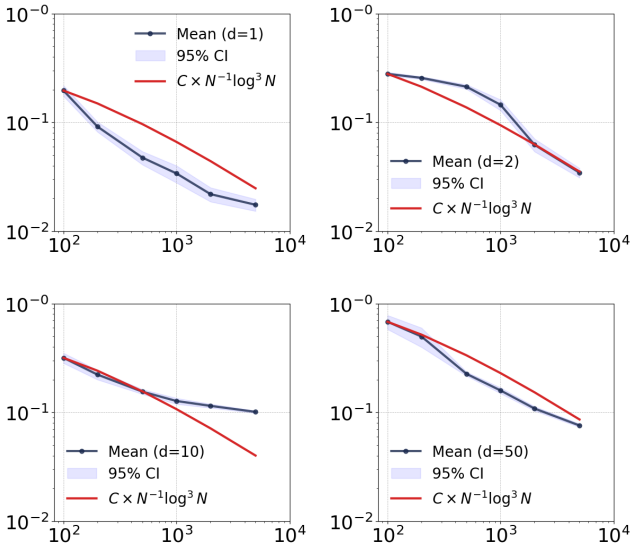


Figure 1: Convergence rates of our estimator for $d = 1$ (top left), $d = 2$ (top right), $d = 10$ (bottom left), and $d = 50$ (bottom right).

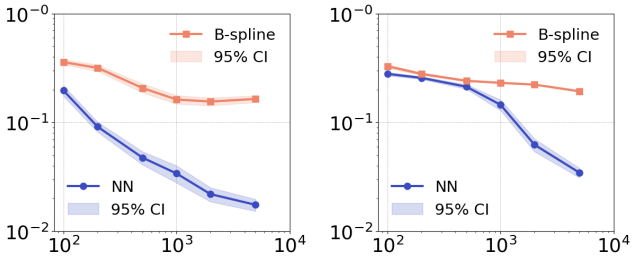
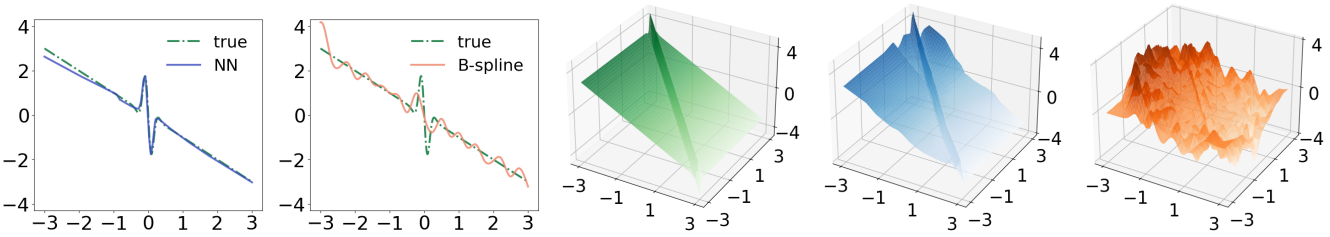


Figure 2: Comparison of convergence rates for $d = 1$ (left) and $d = 2$ (right).

Conclusion

In summary, Figures 2 and 3 clearly demonstrate that our neural network-based estimator outperforms the B-spline-based estimator both in terms of convergence rate and its ability to capture local fluctuations.

Furthermore, for the B-spline-based estimator, the number of basis functions grows exponentially with the dimension d, leading to a substantial increase in memory require-



(a) NN estimator, 1D (b) B-spline estimator, 1D (c) True function, 2D (d) NN estimator, 2D (e) B-spline estimator, 2D

Figure 3: Comparison of the ability to capture local fluctuations (N=5000).

ments. For example, when $d = 5$, assuming 8 basis functions per dimension, the resulting coefficient matrix has size $MN \times 32768$. If each entry is stored as a 64-bit float (8 bytes), this amounts to over 24GB of memory for $M = 100$ and $N = 1000$, not including the additional memory required for matrix inversion. As a result, the B-spline estimator becomes computationally infeasible on a standard laptop even for moderate values of d, for example $d \geq 3$. In contrast, the neural network estimator only requires storing the network parameters and minibatch data during training, making it far more scalable in high-dimensional settings.

4 Proof Sketch

The proof of Theorem 2.1 consists of two steps. In the first step (see Proposition 4.1), we establish an upper bound of the risk $\mathcal{R}(\hat{f}, f_0)$ defined in (2.6) by using the following expected empirical train risk $\hat{\mathcal{R}}_{\mathcal{D}_N}(\hat{f}, f_0)$, defined by

$$\hat{\mathcal{R}}_{\mathcal{D}_N}(\hat{f}, f_0) := \mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1}(\hat{f}(\bar{X}_{t_m}^{(n)}) - f_0(\bar{X}_{t_m}^{(n)}))^2\right], \quad (4.1)$$

and an estimation of the covering number of the neural network estimator set $\mathcal{F}(L, \mathbf{p}, s, F)$. Recall that for any $\delta > 0$, a subset $\mathcal{G} \subset \mathcal{F}$ is called a δ-net of $\mathcal{F}$ with respect to some norm $\|\cdot\|$ on $\mathcal{F}$, if for every $f \in \mathcal{F}$, there exists $g \in \mathcal{G}$ such that $\|f - g\| \leq \delta$. The minimal cardinality of such δ-net net is called the covering number, denoted by $\mathfrak{N}(\delta, \mathcal{F}, \|\cdot\|)$ (see e.g. van der Vaart and Wellner (2023)). When there is no ambiguity about the training set, we simplify the notation by writing $\hat{\mathcal{R}}(\hat{f}, f_0)$ instead of $\hat{\mathcal{R}}_{\mathcal{D}_N}(\hat{f}, f_0)$. In the second step (see Proposition 4.4), we derive an upper bound on the expected empirical training risk $\hat{\mathcal{R}}(\hat{f}, f_0)$ in terms of $\Psi^{\mathcal{F}}(\hat{f})$, the approximation error $\inf_{f\in\mathcal{F}}\|f - f_0\|_\infty^2$, the neural network parameters (s, L, F), and the data parameters Δ, N, and d.

Upper Bound of $\mathcal{R}(\hat{f}, f_0)$

This subsection is devoted to establishing the following proposition.

Proposition 4.1. *Suppose that Assumptions 1 and 2 hold. Fix an arbitrary $\bar{\varepsilon} \in (0, 1)$, we have*

$$\mathcal{R}(\hat{f}, f_0) \leq 3\bar{\varepsilon} + 2\hat{\mathcal{R}}(\hat{f}, f_0), + \frac{44F^2\log\left(Cs(L\log s + \log d + \log 4F - \log\bar{\varepsilon})\right)}{N},$$

where $C > 0$ is a universal constant.

The proof of Proposition 4.1 relies on the following two lemmas.

Lemma 4.2 (Lemma A.2 in Denis, Dion-Blanc, and Martinez (2021))**.** *Let $X_1, \ldots, X_N$ be independent copies of a random variable $X \in \mathcal{X}$. Let $\mathcal{G}$ be a class of real-valued functions on $\mathcal{X}$. For each $g \in \mathcal{G}$, and $x \in \mathcal{X}$, we assume that $0 \leq g(x) \leq L$, with $L > 0$. We consider $\mathcal{G}_\delta$ an δ-net of $\mathcal{G}$ w.r.t. $\|\cdot\|_{\sup}$ and we denote by $\mathfrak{N}(\delta, \mathcal{G}, \|\cdot\|_{\sup})$ its cardinality. Then, the following holds*

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}[g(X)] - \frac{2}{N}\sum_{i=1}^{N} g(X_i)\right)\right] \leq 3\varepsilon + \frac{11L\log\left(\mathfrak{N}(\delta, \mathcal{G}, \|\cdot\|_{\sup})\right)}{N}.$$

Lemma 4.3 (Lemma 4.13 in Oga and Koike (2024))**.** *Let $\mathcal{F} \subset \mathcal{F}(L, \mathbf{p}, s)$. If $s \geq 2$, we have for all $\delta \in (0, 1)$*

$$\log\mathfrak{N}(\delta, \mathcal{F}, \|\cdot\|_\infty) \leq Cs(L\log s + \log d - \log\delta)$$

where $C > 0$ is a universal constant.

Proof of Proposition 4.1. In what follows, we adopt the notation $X_{t_0:t_{M-1}}$ for $(X_{t_0}, \ldots, X_{t_{M-1}})$ and $\bar{X}_{t_0:t_{M-1}}^{(n)}$ for $(\bar{X}_{t_0}^{(n)}, \ldots, \bar{X}_{t_{M-1}}^{(n)})$. For a fixed $\bar{f} \in \mathcal{F}(L, \mathbf{p}, s, F)$, we define the function $g_{\bar{f}}$ from $(\mathbb{R}^d)^M$ to $\mathbb{R}$ by

$$x = (x_0, \ldots, x_{M-1}) \mapsto g_{\bar{f}}(x) = \frac{1}{M}\sum_{m=0}^{M-1}\left(\bar{f}(x_m) - f_0(x_m)\right)^2,$$

which is bounded above by $4F^2$ by Assumption 2 and supported on $([0, 1]^d)^M$ since both f_0 and $\bar{f}$ are supported on $[0, 1]^d$. Moreover, we define

$$\mathcal{G} := \{g_{\bar{f}} : \bar{f} \in \mathcal{F}(L, \mathbf{p}, s, F)\},$$

equipped with the norm $\|g\|_{\sup} := \sup_{x\in([0,1]^d)^M}|g(x)|$.

The estimation risk $\mathcal{R}(\hat{f}, f_0)$ can be decomposed by

$$\mathcal{R}(\hat{f}, f_0) \leq \mathbb{E}\left[g_{\hat{f}}(X_{t_0:t_{M-1}}) - \frac{2}{N}\sum_{n=1}^{N} g_{\hat{f}}\big(\bar{X}^{(n)}_{t_0:t_{M-1}}\big)\right] + 2\hat{\mathcal{R}}(\hat{f}, f_0),$$

where the first term can be upper bounded as follows

$$\begin{aligned}
&\mathbb{E}\left[g_{\hat{f}}(X_{t_0:t_{M-1}}) - \frac{2}{N}\sum_{n=1}^{N} g_{\hat{f}}\big(\bar{X}^{(n)}_{t_0:t_{M-1}}\big)\right] \\
&\leq \mathbb{E}\left[\sup_{g_{\bar{f}}\in\mathcal{G}}\left(g_{\bar{f}}(X_{t_0:t_{M-1}}) - \frac{2}{N}\sum_{n=1}^{N} g_{\bar{f}}\big(\bar{X}^{(n)}_{t_0:t_{M-1}}\big)\right)\right] \\
&\leq 3\varepsilon + \frac{44F^2\log\big(\mathfrak{N}(\varepsilon, \mathcal{G}, \|\cdot\|_{\sup})\big)}{N},
\end{aligned}$$

for any $\varepsilon \in (0,1)$ by applying Lemma 4.2.

We now provide an upper bound on the covering number $\mathfrak{N}(\varepsilon, \mathcal{G}, \|\cdot\|_{\sup})$ using the covering number over $\mathcal{F}(L, \mathbf{p}, s, F)$. Let $\mathfrak{N}_{\frac{\varepsilon}{4F}} \coloneqq \mathfrak{N}(\frac{\varepsilon}{4F}, \mathcal{F}(L, \mathbf{p}, s, F), \|\cdot\|_\infty)$ and let $\mathcal{F}_{\text{net}} \coloneqq \{f_1, ..., f_{\mathfrak{N}_{\frac{\varepsilon}{4F}}}\}$ be an $\frac{\varepsilon}{4F}$-net of $\mathcal{F}(L, \mathbf{p}, s, F)$. Then for every $\bar{f} \in \mathcal{F}(L, \mathbf{p}, s, F)$, there exists $\bar{f}_{\frac{\varepsilon}{4F}} \in \mathcal{F}_{\text{net}}$ such that $\|\bar{f} - \bar{f}_{\frac{\varepsilon}{4F}}\|_\infty \leq \frac{\varepsilon}{4F}$. It follows that

$$\|g_{\bar{f}} - g_{\bar{f}_{\frac{\varepsilon}{4F}}}\|_\infty \leq \|\bar{f} - \bar{f}_{\frac{\varepsilon}{4F}}\|_\infty \|\bar{f} + \bar{f}_{\frac{\varepsilon}{4F}} - 2f_0\|_\infty \leq \varepsilon.$$

Therefore, Lemma 4.3 implies that

$$\begin{aligned}
&\log\big(\mathfrak{N}(\varepsilon, \mathcal{G}, \|\cdot\|_{\sup})\big) \\
&\quad\leq \log\big(\mathfrak{N}(\frac{\varepsilon}{4F}, \mathcal{F}(L, \mathbf{p}, s, F), \|\cdot\|_\infty)\big) \\
&\quad\leq Cs(L\log s + \log d + \log 4F - \log\varepsilon).
\end{aligned}$$

Combining the results above, we get

$$\begin{aligned}
\mathcal{R}(\hat{f}, f_0) &\leq 3\varepsilon + 2\hat{\mathcal{R}}(\hat{f}, f_0) \\
&\quad + \frac{44F^2Cs(L\log s + \log d + \log 4F - \log\varepsilon)}{N}. \quad \square
\end{aligned}$$

Upper Bound of $\hat{\mathcal{R}}(\hat{f}, f_0)$

This section aims to establish an upper bound on the expected empirical training risk $\hat{\mathcal{R}}(\hat{f}, f_0)$. We abbreviate $\mathcal{F}$ for the neural network function class $\mathcal{F}(L, \mathbf{p}, s, F)$, and $\mathfrak{N}_\delta$ for the covering number $\mathfrak{N}(\delta, \mathcal{F}(L, \mathbf{p}, s, F), \|\cdot\|_\infty)$.

Proposition 4.4. *Suppose that Assumptions 1 and 2 hold. There exists a constant $\mathfrak{C}'$ depending on $C_b, C_\sigma, L_b, L_\sigma, T$ and the universal constant C in Lemma 4.3 such that*

$$\begin{aligned}
\hat{\mathcal{R}}(\hat{f}, f_0) &\leq 2\Psi^{\mathcal{F}}(\hat{f}) + 3\inf_{f\in\mathcal{F}}\mathcal{R}(f, f_0) \\
&\quad + \mathfrak{C}'\left(F^2\Delta + \frac{s(L\log s + \log d) + F}{N} + s\frac{\log N}{N}\right).
\end{aligned}$$

To prove Proposition 4.4, we first note that $Y^{(n)}_{t_m}$, defined in (2.3), admits the following decomposition :

$$\begin{aligned}
Y^{(n)}_{t_m} &= b^i(\bar{X}^{(n)}_{t_m}) \\
&\quad + \frac{1}{\Delta}\int_{t_m}^{t_{m+1}} (b^i(\bar{X}^{(n)}_s) - b^i(\bar{X}^{(n)}_{t_m}))\mathrm{d}s \\
&\quad + \frac{1}{\Delta}\int_{t_m}^{t_{m+1}} \sigma^i(\bar{X}^{(n)}_s)\mathrm{d}B^{(n)}_s \\
&=: b^i(\bar{X}^{(n)}_{t_m}) + I^{(n)}_{t_m} + \Sigma^{(n)}_{t_m}, \qquad (4.2)
\end{aligned}$$

where $B^{(n)}, 1 \leq n \leq N$ are i.i.d. d-dimensional standard Brownian motion. Given any two functions f_1, f_2 in $\mathcal{F}(L, \mathbf{p}, s, F)$, we define

$$\Psi_{\mathcal{D}_N}(f_1, f_2) \coloneqq \mathbb{E}\left[\mathcal{Q}_{\mathcal{D}_N}(f_1) - \mathcal{Q}_{\mathcal{D}_N}(f_2)\right]$$

with $\mathcal{Q}_{\mathcal{D}_N}$ defined in (2.4). Let $\bar{f}$ be an arbitrary function in $\mathcal{F}$. A direct computation yields the following decomposition of $\hat{\mathcal{R}}(\hat{f}, f_0)$:

$$\begin{aligned}
\hat{\mathcal{R}}(\hat{f}, f_0) &= \Psi_{\mathcal{D}_N}(\hat{f}, \bar{f}) \qquad (4.3) \\
&\quad + \mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1}(\bar{f}(\bar{X}^{(n)}_{t_m}) - f_0(\bar{X}^{(n)}_{t_m}))^2\right] \\
&\quad + 2\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1} I^{(n)}_{t_m}\Big(\hat{f}(\bar{X}^{(n)}_{t_m}) - \bar{f}(\bar{X}^{(n)}_{t_m})\Big)\right]. \\
&\quad + 2\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1} \Sigma^{(n)}_{t_m}\Big(\hat{f}(\bar{X}^{(n)}_{t_m}) - f_0(\bar{X}^{(n)}_{t_m})\Big)\right],
\end{aligned}$$

where the last term follows from the fact that $\bar{f}$ and f_0 are deterministic (i.e., independent of the training set), and $\mathbb{E}\left[\Sigma^{(n)}_{t_m}\right] = 0$ and independent of $\bar{X}^{(n)}_{t_m}$ for each m and n.

Let $\mathcal{F}_{\text{net}} \coloneqq \{f_1, \ldots, f_{\mathfrak{N}_\delta}\}$ denote a δ-net of $\mathcal{F}$ with respect to $\|\cdot\|_\infty$, Then, there exists a random index $\mathfrak{n}^*$ valued in $\{1, ..., \mathfrak{N}_\delta\}$ such that $|\hat{f} - f_{\mathfrak{n}^*}| \leq \delta$. A reasoning similar to Oga and Koike (2024, Lemma 4.8, 4.11, 4.12) leads to the following lemma.

Lemma 4.5. *Suppose that Assumptions 1 and 2 hold. Let $C_\star = 4(C_b^2 + C_\sigma^2)\exp(4(L_b^2 + L_\sigma^2))$.*

1. *For every $\bar{f} \in \mathcal{F}$ and $\varepsilon \in (0,1)$,*

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1} I^{(n)}_{t_m}\Big(\hat{f}(\bar{X}^{(n)}_{t_m}) - \bar{f}(\bar{X}^{(n)}_{t_m})\Big)\right] \\
&\quad\leq \frac{\varepsilon}{4}\hat{\mathcal{R}}(\hat{f}, f_0) + \frac{\varepsilon}{2}\hat{\mathcal{R}}(\bar{f}, f_0) + \frac{3L_b^2C_\star}{4\varepsilon}\Delta. \qquad (4.4)
\end{aligned}$$

2. *There exists a constant $\bar{C}_\sigma$ depending only on C_σ such that for every $n \in \{1, ..., N\}$,*

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{MN}\sum_{n=1}^{N}\sum_{m=0}^{M-1} \Sigma^{(n)}_{t_m}\big(\hat{f} - f_{\mathfrak{n}^*}\big)(\bar{X}^{(n)}_{t_m})\right] \\
&\quad\leq \delta\sqrt{\frac{1}{2}L_\sigma^2C_\star} + \frac{\bar{C}_\sigma}{\sqrt{T}}\int_0^\delta \sqrt{\log\mathfrak{N}_u}\,\mathrm{d}u. \qquad (4.5)
\end{aligned}$$

Furthermore, the proof of Proposition 4.4 relies on the following lemma.

Lemma 4.6. *Suppose that Assumptions 1 and 2 hold. For every $\varepsilon \in (0,1)$, we have*

$$\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1}\Sigma_{t_m}^{(n)}\big(f_{\mathfrak{n}^*}(\bar{X}_{t_m}^{(n)}) - f_0(\bar{X}_{t_m}^{(n)})\big)\right] \leq \frac{\varepsilon}{4}\hat{\mathcal{R}}(\hat{f}, f_0) + \gamma_\varepsilon,$$

where $\gamma_\varepsilon = \varepsilon F\delta + \varepsilon\frac{F^2L_\sigma^2C_\star}{C_\sigma^2}\Delta + \frac{4C_\sigma^2}{\varepsilon TN}\big(\log 2 + 2\log\mathfrak{N}_\delta\big)$.

Proof sketch of Lemma 4.6. For a fixed function $f \in \mathcal{F}$ and for a fixed n, we define the processes $\widehat{M}^{(n)}(f)$, $\widehat{A}^{(n)}(f)$, $\overline{M}(f)$ and $\bar{A}(f)$ by

$$\widehat{M}^{(n)}(f)_s \coloneqq \sum_{m=0}^{M-1}(f-f_0)(\bar{X}_{t_m}^{(n)})\int_{s\wedge t_m}^{s\wedge t_{m+1}}\sigma^i\big(\bar{X}_u^{(n)}\big)\mathrm{d}B_u^{(n)},$$

$$\overline{M}(f)_s \coloneqq \frac{1}{N}\sum_{n=1}^{N}\widehat{M}^{(n)}(f),$$

$$\widehat{A}^{(n)}(f) \coloneqq \langle\widehat{M}^{(n)}(f)\rangle,$$

$$\bar{A}(f) \coloneqq \big\langle\overline{M}(f)\big\rangle = \frac{1}{N^2}\sum_{n=1}^{N}\widehat{A}^{(n)}(f).$$

It follows that

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1}\Sigma_{t_m}^{(n)}(f_{\mathfrak{n}^*}(\bar{X}_{t_m}^{(n)}) - f_0(\bar{X}_{t_m}^{(n)}))\right]\\
&\leq \mathbb{E}\left[\left|\frac{1}{T}\overline{M}(f_{\mathfrak{n}^*})_T\right|\right] = \tfrac{2}{T}\mathbb{E}\left[\left|\overline{\xi}_{f_{\mathfrak{n}^*}}\right|\big(\bar{A}(f_{\mathfrak{n}^*})_T + \overline{D}_{f_{\mathfrak{n}^*}}\big)^{\frac{1}{2}}\right]\\
&\leq \tfrac{2}{T}\sqrt{\mathbb{E}\left[\overline{\xi}_{f_{\mathfrak{n}^*}}^2\right]\big(2\mathbb{E}\left[\bar{A}(f_{\mathfrak{n}^*})_T\right] + \varepsilon'\big)}, \qquad (4.6)
\end{aligned}$$

where for every $f \in \mathcal{F}$, we denote by $\overline{\xi}_f \coloneqq \frac{\overline{M}(f)_T}{2\sqrt{\bar{A}(f)_T + \overline{D}_f}}$ and $\overline{D}_f = \mathbb{E}\left[\bar{A}(f)_T\right] + \varepsilon'$ for some $\varepsilon' > 0$. As $\overline{M}(f)$ is a continuous local martingale for every $f \in \mathcal{F}$, it follows from Oga and Koike (2024, Lemma 4.9) that

$$\mathbb{E}\left[\frac{\sqrt{\overline{D}_f}}{\sqrt{\bar{A}(f_n)_T + \overline{D}_f}}\exp\left\{2\overline{\xi}_f^2\right\}\right] \leq 1. \qquad (4.7)$$

Hence, an argument analogous to Oga and Koike (2024, Lemma 4.10) yields $\mathbb{E}\left[\overline{\xi}_{f_{n^*}}^2\right] \leq \frac{1}{4}\log 2 + \frac{1}{2}\log\mathfrak{N}_\delta$.

For $\mathbb{E}\left[\bar{A}(f_{\mathfrak{n}^*})_T\right]$, we have

$$\mathbb{E}\left[\bar{A}(f_{\mathfrak{n}^*})_T\right] = \frac{1}{N}\cdot\mathbb{E}\Big[\frac{1}{N}\sum_{n=1}^{N}\widehat{A}^{(n)}(f_{\mathfrak{n}^*})_T\Big],$$

and an argument analogous to Oga and Koike (2024, Lemma 4.10) yields

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\widehat{A}^{(n)}(f_{\mathfrak{n}^*})_T\right]\\
&\leq 2C_\sigma^2T\big(\hat{\mathcal{R}}(\hat{f}, f_0) + 4F\delta\big) + 8F^2L_\sigma^2C_\star T\Delta.
\end{aligned}$$

Finally, inserting the above bounds into (4.6) yields

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=0}^{M-1}\Sigma_{t_m}^{(n)}(f_{\mathfrak{n}^*}(\bar{X}_{t_m}^{(n)}) - f_0(\bar{X}_{t_m}^{(n)}))\right]\\
&\leq \frac{1}{T}\sqrt{\log 2 + 2\log\mathfrak{N}_\delta}\\
&\quad\cdot\sqrt{\frac{4}{N}\left[C_\sigma^2T\big(\hat{\mathcal{R}}(\hat{f}, f_0) + 4F\delta\big) + 4F^2L_\sigma^2C_\star T\Delta\right] + \varepsilon'}.
\end{aligned}$$

By letting $\varepsilon' \to 0$ and by applying the AM-GM inequality $\sqrt{xy} \leq \frac{\varepsilon}{4}x + \frac{1}{\varepsilon}y$, we get the desired inequality. □

Proof sketch of Proposition 4.4. By inserting the results of Lemmas 4.5 and 4.6 into (4.3), we obtain, for a an arbitrary function $\bar{f}$ in $\mathcal{F}$, for $\varepsilon \in (0,1)$,

$$\hat{\mathcal{R}}(\hat{f}, f_0) \leq \frac{\Psi^{\mathcal{F}}(\hat{f})}{1-\varepsilon} + \frac{1+\varepsilon}{1-\varepsilon}\mathcal{R}(\bar{f}, f_0) + \frac{\gamma'_\varepsilon}{1-\varepsilon},$$

where we apply $\hat{\mathcal{R}}(\bar{f}, f_0) = \mathcal{R}(\bar{f}, f_0)$ and

$$\begin{aligned}
\gamma'_\varepsilon =& \frac{3L_b^2C_\star}{2\varepsilon}\Delta + \delta\sqrt{2L_\sigma^2C_\star} + \frac{2\bar{C}_\sigma}{\sqrt{T}}\int_0^\delta\sqrt{\log\mathfrak{N}_u}\,\mathrm{d}u\\
&+ 2\varepsilon F\delta + 2\varepsilon\frac{F^2L_\sigma^2C_\star}{C_\sigma^2}\Delta + \frac{8C_\sigma^2}{\varepsilon TN}\big(\log 2 + 2\log\mathfrak{N}_\delta\big).
\end{aligned}$$

By applying Lemma 4.3, we have

$$\begin{aligned}
\int_0^\delta\sqrt{\log\mathfrak{N}_u}\,\mathrm{d}u \leq& \int_0^\delta\sqrt{Cs(L\log s + \log d)}\,\mathrm{d}u\\
&+ \sqrt{Cs}\int_0^\delta\sqrt{-\log u}\,\mathrm{d}u.
\end{aligned}$$

Using the change of variable $t = -\log u$ and the inequality $\Gamma\big(\frac{3}{2}, -\log(\delta)\big) \leq \frac{3}{4}\delta\sqrt{-\log\delta}$ valid for $\delta < \exp(-\frac{1}{2})$, we set $\varepsilon = \frac{1}{2}$ and $\delta = \frac{1}{N}$ to obtain

$$\begin{aligned}
\hat{\mathcal{R}}(\hat{f}, f_0) \leq& 2\Psi^{\mathcal{F}}(\hat{f}) + 3\mathcal{R}(\bar{f}, f_0)\\
&+ 2L_\sigma^2C_\star\big(3 + \tfrac{F^2}{{C_\sigma}^2}\big)\Delta + C_1\frac{1}{N} + C_2\frac{\log N}{N},
\end{aligned}$$

with

$$\begin{aligned}
C_1 =& 2\sqrt{2L_\sigma^2C_\star} + \frac{4\bar{C}_\sigma}{\sqrt{T}}\sqrt{Cs(L\log s + \log d)} + 2F\\
&+ \frac{32C_\sigma^2}{T}\big(\log 2 + 2Cs(L\log s + \log d)\big)
\end{aligned}$$

and $C_2 = \frac{6\bar{C}_\sigma\sqrt{Cs}}{\sqrt{T}} + \frac{64C_\sigma^2Cs}{T}$. We conclude by noting that $\mathcal{R}(\bar{f}, f_0) \leq \|\bar{f} - f_0\|_\infty$, and that the above inequality holds for any arbitrary function $\bar{f} \in \mathcal{F}$. □

Proof of Theorem 2.1. Theorem 2.1 follows directly from Propositions 4.1 and 4.4 by setting $\bar{\varepsilon} = \frac{\log N}{N}$. □

Proof of Corollary 2.3. Theorem 1 in Schmidt-Hieber (2020) provides an upper bound of $\inf_{f\in\mathcal{F}(L,\mathbf{p},s,F)}\mathcal{R}(f, f_0)$. Corollary 2.3 then follows directly from Theorem 2.1 and Theorem 1 in Schmidt-Hieber (2020). □

Acknowledgments

This work was supported by the French National Research Agency (ANR) under the France 2030 program, reference "ANR-23-EXMA-0011" (MIRTE Project), and originates from the Master's thesis of Yuzhen Zhao, completed during her internship at the LEDa laboratory, Université Paris Dauphine–PSL, under the supervision of Yating Liu.

References

Bae, Y.; Ha, S.; and Jeong, H. 2025. Inferring the Langevin equation with uncertainty via Bayesian neural networks. *Chaos, Solitons & Fractals*, 197: 116440.

Bressloff, P. C. 2014. *Stochastic processes in cell biology*, volume 41 of *Interdisciplinary Applied Mathematics*. Springer, Cham. ISBN 978-3-319-08487-9; 978-3-319-08488-6.

Comte, F.; and Genon-Catalot, V. 2020. Nonparametric drift estimation for i.i.d. paths of stochastic differential equations. *Ann. Statist.*, 48(6): 3336–3365.

Comte, F.; Genon-Catalot, V.; and Rozenholc, Y. 2007. Penalized nonparametric mean square estimation of the coefficients of diffusion processes. *Bernoulli*, 13(2): 514–543.

Dalalyan, A. 2005. Sharp adaptive estimation of the drift function for ergodic diffusions. *Ann. Statist.*, 33(6): 2507–2528.

Denis, C.; Dion-Blanc, C.; and Martinez, M. 2021. A ridge estimator of the drift from discrete repeated observations of the solution of a stochastic differential equation. *Bernoulli*, 27(4): 2675–2713.

Frishman, A.; and Ronceray, P. 2020. Learning Force Fields from Stochastic Trajectories. *Phys. Rev. X*, 10: 021009.

Gao, T.-T.; Barzel, B.; and Yan, G. 2024. Learning interpretable dynamics of stochastic complex systems from experimental data. *Nature communications*, 15(1): 6029.

Gardiner, C. W. 2004. *Handbook of stochastic methods for physics, chemistry and the natural sciences*, volume 13 of *Springer Series in Synergetics*. Springer-Verlag, Berlin, third edition. ISBN 3-540-20882-8.

Györfi, L.; Kohler, M.; Krzyżak, A.; and Walk, H. 2002. *A distribution-free theory of nonparametric regression*. Springer Series in Statistics. Springer-Verlag, New York. ISBN 0-387-95441-4.

Hoffmann, M. 1999. Adaptive estimation in diffusion processes. *Stochastic Process. Appl.*, 79(1): 135–163.

Karatzas, I.; and Shreve, S. E. 1998. *Methods of mathematical finance*, volume 39 of *Applications of Mathematics (New York)*. Springer-Verlag, New York. ISBN 0-387-94839-2.

Kutoyants, Y. A. 2004. *Statistical inference for ergodic diffusion processes*. Springer Series in Statistics. Springer-Verlag London, Ltd., London. ISBN 1-85233-759-1.

Oga, A.; and Koike, Y. 2024. Drift estimation for a multidimensional diffusion process using deep neural networks. *Stochastic Process. Appl.*, 170: Paper No. 104240, 25.

Ren, Y.; Lu, Y.; Ying, L.; and Rotskoff, G. M. 2024. Statistical spatially inhomogeneous diffusion inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14820–14828.

Schmidt-Hieber, J. 2020. Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Statist.*, 48(4): 1875–1897.

van der Vaart, A. W.; and Wellner, J. A. 2023. *Weak convergence and empirical processes—with applications to statistics*. Springer Series in Statistics. Springer, Cham. ISBN 978-3-031-29038-1; 978-3-031-29040-4. Second edition [of 1385671].

Yarotsky, D. 2017. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94: 103–114.

Yildiz, C.; Heinonen, M.; Intosalmi, J.; Mannerstrom, H.; and Lahdesmaki, H. 2018. Learning Stochastic Differential Equations with Gaussian Processes without Gradient Matching. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6.