CIRJE-F- 1212

Solving Kolmogorov PDEs without the Curse of Dimensionality via Deep Learning and Asymptotic Expansion with Malliavin Calculus

Akihiko Takahashi
The University of Tokyo

Toshihiro Yamada
Hitotsubashi University
Japan Science and Technology Agency (JST)

May 2023

Solving Kolmogorov PDEs without the curse of dimensionality via deep learning and asymptotic expansion with Malliavin calculus*

Akihiko Takahashi† and Toshihiro Yamada‡§

First version¶: May 2021, This version : May 9, 2023

Abstract

This paper proposes a new spatial approximation method without the curse of dimensionality for solving high-dimensional partial differential equations (PDEs) by using an asymptotic expansion method with a deep learning-based algorithm. In particular, the mathematical justification on the spatial approximation is provided. Numerical examples for high-dimensional Kolmogorov PDEs show effectiveness of our method.

Keywords. Asymptotic expansion, Deep learning, Kolmogorov PDEs, Malliavin calculus, Curse of dimensionality

1 Introduction

Recently, for solving high-dimensional partial differential equations (PDEs), deep learning-based algorithms have been actively proposed (see [2][3] for instance). Moreover, a number of papers for mathematical justification on the deep learning-based spatial approximations have appeared, where the authors demonstrate that deep neural networks overcome the curse of dimensionality in approximations of high-dimensional PDEs. For the related literature, see [4][5][6][11][19] for example. In particular, these works treat some specific forms of PDEs such as high-dimensional heat equations or Kolmogorov PDEs with constant diffusion and nonlinear drift coefficient. Also, integral kernels are assumed to have explicit forms for justification of the spatial approximations for solutions to high-dimensional PDEs.

However, most high-dimensional PDEs may not have explicit integral forms in practice. In other words, integral forms of solutions themselves should be approximated by a certain method.

In the current paper, we give a new spatial approximation using an asymptotic expansion method with a deep learning-based algorithm for solving high-dimensional PDEs without the curse of dimensionality. More precisely, we follow approaches given in [40] and the literature such as [8][17][18][23][24][26][27][30][32][33][35][38][39][41][43]. Particularly, we provide a uniform error estimate for the asymptotic expansion for solutions of Kolmogorov PDEs with nonlinear coefficients, motivated by the works of [2][11][31]. For a solution to a d-dimensional Kolmogorov PDE with a small parameter λ, namely $u_\lambda : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ given by $u_\lambda(t,x) = E[f(X_t^{\lambda,x})]$ for

*Forthcoming in "*Partial Differential Equations and Applications*"

†The University of Tokyo, Tokyo, Japan

‡Hitotsubashi University, Tokyo, Japan (e-mail: toshihiro.yamada@r.hit-u.ac.jp)

§Japan Science and Technology Agency (JST), Tokyo, Japan

¶The title of the first version of the paper is "Asymptotic expansion and deep neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with nonlinear coefficients" (CIRJE-F-1167 and CARF-F-547).

$(t,x) \in [0,T]\times\mathbb{R}^d$ where $\{X_t^{\lambda,x}\}_{t\geq 0}$ is a d-dimensional diffusion process starting from x, we justify the following spatial approximation on a range $[a,b]^d$:

$$u_\lambda(t,\cdot) \approx \text{“high-dimensional asymptotic expansion”} \ E[f(\bar{X}_t^{\lambda,\cdot})\mathcal{M}_t^{\lambda,\cdot}] \tag{1.1}$$
$$\approx \text{“deep neural network approximation”} \ \mathcal{R}(\phi)(\cdot), \tag{1.2}$$

by applying an appropriate neural network ϕ. Here, for $t>0$ and $x\in\mathbb{R}^d$, $\bar{X}_t^{\lambda,x}$ is a certain Gaussian random variable and $\mathcal{M}_t^{\lambda,x}$ is a stochastic weight for the expansion given based on Malliavin calculus. In order to chose the network ϕ, the analysis of “product of neural networks” and a dimension analysis of asymptotic expansion with Malliavin calculus are crucial in our approach. We show a precise error estimate for the approximation (1.1) and prove that the complexity of the neural network grows at most polynomially in the dimension d and the reciprocal of the precision ε of the approximation (1.2). Moreover, we give an explicit form of the asymptotic expansion in (1.1) and show numerical examples to demonstrate effectiveness of the proposed scheme for high-dimensional Kolmogorov PDEs.

The organization of the paper is as follows. Section 2 is dedicated to notation, definitions and preliminary results on deep learning and Malliavin calculus. Section 3 provides the main result, namely, the deep learning-based asymptotic expansion for solving Kolmogorov PDEs. The proof is shown in Section 4. Section 5 introduces the deep learning implementation. Various numerical examples are shown in Section 6. The useful lemmas on Malliavin calculus and ReLU calculus are summarized, and furthermore the sample code is listed in Appendix.

2 Preliminaries

We first prepare notation. For $d\in\mathbb{N}$ and for a vector $x\in\mathbb{R}^d$, we denote by $\|x\|$ the Euclidean norm. Also, for $k,\ell\in\mathbb{N}$ and for a matrix $A\in\mathbb{R}^{k\times\ell}$, we denote by $\|A\|$ the Frobenius norm. For $d\in\mathbb{N}$, let I_d be the identity matrix. For $m,k,\ell\in\mathbb{N}$, let $C(\mathbb{R}^m,\mathbb{R}^{k\times\ell})$ (resp., $C([0,T]\times\mathbb{R}^m,\mathbb{R}^{k\times\ell})$) be the set of continuous functions $f:\mathbb{R}^k\to\mathbb{R}^{k\times\ell}$ (resp., $f:[0,T]\times\mathbb{R}^m\to\mathbb{R}^{k\times\ell}$) and $C_{Lip}(\mathbb{R}^m,\mathbb{R}^{k\times\ell})$ be the set of Lipschitz continuous functions $f:\mathbb{R}^m\to\mathbb{R}^{k\times\ell}$. Also, we define $C_b^\infty(\mathbb{R}^m,\mathbb{R}^\ell)$ as the set of smooth functions $f:\mathbb{R}^m\to\mathbb{R}^{k\times\ell}$ with bounded derivatives of all orders. For a multi-index α, let $|\alpha|$ be the length of α. For a bounded function $f:\mathbb{R}^m\to\mathbb{R}^{k\times\ell}$, we define $\|f\|_\infty=\sup_{x\in\mathbb{R}^m}\|f(x)\|$. For $m,k,\ell\in\mathbb{N}$, for a function $f\in C_{Lip}(\mathbb{R}^m,\mathbb{R}^{k\times\ell})$, we denote by $C_{Lip}[f]$ the Lipschitz continuous constant. For $d\in\mathbb{N}$ and for a smooth function $f:\mathbb{R}^d\to\mathbb{R}$, we define $\partial_i f=\frac{\partial}{\partial x_i}f$ for $i=1,\dots,d$, moreover we define $\partial^\alpha f=\partial_{\alpha_1}\cdots\partial_{\alpha_k}f$ for $\alpha=(\alpha_1,\dots,\alpha_k)\in\{1,\dots,d\}^k$, $k\in\mathbb{N}$. For $a,b\in\mathbb{R}$, we may write $a\vee b=\max\{a,b\}$.

2.1 Deep neural networks

Let us prepare notation and definitions for deep neural networks. Let $\mathcal{N}$ be the set of deep neural networks (DNNs):

$$\mathcal{N}=\cup_{L\in\mathbb{N}\cap[2,\infty)}\cup_{(N_0,N_1,\dots,N_L)\in\mathbb{N}^{L+1}}\mathcal{N}_L^{N_0,N_1,\dots,N_L}, \tag{2.1}$$

where $\mathcal{N}_L^{N_0,N_1,\dots,N_L}=\times_{\ell=1}^L(\mathbb{R}^{N_\ell\times N_{\ell-1}}\times\mathbb{R}^{N_\ell})$.

Let $\varrho\in C(\mathbb{R},\mathbb{R})$ be an activation function, and for $k\in\mathbb{N}$, define $\varrho_k(x)=(\varrho(x_1),\dots,\varrho(x_k))$, $x\in\mathbb{R}^k$.

We define $\mathcal{R}:\mathcal{N}\to\cup_{m,n\in\mathbb{N}}C(\mathbb{R}^m,\mathbb{R}^n)$, $\mathcal{C}:\mathcal{N}\to\mathbb{N}$, $\mathcal{L}:\mathcal{N}\to\mathbb{N}$, $\dim_{\rm in}:\mathcal{N}\to\mathbb{N}$ and $\dim_{\rm out}:\mathcal{N}\to\mathbb{N}$ as follows:

For $L\in\mathbb{N}\cap[2,\infty)$, $N_0,\dots,N_L\in\mathbb{N}$, $\psi=((W_1,B_1),\dots,(W_L,B_L))\in\mathcal{N}_L^{N_0,N_1,\dots,N_L}$, let $\mathcal{L}(\psi)=L$, $\dim_{\rm in}(\psi)=N_0$, $\dim_{\rm out}(\psi)=N_L$, $\mathcal{C}(\psi)=\sum_{\ell=1}^L N_\ell(N_{\ell-1}+1)$, and

$$\mathcal{R}(\psi)(\cdot)=\mathcal{A}_{W_L,B_L}\circ\varrho_{N_{L-1}}\circ\mathcal{A}_{W_{L-1},B_{L-1}}\circ\cdots\circ\varrho_{N_1}\circ\mathcal{A}_{W_1,B_1}(\cdot)\in C(\mathbb{R}^{N_0},\mathbb{R}^{N_L}), \tag{2.2}$$

where $\mathcal{A}_{W_k,B_k}(x)=W_kx+B_k$, $x\in\mathbb{R}^{N_{k-1}}$, $k=1,\dots,L$.

2.2 Malliavin calculus

We prepare basic notation and definitions on Malliavin calculus following Bally (2003) [1] Ikeda and Watanabe (1989) [16], Malliavin (1997) [25], Malliavin and Thalmaier (2006) [26] and Nualart (2006) [29].

Let $\Omega^d = \{\omega : [0,T] \to \mathbb{R}^d;\ \omega \text{ is continuous},\ \omega(0) = 0\}$, $H^d = L^2([0,T], \mathbb{R}^d)$ and let μ^d be the Wiener measure on $(\Omega^d, \mathcal{B}(\Omega^d))$, where $\mathcal{B}(\Omega^d)$ is the Borel σ-field induced by the topology of the uniform convergence on $[0,T]$. We call (Ω^d, H^d, μ^d) the d-dimensional Wiener space. For a Hilbert space V with the norm $\|\cdot\|_V$ and $p \in [1,\infty)$, the L^p-space of V-valued Wiener functionals is denoted by $L^p(\Omega^d, V)$, that is, $L^p(\Omega^d, V)$ is a real Banach space of all μ^d-measurable functionals $F : \Omega^d \to V$ such that $\|F\|_p = E[\|F\|_V^p]^{1/p} < \infty$ with the identification $F = G$ if and only if $F(\omega) = G(\omega)$, a.s. When $V = \mathbb{R}$, we write $L^p(\Omega^d)$. For a real separable Hilbert space V and $F : \Omega^d \to V$, we write $\|F\|_{p,V} = E[\|F\|_V^p]^{1/p}$, in particular, $\|F\|_p$ when $V = \mathbb{R}$. Let $B^d = \{B_t^d\}_t$ be a coordinate process defined by $B_t^d(\omega) = \omega(t)$, $\omega \in \Omega^d$, i.e. B^d is a d-dimensional Brownian motion, and $B^d(h)$ be the Wiener integral $B^d(h) = \sum_{j=1}^d \int_0^T h^j(s) dB_s^{d,j}$ for $h \in H^d$.

Let $\mathscr{S}(\Omega^d)$ denote the class of smooth random variables of the form $F = f(B^d(h_1), \dots, B^d(h_n))$ where $f \in C_b^\infty(\mathbb{R}^n, \mathbb{R})$, $h_1, \dots, h_n \in H^d$, $n \geq 1$. For $F \in \mathscr{S}(\Omega^d)$, we define the derivative DF as the H-valued random variable $DF = \sum_{j=1}^n \partial_j f(B^d(h_1), \dots, B^d(h_n))h_j$, which is regarded as the stochastic process:

$$D_{i,t}F = \textstyle\sum_{j=1}^n \partial_j f(B^d(h_1), \dots, B^d(h_n))h_j^i(t), \quad i = 1, \dots, d, \quad t \in [0,T]. \tag{2.3}$$

For $F \in \mathscr{S}(\Omega^d)$ and $j \in \mathbb{N}$, we set $D^j F$ as the $(H^d)^{\otimes j}$-valued random variable obtained by the j-times iteration of the operator D. For a real separable Hilbert space V, consider $\mathscr{S}_V$ of V-valued smooth Wiener functionals of the form $F = \sum_{i=1}^\ell F_i v_i$, $v_i \in V$, $F_i \in \mathscr{S}(\Omega^d)$, $i \leq \ell$, $\ell \in \mathbb{N}$. Define $D^j F = \sum_{i=1}^\ell D^j F_i \otimes v_i$, $j \in \mathbb{N}$. Then for $j \in \mathbb{N}$, D^j is a closable operator from $\mathscr{S}_V$ into $L^p(\Omega^d, (H^d)^{\otimes j} \otimes V)$ for any $p \in [1,\infty)$ (see p.31 of Nualart (2006) [29]). For $k \in \mathbb{N}$, $p \in [1,\infty)$, we define $\|F\|_{k,p,V}^p = E[\|F\|_V^p] + \sum_{j=1}^k E[\|D^j F\|_{(H^d)^{\otimes j} \otimes V}^p]$, $F \in \mathscr{S}_V$. Then, the space $\mathbb{D}^{k,p}(\Omega^d, V)$ is defined as the completion of $\mathscr{S}_V$ with respect to the norm $\|\cdot\|_{k,p,V}$. Moreover, let $\mathbb{D}^\infty(\Omega^d, V)$ be the space of smooth Wiener functionals in the sense of Malliavin $\mathbb{D}^\infty(\Omega^d, V) = \cap_{p \geq 1} \cap_{k \in \mathbb{N}} \mathbb{D}^{k,p}(\Omega^d, V)$. We write $\mathbb{D}^{k,p}(\Omega^d)$, $k \in \mathbb{N}$, $p \in [1,\infty)$ and $\mathbb{D}^\infty(\Omega^d)$, when $V = \mathbb{R}$. Let δ be an unbounded operator from $L^2(\Omega^d, H^d)$ into $L^2(\Omega^d)$ such that the domain of δ, denoted by $\mathrm{Dom}(\delta)$, is the set of H^d-valued square integrable random variables u such that $|E[\langle DF, u\rangle_{H^d}]| \leq c\|F\|_{1,2}$ for all $F \in \mathbb{D}^{1,2}(\Omega^d)$ where c is some constant depending on u, and if $u \in \mathrm{Dom}(\delta)$, there exists $\delta(u) \in L^2(\Omega^d)$ satisfying

$$E[\langle DF, u\rangle_{H^d}] = E[F\delta(u)] \tag{2.4}$$

for any $F \in \mathbb{D}^{1,2}(\Omega^d)$. For $u = (u^1, \dots, u^d) \in \mathrm{Dom}(\delta)$, $\delta(u) = \sum_{i=1}^d \delta^i(u^i)$ is called the Skorohod integral of u, and it holds that $E[\int_0^T D_{i,s}F u_s^i ds] = E[F\delta^i(u^i)]$, $i = 1, \dots, d$ for all $F \in \mathbb{D}^{1,2}$ (see Proposition 6 of Bally (2003) [1]). For all $k \in \mathbb{N} \cup \{0\}$ and $p > 1$, the operator δ is continuous from $\mathbb{D}^{k+1,p}(\Omega^d, H^d)$ into $\mathbb{D}^{k,p}(\Omega^d)$ (see Proposition 1.5.7 of Nualart (2006) [29]). For $G \in \mathbb{D}^{1,2}(\Omega^d)$ and $h \in \mathrm{Dom}(\delta)$ such that $Gh \in L^2(\Omega^d, H^d)$, it holds that

$$\delta^i(Gh^i) = G\delta^i(h^i) - \int_0^T D_{i,s}G h_s^i ds, \quad i = 1, \dots, d, \tag{2.5}$$

and in particular, if $h \in \mathrm{Dom}(\delta)$ is an adapted process, $\delta^i(h^i)$ is given by the Itô integral, i.e. $\delta^i(h^i) = \int_0^T h_s^i dB_s^{d,i}$ for $i = 1, \dots, d$ (e.g. see Section 3.1.1 of Bally (2003) [1], Proposition 1.3.3 and Proposition 1.3.11 of Nualart (2006) [29]).

For $F = (F^1, \dots, F^d) \in (\mathbb{D}^\infty(\Omega^d))^d$, define the Malliavin covariance matrix of F, $\sigma^F = (\sigma_{ij}^F)_{1 \leq i,j \leq d}$, by $\sigma_{ij}^F = \langle DF^i, DF^j\rangle_{H^d} = \sum_{k=1}^d \int_0^T D_{k,s}F^i D_{k,s}F^j ds$, $1 \leq i,j \leq d$. We say that $F \in (\mathbb{D}^\infty(\Omega^d))^d$ is nondegenerate if the matrix σ^F is invertible *a.s.* and satisfies $\|(\det \sigma^F)^{-1}\|_p < \infty$, $p > 1$. Malliavin's theorem claims that if $F \in (\mathbb{D}^\infty(\Omega^d))^d$ is nondegenerate, then F has the smooth density $p^F(\cdot)$. Malliavin calculus is further refined by Watanabe's theory. Let $\mathcal{S}(\mathbb{R}^d)$

be the Schwartz space or the space of rapidly decreasing functions and $\mathcal{S}'(\mathbb{R}^d)$ be the dual of $\mathcal{S}(\mathbb{R}^d)$, i.e. $\mathcal{S}'(\mathbb{R}^d)$ is the space of Schwartz tempered distributions. For a tempered distribution $\mathcal{T} \in \mathcal{S}'(\mathbb{R}^d)$ and a nondegenerate Wiener functional in the sense of Malliavin $F \in (\mathbb{D}^\infty(\Omega^d))^d$, $\mathcal{T}(F) = \mathcal{T} \circ F$ is well-defined as an element of the space of Watanabe distributions $\mathbb{D}^{-\infty}(\Omega^d)$, that is the dual space of $\mathbb{D}^\infty(\Omega^d)$ (e.g. see p.379, Corollary of Ikeda and Watanabe (1989) [16], Theorem of Chapter III 6.2 of Malliavin (1997) [25], Theorem 7.3 of Malliavin and Thalmaier (2006) [26]). Also, for $G \in \mathbb{D}^\infty(\Omega^d)$, a (generalized) expectation $E[\mathcal{T}(F)G]$ is understood as a pairing of $\mathcal{T}(F) \in \mathbb{D}^{-\infty}(\Omega^d)$ and $G \in \mathbb{D}^\infty(\Omega^d)$, namely ${}_{\mathbb{D}^\infty}\langle \mathcal{T}(F), G\rangle_{\mathbb{D}^{-\infty}}$, and it holds that

$$ {}_{\mathbb{D}^{-\infty}}\langle \mathcal{T}(F), G\rangle_{\mathbb{D}^\infty} = {}_{\mathcal{S}'}\langle \mathcal{T}, E[G|F=\cdot]p^F(\cdot)\rangle_{\mathcal{S}} \tag{2.6} $$

where ${}_{\mathcal{S}'}\langle \cdot,\cdot\rangle_{\mathcal{S}}$ is the bilinear form on $\mathcal{S}'(\mathbb{R}^d)$ and $\mathcal{S}(\mathbb{R}^d)$, $E[G|F=\xi]$ is the conditional expectation of G conditioned on the set $\{\omega; F(\omega)=\xi\}$ (e.g. see Chapter III 6.2.2 of Malliavin (1997) [25], (7.5) of Theorem 7.3 of Malliavin and Thalmaier (2006) [26]). In particular, we have ${}_{\mathbb{D}^{-\infty}}\langle \delta_y(F), 1\rangle_{\mathbb{D}^\infty} = {}_{\mathcal{S}'}\langle \delta_y, p^F(\cdot)\rangle_{\mathcal{S}} = p^F(y)$ for $y \in \mathbb{R}^d$, and thus p^F is not only smooth but also in $\mathcal{S}(\mathbb{R}^d)$, i.e. a rapidly decreasing function (see Theorem 9.2 of Ikeda and Watanabe (1989) [16]), Proposition 2.1.5 of Nualart (2006) [29]). For a nondegenerate $F \in (\mathbb{D}^\infty(\Omega^d))^d$, $G \in \mathbb{D}^\infty(\Omega^d)$ and a multi-index $\gamma = (\gamma_1,\dots,\gamma_k)$, there exists $H_\gamma(F,G) \in \mathbb{D}^\infty(\Omega^d)$ such that

$$ {}_{\mathbb{D}^{-\infty}}\langle \partial^\gamma \mathcal{T}(F), G\rangle_{\mathbb{D}^\infty} = {}_{\mathbb{D}^{-\infty}}\langle \mathcal{T}(F), H_\gamma(F,G)\rangle_{\mathbb{D}^\infty} \tag{2.7} $$

for all $\mathcal{T} \in \mathcal{S}'(\mathbb{R}^d)$ (e.g. see Chapter 4.4 and Theorem 7.3 of Malliavin and Thalmaier (2006) [26]), where $H_\gamma(F,G)$ is given by $H_\gamma(F,G) = H_{(\gamma_k)}(F, H_{(\gamma_1,\dots,\gamma_{k-1})}(F,G))$ with

$$ H_{(i)}(F,G) = \delta(\textstyle\sum_{j=1}^d (\sigma^F)^{-1}_{ij} DF^j G). \tag{2.8} $$

3 Main result

Let $a \in \mathbb{R}$, $b \in (a,\infty)$ and $T > 0$. For $d \in \mathbb{N}$, consider the solution to the following stochastic differential equation (SDE) driven by a d-dimensional Brownian motion $B^d = (B^{d,1},\dots,B^{d,d})$ on the d-dimensional Wiener space (Ω^d, H^d, μ^d):

$$ dX_t^{d,\lambda,x} = \mu_d^\lambda(X_t^{d,\lambda,x})dt + \sigma_d^\lambda(X_t^{d,\lambda,x})dB_t^d, \quad X_0^{d,\lambda,x} = x \in \mathbb{R}^d, \tag{3.1} $$

where $\mu_d^\lambda : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma_d^\lambda : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ are Lipschitz continuous functions depending on a parameter $\lambda \in (0,1]$. The solution $X_t^{d,\lambda,x} = (X_t^{d,\lambda,x,1},\dots,X_t^{d,\lambda,x,d})$ is equivalently written in the integral form as:

$$ X_t^{d,\lambda,x,j} = x_j + \int_0^t \mu_d^{\lambda,j}(X_s^{d,\lambda,x})ds + \sum_{i=1}^d \int_0^t \sigma_{d,i}^{\lambda,j}(X_s^{d,\lambda,x})dB_s^{d,i}, \quad X_0^{d,\lambda,x,j} = x_j \in \mathbb{R}, \tag{3.2} $$

for $j = 1,\dots,d$. Furthermore, for a given appropriate continuous function $f_d : \mathbb{R}^d \to \mathbb{R}$ and for $\lambda \in (0,1]$, we consider $u_\lambda^d \in C([0,T]\times\mathbb{R}^d, \mathbb{R})$ given by

$$ u_\lambda^d(t,x) = E[f_d(X_t^{d,\lambda,x})] \tag{3.3} $$

for $t \in [0,T]$ and $x \in \mathbb{R}^d$, which is a solution of Kolmogorov PDE:

$$ \partial_t u_\lambda^d(t,x) = \mathcal{L}^{d,\lambda} u_\lambda^d(t,x), \tag{3.4} $$

for all $(t,x) \in (0,T)\times\mathbb{R}^d$ and $u_\lambda^d(0,\cdot) = f_d(\cdot)$, where $\mathcal{L}^{d,\lambda}$ is the following second order differential operator:

$$ \mathcal{L}^{d,\lambda} = \sum_{j=1}^d \mu_d^{\lambda,j}(\cdot)\frac{\partial}{\partial x_j} + \frac{1}{2}\sum_{i,j_1,j_2=1}^d \sigma_{d,i}^{\lambda,j_1}(\cdot)\sigma_{d,i}^{\lambda,j_2}(\cdot)\frac{\partial^2}{\partial x_{j_1}\partial x_{j_2}}. \tag{3.5} $$

Our purpose is to show a new spatial approximation scheme of $u_\lambda^d(t,\cdot)$ for $t > 0$ by using asymptotic expansion and deep neural network approximation. The main theorem (Theorem 1) is stated at the end of this section.

3.1 Asymptotic expansion

We first put the following assumptions on $\{\mu_d^\lambda\}_{\lambda\in(0,1]}$, $\{\sigma_d^\lambda\}_{\lambda\in(0,1]}$ and f_d.

Assumption 1 (Assumptions for the family of SDEs and asymptotic expansion)**.** *Let $C>0$. For $d\in\mathbb{N}$, let $\{\mu_d^\lambda\}_{\lambda\in(0,1]}\subset C_{Lip}(\mathbb{R}^d,\mathbb{R}^d)$ and $\{\sigma_d^\lambda\}_{\lambda\in(0,1]}\subset C_{Lip}(\mathbb{R}^d,\mathbb{R}^{d\times d})$ be families of functions, and $f_d\in C_{Lip}(\mathbb{R}^d,\mathbb{R})$ be a function satisfying*

1. *there are $V_{d,0}\in C_b^\infty(\mathbb{R}^d,\mathbb{R}^d)$ and $V_d=(V_{d,1},\dots,V_{d,d})\in C_b^\infty(\mathbb{R}^d,\mathbb{R}^{d\times d})$ such that (i) $\mu_d^\lambda=\lambda V_{d,0}$ and $\sigma_d^\lambda=\lambda V_d$ for all $\lambda\in(0,1]$, (ii) $C_{Lip}[V_{d,0}]\vee C_{Lip}[V_d]=C$ and $\|V_{d,0}(0)\|\vee\|V_d(0)\|\leq C$, (iii) $\|\partial^\alpha V_{d,i}\|_\infty\leq C$ for any multi-index α and $i=0,1,\dots,d$;*
2. *$\sum_{i=1}^d\sigma_{d,i}^\lambda(x)\otimes\sigma_{d,i}^\lambda(x)\geq\lambda^2 I_d$ for all $x\in\mathbb{R}^d$ and $\lambda\in(0,1]$;*
3. *$C_{Lip}[f_d]=C$ and $\|f_d(0)\|\leq C$.*

Remark 1. *Assumption 1 justify an asymptotic expansion under the uniformly elliptic condition for the solutions of the perturbed systems of PDEs. Assumption 1.3 is also useful for constructing deep neural network approximations for the family of PDE solutions.*

From Assumption 1.2, we may write each SDE (3.1) for $d\in\mathbb{N}$ as

$$dX_t^{d,\lambda,x}=\lambda\sum_{i=0}^d V_{d,i}(X_t^{d,\lambda,x})dB_t^{d,i}, \tag{3.6}$$

with $X_0^{d,\lambda,x}=x\in\mathbb{R}^d$, where the notation $dB_t^{d,0}=dt$ is used. We define

$$\mathbb{B}_t^{d,\alpha}=\int_{0<t_1<\cdots<t_k<t}dB_{t_1}^{d,\alpha_1}\cdots dB_{t_k}^{d,\alpha_k},\quad t\geq 0,\ \alpha\in\{0,1,\dots,d\}^k,\ k\in\mathbb{N}, \tag{3.7}$$

and $L_{d,0}=\sum_{j=1}^d V_{d,0}^j(\cdot)\frac{\partial}{\partial x_j}+\frac{1}{2}\sum_{i,j_1,j_2=1}^d V_{d,i}^{j_1}(\cdot)V_{d,i}^{j_2}(\cdot)\frac{\partial^2}{\partial x_{j_1}\partial x_{j_2}}$, $L_{d,i}=\sum_{j=1}^d V_{d,i}^j(\cdot)\frac{\partial}{\partial x_j}$, $i=1,\dots,d$. We define

$$\bar{X}_t^{d,\lambda,x}=x+\lambda\sum_{i=0}^d V_{d,i}(x)B_t^{d,i}. \tag{3.8}$$

Proposition 1 (Asymptotic expansion and the error bound)**.** *For $m\in\mathbb{N}\cup\{0\}$, there exists $c>0$ such that for all $d\in\mathbb{N}$, $t>0$, $\lambda\in(0,1]$,*

$$\begin{aligned}\sup_{x\in[a,b]^d}&\Big|E[f_d(X_t^{d,\lambda,x})]-\Big\{E\Big[f_d(\bar{X}_t^{d,\lambda,x})\Big]\\&+\sum_{j=1}^m\lambda^j E\Big[f_d(\bar{X}_t^{d,\lambda,x})\sum_{\beta^{(k)},\gamma^{(k)}}^{(j)}H_{\gamma^{(k)}}\Big(\sum_{i=0}^d V_{d,i}(x)B_t^{d,i},\prod_{\ell=1}^k\sum_{|\alpha|=\beta_\ell}\hat{V}_{d,\alpha}^{\gamma_\ell}(x)\mathbb{B}_t^{d,\alpha}\Big)\Big]\Big\}\Big|\\ \leq\ & cd^c\lambda^{m+1}t^{(m+1)/2},\end{aligned} \tag{3.9}$$

where $\hat{V}_{d,\alpha}^e(x)=L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}}V_{d,\alpha_r}^e(x)$, $e\in\{1,\dots,d\}$, $\alpha\in\{1,\dots,d\}^p$, and

$$\sum_{\beta^{(k)},\gamma^{(k)}}^{(j)}=\sum_{k=1}^j\ \sum_{\beta^{(k)}=(\beta_1,\dots,\beta_k)\ s.t.\ \beta_1+\cdots+\beta_k=j+k,\beta_i\geq 2}\ \sum_{\gamma^{(k)}=(\gamma_1,\dots,\gamma_k)\in\{1,\dots,d\}^k}\frac{1}{k!},\quad j\geq 1. \tag{3.10}$$

Proof of Proposition 1. See Section 4. □

The weights in the expansion terms in Proposition 1 can be represented by some polynomials of Brownian motion. We show it through distribution theory on Wiener space. Let $d\in\mathbb{N}$, for $t\in(0,T]$ and $\alpha=(\alpha_1,\dots,\alpha_k)\in\{0,1,\dots,d\}^k$, $k\in\mathbb{N}\cap[2,\infty)$, let

$$\mathbf{B}_t^{d,\alpha}=\delta^{\alpha_k}(\mathbf{B}_t^{d,(\alpha_1,\dots,\alpha_{k-1})})=B_t^{d,\alpha_k}\mathbf{B}_t^{d,(\alpha_1,\dots,\alpha_{k-1})}-\int_0^t D_{\alpha_k,s}\mathbf{B}_t^{d,(\alpha_1,\dots,\alpha_{k-1})}ds, \tag{3.11}$$

with $\mathbf{B}_t^{d,(\alpha_1)} = B_t^{d,\alpha_1}$, which can be obtained by (2.5). For example, we have $\mathbf{B}_t^{d,(\alpha_1,\alpha_2)} = B_t^{d,\alpha_1}B_t^{d,\alpha_2} - t\mathbf{1}_{\alpha_1=\alpha_2\neq 0}$ for $\alpha = (\alpha_1,\alpha_2) \in \{0,1,\ldots,d\}^2$. Let $\sigma_\ell \in \mathbb{R}^d$, $\ell = 0,1,\ldots,d$ and Σ be a matrix given by $\Sigma_{i,j} = \sum_{\ell=1}^d \sigma_\ell^i \sigma_\ell^j$, $1 \le i,j \le d$ and satisfying $\det\Sigma > 0$. Let $\mathcal{T} \in \mathcal{S}'(\mathbb{R}^d)$. We show an efficient computation of ${}_{\mathbb{D}^{-\infty}}\langle \mathcal{T}(\sum_{i=0}^d \sigma_i B_t^{d,i}), H_\gamma(\sum_{i=0}^d \sigma_i B_t^{d,i}, \mathbb{B}_t^{d,\alpha})\rangle_{\mathbb{D}^\infty}$ in order to give a polynomial representation of the Malliavin weights in the expansion terms of the asymptotic expansion in Proposition 1. Note that we have

$$
\begin{aligned}
&{}_{\mathbb{D}^{-\infty}}\Big\langle \mathcal{T}\Big(\sum_{i=0}^d \sigma_i B_t^{d,i}\Big), H_\gamma\Big(\sum_{i=0}^d \sigma_i B_t^{d,i}, \mathbb{B}_t^{d,\alpha}\Big)\Big\rangle_{\mathbb{D}^\infty} = {}_{\mathbb{D}^{-\infty}}\Big\langle \partial^\gamma \mathcal{T}\Big(\sum_{i=0}^d \sigma_i B_t^{d,i}\Big), \mathbb{B}_t^{d,\alpha}\Big\rangle_{\mathbb{D}^\infty} \\
&= {}_{\mathcal{S}'}\langle \partial^\gamma \mathcal{T}(\sigma_0 B_t^{d,0} + \sigma\ \cdot), E[\mathbb{B}_t^{d,\alpha} | B_t^d = \ \cdot\ ] p^{B_t^d}(\cdot)\rangle_{\mathcal{S}}, \qquad (3.12)
\end{aligned}
$$

by (2.7) and (2.6), where σ is the matrix $\sigma = (\sigma_1,\ldots,\sigma_d)$, and for $y \in \mathbb{R}^d$, it holds that

$$
E[\mathbb{B}_t^{d,\alpha} | B_t^d = y] p^{B_t^d}(y) = {}_{\mathcal{S}'}\langle \delta_y, E[\mathbb{B}_t^{d,\alpha} | B_t^d = \ \cdot\ ] p^{B_t^d}(\cdot)\rangle_{\mathcal{S}} = {}_{\mathbb{D}^{-\infty}}\langle \delta_y(B_t^d), \mathbb{B}_t^{d,\alpha}\rangle_{\mathbb{D}^\infty},
$$

by (2.6). Also, one has

$$
\begin{aligned}
{}_{\mathbb{D}^{-\infty}}\langle \delta_y(B_t^d)\mathbb{B}_t^{d,\alpha}\rangle_{\mathbb{D}^\infty} &= {}_{\mathbb{D}^{-\infty}}\langle \partial^{\alpha^\star}\delta_y(B_t^d), 1\rangle_{\mathbb{D}^\infty} \frac{1}{k!} t^k \\
&= {}_{\mathbb{D}^{-\infty}}\langle \delta_y(B_t^d), H_{\alpha^\star}(B_t^d, 1)\rangle_{\mathbb{D}^\infty} \frac{1}{k!} t^k = {}_{\mathbb{D}^{-\infty}}\langle \delta_y(B_t^d), \frac{1}{k!}\mathbf{B}_t^{d,\alpha}\rangle_{\mathbb{D}^\infty}, \qquad (3.13)
\end{aligned}
$$

by (2.5), (2.7) and (2.8), where $\alpha^\star$ is a multi-index such that $\alpha^\star = (\alpha_1^\star,\ldots,\alpha_{\ell(\alpha)}^\star) = (\alpha_{j_1},\ldots,\alpha_{j_{\ell(\alpha)}})$ satisfying $\ell(\alpha) = \#\{i; \alpha_i \neq 0\}$ and $\alpha_{j_i} \neq 0$, $i = 1,\ldots,\ell(\alpha)$. Then, we have

$$
\begin{aligned}
&{}_{\mathbb{D}^{-\infty}}\langle \mathcal{T}(\sum_{i=0}^d \sigma_i B_t^{d,i}), H_\gamma(\sum_{i=0}^d \sigma_i B_t^{d,i}, \mathbb{B}_t^{d,\alpha})\rangle_{\mathbb{D}^\infty} = {}_{\mathcal{S}'}\langle \partial^\gamma \mathcal{T}(\sigma_0 B_t^{d,0} + \sigma\ \cdot), \frac{1}{k!} E[\mathbf{B}_t^{d,\alpha} | B_t^d = \ \cdot\ ] p^{B_t^d}(\cdot)\rangle_{\mathcal{S}} \\
&= {}_{\mathbb{D}^{-\infty}}\langle \partial^\gamma \mathcal{T}(\sum_{i=0}^d \sigma_i B_t^{d,i}), \frac{1}{k!}\mathbf{B}_t^{d,\alpha}\rangle_{\mathbb{D}^\infty} = {}_{\mathbb{D}^{-\infty}}\langle \mathcal{T}(\sum_{i=0}^d \sigma_i B_t^{d,i}), H_\gamma(\sum_{i=0}^d \sigma_i B_t^{d,i}, \frac{1}{k!}\mathbf{B}_t^{d,\alpha})\rangle_{\mathbb{D}^\infty} \\
&= {}_{\mathbb{D}^{-\infty}}\Big\langle \mathcal{T}(\sum_{i=0}^d \sigma_i B_t^{d,i}), \sum_{j_1,\ldots,j_{|\gamma|},\beta_1,\ldots,\beta_{|\gamma|}=1}^d \frac{1}{t^{|\gamma|}} \prod_{q=1}^{|\gamma|} \Sigma^{-1}_{\gamma_q,j_q} \sigma_{\beta_q}^{j_q} \frac{1}{k!} \mathbf{B}_t^{d,(\alpha_1,\ldots,\alpha_k,\beta_1,\ldots,\beta_{|\gamma|})}\Big\rangle_{\mathbb{D}^\infty}, \qquad (3.14)
\end{aligned}
$$

where, we iteratively used (2.5), (2.6), (2.7) and (2.8). An explicit polynomial representation of the asymptotic expansion is derived through (3.14). For instance, the first order expansion ($m = 1$) as follows:

(*First order asymptotic expansion with Malliavin weight*)

$$
\begin{aligned}
&E\Big[f_d(\bar{X}_t^{d,\lambda,x})\Big\{1 + \lambda \sum_{\ell=1}^d H_{(\ell)}\Big(\sum_{i=0}^d V_{d,i}(x) B_t^{d,i}, \sum_{\alpha_1,\alpha_2=0}^d L_{d,\alpha_1} V_{d,\alpha_2}^\ell(x) \mathbb{B}_t^{d,(\alpha_1,\alpha_2)}\Big)\Big\}\Big] \\
=&E\Big[f_d(\bar{X}_t^{d,\lambda,x})\Big] + \lambda \sum_{\ell=1}^d \int_{\mathbb{R}^d} f_d(x + \lambda y) \sum_{\alpha_1,\alpha_2=0}^d L_{d,\alpha_1} V_{d,\alpha_2}^\ell(x) \\
&\qquad {}_{\mathbb{D}^{-\infty}}\Big\langle \delta_y(\sum_{i=0}^d V_{d,i}(x) B_t^{d,i}), H_{(\ell)}\Big(\sum_{i=0}^d V_{d,i}(x) B_t^{d,i}, \mathbb{B}_t^{d,(\alpha_1,\alpha_2)}\Big)\Big\rangle_{\mathbb{D}^\infty} dy \\
=&E\Big[f_d(\bar{X}_t^{d,\lambda,x})\Big] + \lambda \sum_{\ell=1}^d \int_{\mathbb{R}^d} f_d(x + \lambda y) \sum_{\alpha_1,\alpha_2=0}^d L_{d,\alpha_1} V_{d,\alpha_2}^\ell(x) \\
&\qquad {}_{\mathbb{D}^{-\infty}}\Big\langle \delta_y(\sum_{i=0}^d V_{d,i}(x) B_t^{d,i}), \sum_{\alpha_3=1}^d \sum_{j=1}^d \frac{1}{2t} [A_d^{-1}]_{\ell j}(x) V_{d,\alpha_3}^j(x) \mathbf{B}_t^{d,(\alpha_1,\alpha_2,\alpha_3)}\Big\rangle_{\mathbb{D}^\infty} dy \\
=&E\Big[f_d(\bar{X}_t^{d,\lambda,x})\{1 + \lambda \sum_{\ell,j=1}^d \sum_{\alpha_1,\alpha_2=0}^d \sum_{\alpha_3=1}^d L_{d,\alpha_1} V_{d,\alpha_2}^\ell(x) \frac{1}{2t} [A_d^{-1}]_{\ell j}(x) V_{d,\alpha_3}^j(x) \mathbf{B}_t^{d,(\alpha_1,\alpha_2,\alpha_3)}\Big].
\end{aligned}
$$

Thus, the first order expansion is expressed with a Malliavin weight given by third order polynomials of Brownian motion. In general, we have the following representation.

Proposition 2. *For $m \in \mathbb{N}$, $d \in \mathbb{N}$, $\lambda \in (0,1]$, $t \in (0,T]$ and $x \in \mathbb{R}^d$, there exists a Malliavin weight $\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)$ such that*

$$\begin{aligned} &E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)] \\ =&E\Big[f_d(\bar{X}^{d,\lambda,x}_t)\Big\{1+\sum_{j=1}^m \lambda^j \sum_{\beta^{(k)},\gamma^{(k)}}^{(j)} H_{\gamma^{(k)}}\Big(\sum_{i=0}^d V_{d,i}(x)B^{d,i}_t, \prod_{\ell=1}^k \sum_{|\alpha|=\beta_\ell} \hat{V}^{\gamma_\ell}_{d,\alpha}(x)\mathbb{B}^{d,\alpha}_t\Big)\Big\}\Big], \end{aligned} \tag{3.15}$$

and

$$\mathcal{M}^m_{d,\lambda}(t,x,B^d_t) = 1 + \textstyle\sum_{e\le n(m)} \lambda^{p(e)} g_e(t) h_e(x) \mathrm{Poly}_e(B^d_t) \tag{3.16}$$

for some integers $n(m) \in \mathbb{N}$ and $p(e) \in \mathbb{N}$, $e = 1,\dots,n(m)$, polynomials $\mathrm{Poly}_e : \mathbb{R}^d \to \mathbb{R}$, $e = 1,\dots,n(m)$, continuous functions $g_e : (0,T] \to \mathbb{R}$, $e = 1,\dots,n(m)$, and continuous functions $h_e : \mathbb{R}^d \to \mathbb{R}$, $e = 1,\dots,n(m)$ constructed by some products of A_d^{-1}, $\{V_{d,i}\}_{0\le i\le d}$ and $\{\partial^\alpha V_{d,i}\}_{0\le i\le d,\alpha\in\{1,\dots,d\}^\ell,\ell\le 2m}$ given in Assumption 1 of the form:

$$x \mapsto h_e(x) = c_e \textstyle\prod_{\ell=1}^{q_e} L_{d,\alpha^e_{\ell,1}} \cdots L_{d,\alpha^e_{\ell,p^e_\ell -1}} V^{\gamma^e_\ell}_{d,\alpha^e_{\ell,p^e_\ell}}(x) \sum_{\xi,\iota=1}^d [A_d^{-1}]_{\gamma^e_\ell,\xi}(x) V^\xi_{d,\iota}(x) \tag{3.17}$$

with some constants $c_e \in (0,\infty)$, $q_e \in \mathbb{N}$ and some multi-indices $(\gamma^e_1,\dots,\gamma^e_\ell) \in \{1,\dots,d\}^\ell$ and $(\alpha^e_{\ell,1},\dots,\alpha^e_{\ell,p^e_\ell}) \in \{0,1,\dots,d\}^{p^e_\ell}$ with $p^e_\ell \in \mathbb{N}$, $\ell = 1,\dots,e$, which satisfies that for $p \ge 1$,

$$\sup_{(t,x)\in(0,T]\times[a,b]^d,\lambda\in(0,1]} \|\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)\|_p \le cd^c \tag{3.18}$$

for some constant $c > 0$ independent of d.

Proof of Proposition 2. See Section 4. □

Remark 2 (Remark on computation of Malliavin weights)**.** *Malliavin weight is initially used in Fournie et. al [7] in sensitivity analysis in financial mathematics, especially in Monte-Carlo computation of "Greeks". Then a discretization scheme for probabilistic automatic differentiation using Malliavin weights is analyzed in Gobet and Munos [10]. The computation of asymptotic expansion with Malliavin weights is developed in Takahashi and Yamada [35][37], and is further extended to weak approximation of SDEs in Takahashi and Yamada [38]. Note that a PDE expansion is shown in Takahashi and Yamada [36] to partially connect it with the stochastic calculus approach. The computation method of the expansion with Malliavin weights is improved in Yamada [41], Naito and Yamada [27, 28], Iguchi and Yamada [17, 18], and Takahashi et al. [34] where technique of stochastic calculus is refined. The main advantages of the stochastic calculus approach are that (i) it provides efficient computation scheme using Watanabe distributions on Wiener space as in (3.13) and (3.14), (ii) it enables us to give precise bounds for approximations of expectations or the corresponding solutions of PDEs. Actually, the computational effort of the expansions is much reduced in the sense that Itô's iterated integrals are transformed into simple polynomials of Brownian motion, and also the desired deep neural network approximation will be obtained in the next subsection through the approach.*

3.2 Deep neural network approximation

In order to construct a deep neural network approximation for the function with respect to the space variable of the asymptotic expansion, i.e. $x \mapsto E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)]$, we consider the further assumptions.

Assumption 2 (Assumptions for deep neural network approximation)**.** *Suppose that Assumption 1 holds. There exist a constant $\kappa > 0$ and sets of networks $\{\psi^{V_{d,i}}_{\varepsilon,d}\}_{\varepsilon\in(0,1),d\in\mathbb{N},i\in\{0,1,\ldots,d\}} \subset \mathcal{N}$, $\{\psi^{\partial^\alpha V_{d,i}}_{\varepsilon,d}\}_{\varepsilon\in(0,1),d\in\mathbb{N},i\in\{0,1,\ldots,d\},\alpha\in\{1,\ldots,d\}^{\mathbb{N}}} \subset \mathcal{N}$, $\{\psi^{A_d^{-1}}_{\varepsilon}\}_{\varepsilon\in(0,1),d\in\mathbb{N}} \subset \mathcal{N}$ and $\{\psi^{f_d}_{\varepsilon}\}_{\varepsilon\in(0,1),d\in\mathbb{N}} \subset \mathcal{N}$ such that*

1. *for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $\mathcal{C}(\psi^{V_{d,i}}_{\varepsilon,d}) \leq \kappa d^\kappa \varepsilon^{-\kappa}$, $i = 0,1,\ldots,d$, $\mathcal{C}(\psi^{\partial^\alpha V_{d,i}}_{\varepsilon,d}) \leq \kappa d^\kappa \varepsilon^{-\kappa}$, $i = 0,1,\ldots,d$, $\alpha \in \{1,\ldots,d\}^\ell$, $\ell \in \mathbb{N}$, $\mathcal{C}(\psi^{A_d^{-1}}_{\varepsilon}) \leq \kappa d^\kappa \varepsilon^{-\kappa}$, and $\mathcal{C}(\psi^{f_d}_{\varepsilon}) \leq \kappa d^\kappa \varepsilon^{-\kappa}$;*
2. *for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\|V_{d,i}(x) - V^\varepsilon_{d,i}(x)\| \leq \varepsilon\kappa d^\kappa$, $i = 0,1,\ldots,d$, and $\|\partial^\alpha V_{d,i}(x) - V^\varepsilon_{d,i,\alpha}(x)\| \leq \varepsilon\kappa d^\kappa$, $i = 0,1,\ldots,d$, $\alpha \in \{1,\ldots,d\}^\ell$, $\ell \in \mathbb{N}$, where $V^\varepsilon_{d,i} = \mathcal{R}(\psi^{V_{d,i}}_\varepsilon) \in C(\mathbb{R}^d,\mathbb{R}^d)$ and $V^\varepsilon_{d,i,\alpha} = \mathcal{R}(\psi^{\partial^\alpha V_{d,i}}_\varepsilon) \in C(\mathbb{R}^d,\mathbb{R}^d)$;*
3. *for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\|A_d^{-1}(x) - A_{d,\varepsilon}^{-1}(x)\| \leq \varepsilon\kappa d^\kappa$, where $A_d^{-1}(\cdot)$ is the inverse matrix of $A_d(\cdot) := \sum_{i=1}^d V_{d,i}(\cdot) \otimes V_{d,i}(\cdot)$ and $A^{-1}_{d,\varepsilon} = \mathcal{R}(\psi^{A_d^{-1}}_\varepsilon) \in C(\mathbb{R}^d,\mathbb{R}^{d\times d})$, and for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $\sup_{x\in[a,b]^d}\|A^{-1}_{d,\varepsilon}(x)\| \leq \kappa d^\kappa$;*
4. *for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $x \in \mathbb{R}^d$, $|f_d(x) - f^\varepsilon_d(x)| \leq \varepsilon\kappa d^\kappa$, where $f^\varepsilon_d = \mathcal{R}(\psi^{f_d}_\varepsilon) \in C(\mathbb{R}^d,\mathbb{R})$.*

Remark 3. *Assumption 2 provides the deep neural network approximation of the asymptotic expansion with an appropriate complexity. Note that Assumption 1.1, 1.3, 2.2 and 2.4 give that there exists $\eta > 0$ such that $|f^\varepsilon_d(x)| \leq \eta d^\eta(1+\|x\|)$ for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, and $\sup_{x\in[a,b]^d}\|V^\varepsilon_{d,i}(x)\| \leq \eta d^\eta$ for all $i = 0,1,\ldots,d$, $\sup_{x\in[a,b]^d}\|V^\varepsilon_{d,i,\alpha}(x)\| \leq \eta d^\eta$ for all $i = 0,1,\ldots,d$, $\alpha \in \{1,\ldots,d\}^\ell$ with $\ell \in \mathbb{N}$. In the following, Assumption 2.2, 2.3 and 2.4 plays an important role for the analysis of "product of neural networks" in the construction of the approximation with asymptotic expansion.*

Remark 4. *In particular, Assumption 2.3 is satisfied for the cases $A_d(x) = I_d$ and $A_d(x) = s(d)I_d$ with a function $s : \mathbb{N} \to \mathbb{R}$. For instance, the case $A_d(x) = I_d$ corresponds to the d-dimensional heat equation when $V_{d,0} \equiv 0$. Also, the SDEs with the diffusion matrix $V_d = (1/\sqrt{d})I_d$ discussed in Section 5.1 and Section 5.2 of [9] and Section 5.2 of [13] are examples of (3.1) (or (3.6)). For those cases, the neural network approximations in Assumption 2 are not necessary, since $V_{d,i}$, $i = 1,\ldots,d$ and hence A_d do not depend on the state variable x, whence $V_{d,i,\varepsilon}$ and $A^{-1}_{d,\varepsilon}$ are $V_{d,i}$ and A_d^{-1} themselves. Furthermore, in such cases (e.g. the high-dimensional heat equations) the asymptotic expansion will be simply obtained (usually as the Gaussian approximation), which are exactly reduced to the methods in Beck et al. [2] and Gonon et al. [11].*

The main result of the paper is summarized as follows.

Theorem 1 (Deep learning-based asymptotic expansion overcomes the curse of dimensionality)**.** *Suppose that Assumption 1 and Assumption 2 hold. Let $m \in \mathbb{N}$. For $d \in \mathbb{N}$, consider the SDE (3.1) on the d-dimensional Wiener space and let $u^d_\lambda \in C([0,T]\times\mathbb{R}^d,\mathbb{R})$ given by (3.3) be a solution to the Kolmogorov PDE (3.4). Then we have*

$$\sup_{x\in[a,b]^d}|u^d_\lambda(t,x) - E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)]| = O(\lambda^{m+1}t^{(m+1)/2}). \tag{3.19}$$

Furthermore, for $t \in (0,T]$ and $\lambda \in (0,1]$, there exist $\{\phi^{\varepsilon,d}\}_{\varepsilon\in(0,1),d\in\mathbb{N}} \subset \mathcal{N}$ and $c > 0$ which depend only on a, b, C, m, κ, t and λ, such that for all $\varepsilon \in (0,1)$ and $d \in \mathbb{N}$, we have $\mathcal{R}(\phi^{\varepsilon,d}) \in C(\mathbb{R}^d,\mathbb{R})$, $\mathcal{C}(\phi^{\varepsilon,d}) \leq c\varepsilon^{-c}d^c$ and

$$\sup_{x\in[a,b]^d}|E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)] - \mathcal{R}(\phi^{\varepsilon,d})(x)| \leq \varepsilon. \tag{3.20}$$

Proof. See Section 4. □

We provide the weight $\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)$ with $m = 0,1$ in Theorem 1 for our scheme (the expression for general m will be given in Section 4 below). That is, for $d \in \mathbb{N}$, $\lambda \in (0,1]$, $t > 0$ and

$x \in \mathbb{R}^d$,

$$\mathcal{M}^0_{d,\lambda}(t,x,B^d_t) = 1, \tag{3.21}$$

$$\mathcal{M}^1_{d,\lambda}(t,x,B^d_t) = 1 + \lambda \sum_{\alpha_1,\alpha_2=0}^{d} \sum_{\alpha_3=1}^{d} \sum_{\ell,j=1}^{d} \frac{1}{2t} L_{d,\alpha_1} V^{\ell}_{d,\alpha_2}(x)[A^{-1}_d]_{\ell j}(x) V^j_{d,\alpha_3}(x)$$
$$\{B^{d,\alpha_1}_t B^{d,\alpha_2}_t B^{d,\alpha_3}_t - tB^{d,\alpha_1}_t \mathbf{1}_{\alpha_2=\alpha_3\neq 0} - tB^{d,\alpha_2}_t \mathbf{1}_{\alpha_1=\alpha_3\neq 0} - tB^{d,\alpha_3}_t \mathbf{1}_{\alpha_1=\alpha_2\neq 0}\}, \tag{3.22}$$

where

$$L_{d,0} = \sum_{j=1}^{d} V^j_{d,0}(\cdot)\frac{\partial}{\partial x_j} + \frac{1}{2}\sum_{i,j_1,j_2=1}^{d} V^{j_1}_{d,i}(\cdot) V^{j_2}_{d,i}(\cdot)\frac{\partial^2}{\partial x_{j_1}\partial x_{j_2}}, \tag{3.23}$$

$$L_{d,i} = \sum_{j=1}^{d} V^j_{d,i}(\cdot)\frac{\partial}{\partial x_j}, \quad i = 1,\dots,d. \tag{3.24}$$

Hence, the weight for $m = 0$, i.e. $\mathcal{M}^0_{d,\lambda}(t,x,B^d_t) = 1$ provides a simple (but coarse) Gaussian approximation, and the Malliavin weight for $m = 1$ will be worked as the correction term for the Gaussian approximation. The derivation is provided in the next section.

4 Proofs of Proposition 1, 2 and Theorem 1

We give the proofs of Proposition 1, 2 and Theorem 1. Before providing full proofs, we show their brief outlines below.

- Proposition 1 (Asymptotic expansion)
  - take a family of uniformly non-degenerate functionals $F^{d,\lambda,x}_t = (X^{d,\lambda,x}_t - x)/\lambda$, $\lambda \in (0,1]$, as the family $X^{d,\lambda,x}_t$, $\lambda \in (0,1]$ itself degenerates when $\lambda \downarrow 0$, and consider the expansion $F^{d,\lambda,x}_t = F^{d,0,x}_t + \cdots$ in $\mathbb{D}^\infty$.
  - expand $\delta_y(F^{d,\lambda,x}_t) \sim \delta_y(F^{d,0,x}_t) + \cdots$ in $\mathbb{D}^{-\infty}$ and take expectation to obtain the expansion of the density $p^{F^{d,\lambda,x}_t}(y) = E[\delta_y(F^{d,\lambda,x}_t)] \sim E[\delta_y(F^{d,0,x}_t)] + \cdots$ in $\mathbb{R}$.
  - derive precise expression of the right-hand side of $E[f_d(X^{d,\lambda,x}_t)] = c^{d,\lambda,t}_0 + c^{d,\lambda,t}_1 + \cdots + c^{d,\lambda,t}_m + \mathrm{Residual}^{d,\lambda,t}_m$ by using Malliavin's integration by parts.
  - give a precise estimate for $\mathrm{Residual}^{d,\lambda,t}_m(x)$ (w.r.t λ, t and the dimension d) uniformly in x by using the key inequality on Malliavin weight (Lemma 5 in Appendix A) which yields a sharp upper bound of $\mathrm{Residual}^{d,\lambda,t}_m(x)$.
- Proposition 2 (Representation and property of Malliavin weight)
  - use the formula (3.14) to prove that $c^{d,\lambda,t}_0 + c^{d,\lambda,t}_1 + \cdots + c^{d,\lambda,t}_m$ above can be represented by an expectation $E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)]$ with a Malliavin weight $\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)$ constructed by polynomials of Brownian motion.
  - check that the moment of the Malliavin weight $\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)$ grows polynomially in d from the representation.
- Theorem 1 (Deep learning-based asymptotic expansion overcomes the curse of dimensionality)
  - (0) for $d \in \mathbb{N}$, first check the expansion $E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)]$ obtained in Proposition 1 and 2 gives an approximation for $u^\lambda_d(t,x)$ on the cube $[a,b]^d$ with a sharp asymptotic error bound.

– (1) for an error precision ε, construct an approximation $E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] \approx E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda,\delta}^m(t,x,B_t^d)]$ on the cube $[a,b]^d$ by using stochastic calculus, where f_d^δ, $\bar{X}_t^{d,\lambda,x,\delta}$ and $\mathcal{M}_{d,\lambda,\delta}^m(t,x,B_t^d)$ are given by replacing $\{V_{d,i}\}_i$, A_d^{-1}, $\{V_{d,i,\alpha}\}_{i,\alpha}$ with their neural network approximations $\{V_{d,i}^\delta\}_i$, $A_{d,\delta}^{-1}$, $\{V_{d,i,\alpha,\delta}\}_{i,\alpha}$ with $\delta=(\varepsilon^c d^{-c})$ for some $c>0$ independent of ε and d.
– (2) for an error precision ε, construct a realization of the Monte-Carlo approximation $E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda,\delta}^m(t,x,B_t^d)] \approx \frac{1}{M}\sum_{\ell=1}^M f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta,(\ell)}(\omega_{\varepsilon,d}))\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^{d,(\ell)}(\omega_{\varepsilon,d}))$ on the cube $[a,b]^d$ with a choice $M=O(\varepsilon^{-c}d^c)$ for some $c>0$ independent of ε and d, by using stochastic calculus.
– (3) for an error precision ε, construct a realization of the deep neural network approximation $\frac{1}{M}\sum_{\ell=1}^M f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta,(\ell)}(\omega_{\varepsilon,d}))\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^{d,(\ell)}(\omega_{\varepsilon,d})) \approx \mathcal{R}(\phi_{\varepsilon,d})(x)$ on the cube $[a,b]^d$ whose complexity is bounded by $\mathcal{C}(\phi_{\varepsilon,d}) \le c\varepsilon^{-c}d^c$ for some $c>0$ independent of ε and d, where ReLU calculus (Lemma 9, 10, 12 in Appendix B) is essentially used.
– apply (0), (1), (2) and (3) to obtain the main result.

In the proof, we frequently use an elementary result: $\sup_{x\in[a,b]^d}\|x\| \le d^{1/2}\max\{|a|,|b|\}$, which is obtained in the proof of Corollary 4.2 of [11].

4.1 Proof of Proposition 1

For $x\in\mathbb{R}^d$, $t\in(0,T]$ and $\lambda\in(0,1]$, let $F_t^{d,\lambda,x}=(F_t^{d,\lambda,x,1},\dots,F_t^{d,\lambda,x,d})\in(\mathbb{D}^\infty(\Omega^d))^d$ be given by $F_t^{d,\lambda,x,j}=(X_t^{d,\lambda,x,j}-x_j)/\lambda$, $j=1,\dots,d$. We note that $\{F_t^{d,\lambda,x}\}_\lambda$ is a family of uniformly non-degenerate Wiener functionals (see Theorem 3.4 of [40]). Then, for $\mathcal{T}\in\mathcal{S}'(\mathbb{R}^d)$, the composition $\mathcal{T}(F_t^{d,\lambda,x})$ is well-defined as an element of $\mathbb{D}^{-\infty}(\Omega^d)$, and the density of $F_t^{d,\lambda,x}$, namely $p^{F_t^{d,\lambda,x}}\in\mathcal{S}(\mathbb{R}^d)$ has the representation $p^{F_t^{d,\lambda,x}}(y)={}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\lambda,x}),1\rangle_{\mathbb{D}^{-\infty}}$ for $y\in\mathbb{R}^d$. Then, for $x\in\mathbb{R}^d$, $t>0$ and $\lambda\in(0,1]$, it holds that

$$E[f_d(X_t^{d,\lambda,x})]=\int_{\mathbb{R}^d} f_d(x+\lambda y)_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\lambda,x}),1\rangle_{\mathbb{D}^{-\infty}}dy. \tag{4.1}$$

For $x\in\mathbb{R}^d$, $t\in(0,T]$, let $F_t^{d,0,x}=\sum_{i=0}^d V_{d,i}(x)B_t^{d,i}$. Thus, for $S\in\mathcal{S}'(\mathbb{R}^d)$, the composition $S(F_t^{d,\lambda,x})$ is well-defined as an element of $\mathbb{D}^{-\infty}(\Omega^d)$ and has an expansion:

$$\begin{aligned}{}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\lambda,x}),1\rangle_{\mathbb{D}^\infty} ={}&{}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,0,x}),1\rangle_{\mathbb{D}^\infty}\\ &+\sum_{j=1}^m\frac{\lambda^j}{j!}\frac{\partial^j}{\partial\lambda^j}{}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\lambda,x}),1\rangle_{\mathbb{D}^\infty}|_{\lambda=0}+\lambda^{m+1}\mathcal{E}_{m,t}^{d,\lambda,x,y},\end{aligned} \tag{4.2}$$

for $x\in\mathbb{R}^d$, $t>0$ and $\lambda\in(0,1]$, where

$$\mathcal{E}_{m,t}^{d,\lambda,x,y}=\int_0^1\frac{(1-u)^m}{m!}\frac{\partial^{m+1}}{\partial\eta^{m+1}}{}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\eta,x}),1\rangle_{\mathbb{D}^\infty}|_{\eta=\lambda u}du. \tag{4.3}$$

By the integration by parts (2.7) and Theorem 2.6 of [35] yield that

$$\begin{aligned}&\frac{1}{j!}\frac{\partial^j}{\partial\lambda^j}{}_{\mathbb{D}^{-\infty}}\langle\delta_y(F_t^{d,\lambda,x}),1\rangle_{\mathbb{D}^\infty}|_{\lambda=0}\\ &=\sum_{i^{(k)},\gamma^{(k)}}^j{}_{\mathbb{D}^{-\infty}}\Big\langle\delta_y(F_t^{d,0,x}),H_{\gamma^{(k)}}(F_t^{d,0,x},\prod_{\ell=1}^k\frac{1}{i_\ell!}\frac{\partial^{i_\ell}}{\partial\lambda^{i_\ell}}F_t^{d,\lambda,x,\gamma_\ell}|_{\lambda=0})\Big\rangle_{\mathbb{D}^\infty}.\end{aligned} \tag{4.4}$$

where $\sum_{i^{(k)},\gamma^{(k)}}^j=\sum_{k=1}^j\sum_{i^{(k)}=(i_1,\dots,i_k)\ s.t.\ i_1+\cdots+i_k=j,i_e\ge1}\sum_{\gamma^{(k)}=(\gamma_1,\dots,\gamma_k)\in\{1,\cdots,d\}^k}\frac{1}{k!}$ With a calculation

$$\frac{1}{i!}\frac{\partial^i}{\partial\lambda^i}F_t^{d,\lambda,x,j}|_{\lambda=0}=\sum_{|\alpha|=i+1}L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}}V_{d,\alpha_r}^j(x)\mathbb{B}_t^{d,\alpha} \tag{4.5}$$

for $j = 1, \dots, d$ and $i \in \mathbb{N}$, it holds that

$$\begin{aligned}
&{}_{\mathbb{D}^{-\infty}}\langle \delta_y(F_t^{d,\lambda,x}), 1\rangle_{\mathbb{D}^\infty} = {}_{\mathbb{D}^{-\infty}}\langle \delta_y(F_t^{d,0,x}), 1\rangle_{\mathbb{D}^\infty} \\
&+ \sum_{j=1}^m \lambda^j \sum_{i^{(k)},\gamma^{(k)}}^{j} {}_{\mathbb{D}^{-\infty}}\langle \delta_y(F_t^{d,0,x}), H_{\gamma^{(k)}}(F_t^{d,0,x}, \prod_{\ell=1}^k \sum_{|\alpha|=i_\ell} L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}} V_{d,\alpha_r}^{\gamma_\ell}(x)\mathbb{B}_t^{d,\alpha})\rangle_{\mathbb{D}^\infty} \\
&+ \lambda^{m+1}\mathcal{E}_{m,t}^{d,\lambda,x,y},
\end{aligned} \tag{4.6}$$

Again by the integration by parts (2.7), $\frac{\partial^{m+1}}{\partial\eta^{m+1}} {}_{\mathbb{D}^{-\infty}}\langle \delta_y(F_t^{d,\lambda,x}), 1\rangle_{\mathbb{D}^\infty}|_{\eta=\lambda u}$ (with $\lambda u \in (0,1]$) in $\mathcal{E}_{m,t}^{d,\lambda,x,y}$ in (4.3) is given by a linear combination of the expectations of the form

$${}_{\mathbb{D}^{-\infty}}\langle \delta_y(F_t^{d,\lambda u,x}), H_\gamma(F_t^{d,\lambda u,x}, \textstyle\prod_{\ell=1}^k \frac{1}{\beta_\ell!}\partial_\eta^{\beta_\ell} F_t^{d,\eta,x,\gamma_\ell}|_{\eta=\lambda u})\rangle_{\mathbb{D}^\infty}$$

with $k \le m+1$, $\gamma \in \{1,\dots,d\}^k$ and $\beta_1,\dots,\beta_k \ge 1$ such that $\sum_{\ell=1}^k \beta_\ell = m+1$. By the inequality of Lemma 5 with $k=0$ in Appendix A, we have for all $p \ge 1$ and multi-index γ, there are $c > 0$, $p_1, p_2, p_3 > 1$ and $r \in \mathbb{N}$ satisfying

$$\|H_\gamma(F_t^{d,\lambda,x}, G)\|_p \le cd^c \|\det(\sigma^{F_t^{d,\lambda,x}})^{-1}\|_{p_1}^r \|DF_t^{d,\lambda,x}\|_{|\gamma|,p_2,H^d}^{2dr-|\gamma|} \|G\|_{|\gamma|,p_3}, \tag{4.7}$$

for all $G \in \mathbb{D}^\infty$, $t \in (0,T]$, $\lambda \in (0,1]$ and $x \in [a,b]^d$. In order to show the upper bound of the weight appearing in the residual term of the expansion, we list the following results:

Lemma 1.

1. *For all $p > 1$, there exists $\kappa_1 > 0$ such that for all $d \in \mathbb{N}$, $t \in (0,T]$, $x \in [a,b]^d$ and $\lambda \in (0,1]$,*

$$\|\det(\sigma^{F_t^{d,\lambda,x}})^{-1}\|_p \le \kappa_1 d^{\kappa_1} t^{-d}. \tag{4.8}$$

2. *For all $p > 1$, $r \in \mathbb{N}$, there exists $\kappa_2 > 0$ such that for all $d \in \mathbb{N}$, $t \in (0,T]$, $x \in [a,b]^d$ and $\lambda \in (0,1]$,*

$$\|DF_t^{d,\lambda,x}\|_{r,p,H} \le \kappa_2 d_2^{\kappa} t^{1/2}. \tag{4.9}$$

3. *For all $\ell \in \mathbb{N}$, $p > 1$ and $r \in \mathbb{N}$, there exists $\eta > 0$ such that for all $d \in \mathbb{N}$, $t \in (0,T]$, $x \in [a,b]^d$ and $\lambda \in (0,1]$,*

$$\|\partial_\lambda^\ell F_t^{d,\lambda,x}\|_{r,p} \le \eta d^\eta t^{(\ell+1)/2}. \tag{4.10}$$

Proof of Lemma 1. For $d \in \mathbb{N}$, let $V_d : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be such that $V_d = (V_{d,1},\dots,V_{d,d})$ and for $\lambda \in (0,1]$, let $V_d^\lambda : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be such that $V_d^\lambda = (V_{d,1}^\lambda,\dots,V_{d,d}^\lambda)$. Moreover, for $d \in \mathbb{N}$, we use the notation $J_{0\to t} = \frac{\partial}{\partial x}X_t^{d,\lambda,x} = (\frac{\partial}{\partial x_i}X_t^{d,\lambda,x,j})_{1\le i,j\le d}$ for $x \in \mathbb{R}^d$, $t > 0$ and $\lambda \in (0,1]$.

1. Note that for $d \in \mathbb{N}$, $t \in (0,T]$, $x \in \mathbb{R}^d$ and $\lambda \in (0,1]$, we have

$$\begin{aligned}
\sigma^{F_t^{d,\lambda,x}} &= \int_0^t [D_s(X_t^{d,\lambda,x}-x)/\lambda][D_s(X_t^{d,\lambda,x}-x)/\lambda]^\top ds && (4.11)\\
&= \int_0^t J_{0\to t}J_{0\to s}^{-1}V_d(X_s^{d,\lambda,x})V_d(X_s^{d,\lambda,x})^\top {J_{0\to s}^{-1}}^\top J_{0\to t}^\top ds. && (4.12)
\end{aligned}$$

Under the condition $\sigma_d^\lambda(\cdot)\sigma_d^\lambda(\cdot)^\top \ge \lambda^2 I_d$, (i.e. $V_d(\cdot)V_d(\cdot)^\top \ge I_d$) in Assumption 1.3, we have that there is $c > 0$ such that

$$\sup_{x\in[a,b]^d} \|(\det\sigma^{F_t^{d,\lambda,x}})^{-1}\|_p \le cd^c t^{-d}, \tag{4.13}$$

for all $d \in \mathbb{N}$, $t \in (0,T]$ and $\lambda \in (0,1]$, by Theorem 3.5 of Kusuoka and Stroock (1984) [22].

2. We recall that for $d \in \mathbb{N}$, $\lambda \in (0,1]$ and $0 \leq s < t$, $D_s(X_t^{d,\lambda,x} - x)/\lambda = J_{0\to t} J_{0\to s}^{-1} V(X_s^{d,\lambda,x})$. Then, there is $c > 0$ such that

$$\sup_{x \in [a,b]^d} \|DF_t^{d,\lambda,x}\|_{k,p,H^d} \leq cd^c t^{1/2}, \tag{4.14}$$

for all $d \in \mathbb{N}$, $t \in (0,T]$ and $\lambda \in (0,1]$, by Theorem 2.19 of Kusuoka and Stroock (1984) [22].

3. Note that

$$\frac{1}{\ell!}\frac{\partial^\ell}{\partial \lambda^\ell} X_t^{d,\lambda,x,r} = \sum_{i^{(k)},\gamma^{(k)}}^{\ell-1} \int_0^t \prod_{e=1}^k \frac{1}{i_e!}\frac{\partial^{i_e}}{\partial \lambda^{i_e}} X_t^{d,\lambda,x,\gamma_e} \sum_{j=0}^d \partial^{\gamma^{(k)}} V_j^r(X_s^{d,\lambda,x}) dB_s^{d,j} \tag{4.15}$$

$$+\lambda \sum_{i^{(k)},\gamma^{(k)}}^{\ell} \int_0^t \prod_{e=1}^k \frac{1}{i_e!}\frac{\partial^{i_e}}{\partial \lambda^{i_e}} X_t^{d,\lambda,x,\gamma_e} \sum_{j=0}^d \partial^{\gamma^{(k)}} V_j^r(X_s^{d,\lambda,x}) dB_s^{d,j}. \tag{4.16}$$

Since the above is a linear SDE, it has the explicit form and we have

$$\sup_{x\in[a,b]^d} \left\| \frac{1}{\ell!}\frac{\partial^\ell}{\partial\lambda^\ell} X_t^{d,\lambda,x} \right\|_{k,p} \leq cd^c t^{\ell/2}, \tag{4.17}$$

for some $c > 0$ independent of t and d, due to the result:

$$\sup_{x\in[a,b]^d} \Big\| \sum_{i^{(k)},\gamma^{(k)}}^{\ell-1} \int_0^t J_{0\to t} J_{0\to s}^{-1} \prod_{e=1}^k \frac{1}{i_e!}\frac{\partial^{i_e}}{\partial\lambda^{i_e}} X_t^{d,\lambda,x,\gamma_e} \sum_{j=0}^d \partial^{\gamma^{(k)}} V_j(X_s^{d,\lambda,x}) dB_s^{d,j} \Big\|_{k,p} \leq cd^c t^{\ell/2}, \tag{4.18}$$

which is obtained by using Lemma 6 and Lemma 7 in Appendix A. Then, the process

$$\frac{1}{\ell!}\frac{\partial^\ell}{\partial\lambda^\ell} F_t^{d,\lambda,x} = \sum_{i^{(k)},\gamma^{(k)}}^{\ell} \int_0^t \prod_{e=1}^k \frac{1}{i_e!}\frac{\partial^{i_e}}{\partial\lambda^{i_e}} X_t^{d,\lambda,x,\gamma_e} \sum_{j=0}^d \partial^{\gamma^{(k)}} V_j(X_s^{d,\lambda,x}) dB_s^{d,j}, \quad t \geq 0, x \in \mathbb{R}^d \tag{4.19}$$

satisfies

$$\sup_{x\in[a,b]^d} \left\| \frac{1}{\ell!}\frac{\partial^\ell}{\partial\lambda^\ell} F_t^{d,\lambda,x} \right\|_{k,p} \leq cd^c t^{(\ell+1)/2}, \tag{4.20}$$

for some $c > 0$ independent of t and d. □

Using above, we have that for all $k \leq m+1$, $\gamma \in \{1,\ldots,d\}^k$ and $\beta_1,\ldots,\beta_k \geq 1$ such that $\sum_{\ell=1}^k \beta_\ell = m+1$, $p > 1$ and multi-index γ, there exists $\nu > 0$ such that

$$\|H_\gamma(F_t^{d,\lambda,x}, \textstyle\prod_{\ell=1}^k \frac{1}{\beta_\ell!}\partial_\lambda^{\beta_\ell} F_t^{d,\lambda,x,\gamma_\ell})\|_p \leq \nu d^\nu t^{-k/2} t^{(\beta_1+\cdots+\beta_k+k)/2} = \nu d^\nu t^{(m+1)/2}, \tag{4.21}$$

for all $t \in (0,T]$, $x \in [a,b]^d$ and $\lambda \in (0,1]$. Let us define $r_{m,t}^{d,\lambda,x}$ for $t \in (0,T]$, $x \in [a,b]^d$ and $\lambda \in (0,1]$ from (4.1) and (4.6) as

$$\begin{aligned} r_{m,t}^{d,\lambda,x} &= E[f_d(X_t^{d,\lambda,x})] \\ &- E\Big[f_d(\bar{X}_t^{d,\lambda,x})\Big\{1 + \sum_{j=1}^m \lambda^j \sum_{\beta^{(k)},\gamma^{(k)}}^{(j)} H_{\gamma^{(k)}}\Big(\sum_{i=0}^d V_{d,i}(x) B_t^{d,i}, \prod_{\ell=1}^k \sum_{|\alpha|=\beta_\ell} L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}} V_{d,\alpha_r}^{\gamma_\ell}(x)\mathbb{B}_t^{d,\alpha}\Big)\Big\}\Big] \\ &= \lambda^{m+1}\int_0^1 \frac{(1-u)^m}{m!} E[f_d(\tilde{X}_t^{d,\lambda,u,x})\mathcal{W}_{m+1,t}^{d,\lambda,u,x}]du, \end{aligned} \tag{4.22}$$

where $\tilde{X}_t^{d,\lambda,u,x} = x + \lambda F_t^{d,\lambda u,x}$, $u \in [0,1]$ and

$$\mathcal{W}_{m+1,t}^{d,\lambda,u,x} = \sum_{\beta^{(k)},\gamma^{(k)}}^{[m+1]} H_\gamma(F_t^{d,\lambda u,x}, \prod_{\ell=1}^{k} \frac{1}{\beta_\ell!} \partial_\eta^{\beta_\ell} F_t^{d,\eta,x,\gamma_\ell}|_{\eta=\lambda u}), \quad u \in [0,1], \tag{4.23}$$

with $\sum_{\beta^{(k)},\gamma^{(k)}}^{[m+1]} := (m+1)! \sum_{k=1}^{j} \sum_{\beta^{(k)}=(\beta_1,\ldots,\beta_k) s.t. \sum_{\ell=1}^{k}\beta_\ell=j, \beta_i \geq 1} \sum_{\gamma^{(k)}=(\gamma_1,\ldots,\gamma_k)\in\{1,\cdots,d\}^k} \frac{1}{k!}$. Here, $X_t^{d,\lambda,u,x}$, $u \in [0,1]$ and $\mathcal{W}_{m+1,t}^{d,\lambda,u,x}$, $u \in [0,1]$ satisfy that for $p \geq 1$, there exists $\eta > 0$ such that

$$\sup_{x\in[a,b]^d, u\in[0,1]} \|X_t^{d,\lambda,u,x}\|_p \leq \eta d^\eta \text{ and } \sup_{x\in[a,b]^d,u\in[0,1]} \|\mathcal{W}_{m+1,t}^{d,\lambda,u,x}\|_p \leq \eta d^\eta t^{(m+1)/2}$$

for all $\lambda \in (0,1]$ and $t > 0$. Therefore, there exists $c > 0$ such that

$$\sup_{x\in[a,b]^d} |r_{m,t}^{d,\lambda,x}| \leq c d^c \lambda^{m+1} t^{(m+1)/2}, \tag{4.24}$$

for all $\lambda \in (0,1]$ and $t \in (0,T]$, and then the assertion of Proposition 1 holds.

4.2 Proof of Proposition 2

For $d \in \mathbb{N}$ and for $m \in \mathbb{N}$, first note that the following representation holds:

$$E\Big[f_d(\bar{X}_t^{d,\lambda,x}) H_\gamma\Big(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}, \prod_{\ell=1}^{k} \sum_{|\alpha|=\beta_\ell} L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}} V_{d,\alpha_r}^{\gamma_\ell}(x) \mathbb{B}_t^{d,\alpha}\Big)\Big] \tag{4.25}$$

$$= \int_{\mathbb{R}^d} f_d(x+\lambda y)_{\mathbb{D}^{-\infty}} \Big\langle \delta_y \Big(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}\Big) \tag{4.26}$$

$$H_\gamma\Big(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}, \prod_{\ell=1}^{k} \sum_{|\alpha|=\beta_\ell} L_{d,\alpha_1}\cdots L_{d,\alpha_{r-1}} V_{d,\alpha_r}^{\gamma_\ell}(x) \mathbb{B}_t^{d,\alpha}\Big)\Big\rangle_{\mathbb{D}^\infty} dy, \tag{4.27}$$

for $t \in (0,T]$, $x \in \mathbb{R}^d$, $\lambda \in (0,1]$, $k = 1,\ldots,j \leq m$, $\beta_1,\ldots,\beta_k \geq 2$ such that $\beta_1+\cdots+\beta_k = j+k$, and $\gamma \in \{1,\ldots,d\}^k$. Using the Itô formula for the products of iterated integrals (Proposition 5.2.3 of [21] for example) and the formula from (3.14): for a multi-index $\gamma \in \{1,\ldots,d\}^p$ and a multi-index $\alpha \in \{0,1,\ldots,d\}^q$,

$$_{\mathbb{D}^{-\infty}}\langle \delta_y(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}), H_\gamma(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}, \mathbb{B}_t^{d,\alpha})\rangle_{\mathbb{D}^\infty}$$

$$= {}_{\mathbb{D}^{-\infty}}\Big\langle \delta_y(\sum_{i=0}^{d} V_{d,i}(x) B_t^{d,i}), \sum_{j_1,\ldots,j_{|\gamma|},\beta_1,\ldots,\beta_{|\gamma|}=1}^{d} \frac{1}{t^{|\gamma|}} \prod_{q=1}^{|\gamma|} [A_d^{-1}]_{\gamma_q,j_q}(x) V_{d,\beta_q}^{j_q}(x) \frac{1}{k!} \mathbf{B}_t^{d,(\alpha_1,\ldots,\alpha_k,\beta_1,\ldots,\beta_{|\gamma|})}\Big\rangle_{\mathbb{D}^\infty}$$

iteratively, we have (3.15) and the representation (3.16).

We can see that for $p \geq 1$ and $e = 1,\ldots,n(m)$, $\|g_e(t)\mathrm{Poly}_e(B_t^d)\|_p = O(t^{\nu_r/2})$ for some $\nu_r \geq 1$, and by Assumption 1 and 2 and the expression of h_e, there is $\eta > 0$ independent of d such that $|h_e(x)| \leq \eta d^\eta$ for all $e = 1,\ldots,n(m)$ and $x \in [a,b]^d$. Then, for $p \geq 1$, there exists $c > 0$ independent of d such that

$$\|\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)\|_p \leq c d^c, \tag{4.28}$$

uniformly in $(t,x) \in (0,T] \times [a,b]^d$ and $\lambda \in (0,1]$.

4.3 Proof of Theorem 1

The first statement is immediately obtained by combining Proposition 1 with 2:

$$\sup_{x\in[a,b]^d} |u_\lambda^d(t,x) - E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)]| = O(\lambda^{m+1}t^{(m+1)/2}). \tag{4.29}$$

Hereafter, we fix $t\in(0,T]$ and $\lambda\in(0,1]$. For $d\in\mathbb{N}$, $x\in\mathbb{R}^d$, $\delta\in(0,1)$, let

$$\bar{X}_t^{d,\lambda,x,\delta} = x + \lambda\textstyle\sum_{i=0}^d V_{d,i}^\delta(x)B_t^{d,i} \tag{4.30}$$

and $\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d)\in\mathbb{D}^\infty(\Omega^d)$ be a functional which has the form:

$$\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d) = 1 + \textstyle\sum_{e\leq n(m)}\lambda^{p(e)}g_e(t)h_e^\delta(x)\mathrm{Poly}_e(B_t^d), \tag{4.31}$$

where $h_e^\delta:\mathbb{R}^d\to\mathbb{R}$, $e=1,\ldots,n(m)$ are functions constructed by some products of $A_{d,\delta}^{-1}$, $\{V_{d,i}^\delta\}_{0\leq i\leq d}$ and $\{V_{d,i,\alpha}^\delta\}_{0\leq i\leq d,\alpha\in\{1,\ldots,d\}^\ell,\ell\leq 2m}$ in Assumption 2, by replacing with A_d^{-1}, $\{V_{d,i}\}_{0\leq i\leq d}$ and $\{V_{d,i,\alpha}\}_{0\leq i\leq d,\alpha\in\{1,\ldots,d\}^\ell,\ell\leq 2m}$ in Proposition 2, satisfying

$$\begin{aligned}
&E[f_d(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d)]\\
=&E\Big[f_d(\bar{X}_t^{d,\lambda,x,\delta})\Big\{1+\sum_{j=1}^m\lambda^j\sum_{k=1}^j\sum_{\beta_1+\cdots+\beta_k=j+k,\beta_i\geq 2}\sum_{(\gamma_1,\cdots,\gamma_k)\in\{1,\cdots,d\}^k}\frac{1}{k!}\\
&\quad H_{(\gamma_1,\ldots,\gamma_k)}\Big(\sum_{i=1}^d V_{d,i}^\delta(x)B_t^{d,i},\prod_{\ell=1}^k\sum_{|\alpha|=\beta_\ell}L_{d,\alpha_1}^\delta\cdots L_{d,\alpha_{r-1}}^\delta V_{d,\alpha_r}^{\delta,\gamma_\ell}(x)\mathbb{B}_t^{d,\alpha}\Big)\Big\}\Big].
\end{aligned}\tag{4.32}$$

Next, we prepare the following lemmas (Lemma 2, Lemma 3 and Lemma 4) to prove the second assertion ((3.20)) in Theorem 1.

Lemma 2. *There exists $c_1>0$ which depends only on a,b,C,m,κ,t and λ such that for all $\varepsilon\in(0,1)$, $d\in\mathbb{N}$, $\delta=O(\varepsilon^{c_1}d^{-c_1})$,*

$$\sup_{x\in[a,b]^d}|E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d)]|\leq\varepsilon, \tag{4.33}$$

where $f_d^\delta=\mathcal{R}(\psi_\delta^{f_d})\in C(\mathbb{R}^d,\mathbb{R})$ is defined in Assumption 2.4.

Proof of Lemma 2. In the proof, we use a generic constant $c>0$ which depends only on a,b,C,m,κ,t and λ. Note that for $x\in[a,b]^d$,

$$\begin{aligned}
&|E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d)]|\\
\leq\;&|E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)]|\\
&+|E[f_d(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)]|\\
&+|E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^{m,\delta}(t,x,B_t^d)]|.
\end{aligned}\tag{4.34}$$

By 2 of Assumption 2 (with Assumption 1), it holds that

$$\begin{aligned}
&|E[f_d(\bar{X}_t^{d,\lambda,x})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)]|\\
&\leq C\|\bar{X}_t^{d,\lambda,x}-\bar{X}_t^{d,\lambda,x,\delta}\|_2\|\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)\|_2\leq\delta cd^c,
\end{aligned}\tag{4.35}$$

for all $x\in[a,b]^d$. By 4 of Assumption 2 (with Assumption 1), it holds that

$$|E[f_d(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)] - E[f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta})\mathcal{M}_{d,\lambda}^m(t,x,B_t^d)]|\leq\delta cd^c, \tag{4.36}$$

for all $x \in [a,b]^d$. Here, the estimate $\|\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)\|_2 \le cd^c$ in (3.18) is used in (4.35) and (4.36). By 2, 3, 4 of Assumption 2 (with Assumption 1), (3.16) and (4.31), we have that for $p \ge 1$,

$$\|\mathcal{M}^m_{d,\lambda}(t,x,B^d_t) - \mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)\|_p \le \delta c d^c \tag{4.37}$$

and

$$|E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)] - E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)]| \le \delta c d^c, \tag{4.38}$$

for all $x \in [a,b]^d$. Then, by taking $\delta = (1/3)c_1^{-1}\varepsilon^{c_1}d^{-c_1}$ with $c_1 = \max\{1,c\}$ where c is the maximum constant appearing in (4.35), (4.36) and (4.38)), we have

$$\sup_{x\in[a,b]^d} |E[f_d(\bar{X}^{d,\lambda,x}_t)\mathcal{M}^m_{d,\lambda}(t,x,B^d_t)] - E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)]| \le \varepsilon. \ \square \tag{4.39}$$

Lemma 3. *For $d \in \mathbb{N}$, $t \in (0,T]$ and $M \in \mathbb{N}$, let $B^{d,(\ell)}_t$, $\ell = 1,\dots,M$ be independent identically distributed random variables such that $B^{d,(\ell)}_t \overset{\text{law}}{=} B^d_t$. There exists $c_2 > 0$ which depends only on a,b,C,m,κ,t and λ such that for $\varepsilon \in (0,1)$, $d \in \mathbb{N}$ and $M = O(\varepsilon^{-c_2}d^{c_2})$, there is $\omega_{\varepsilon,d} \in \Omega^d$ satisfying*

$$\sup_{x\in[a,b]^d}\left|E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)] - \frac{1}{M}\sum_{\ell=1}^M f^\delta_d(\bar{X}^{d,\lambda,x,\delta,(\ell)}_t(\omega_{\varepsilon,d}))\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^{d,(\ell)}_t(\omega_{\varepsilon,d}))\right| \le \varepsilon, \tag{4.40}$$

where $\delta = O(\varepsilon^{c_1}d^{-c_1})$ with the constant c_1 in Lemma 2.

Proof of Lemma 3. There exists a constant $c > 0$ which depends only on a,b,C,m,κ,t and λ such that for all $x \in [a,b]^d$ and $M \in \mathbb{N}$,

$$E\left[\left|E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)] - \frac{1}{M}\sum_{\ell=1}^M f^\delta_d(\bar{X}^{d,\lambda,x,\delta,(\ell)}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^{d,(\ell)}_t)\right|^2\right] \tag{4.41}$$

$$\le \frac{1}{M}E[|f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)|^2] \le \frac{cd^c}{M}. \tag{4.42}$$

Then, by choosing $c_2 = \max\{1,c\}$, we have that for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$ and $M = c_2\varepsilon^{-c_2}d^{c_2}$,

$$E\left[\left|E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)] - \frac{1}{M}\sum_{\ell=1}^M f^\delta_d(\bar{X}^{x,\delta,(\ell)}_t)\mathcal{M}^{m,\delta}_{d}(t,x,B^{(\ell)}_t)\right|^2\right]^{1/2} \le \varepsilon, \tag{4.43}$$

for all $x \in [a,b]^d$, and therefore, there is $\omega_{\varepsilon,d} \in \Omega^d$ satisfying

$$\sup_{x\in[a,b]^d}\left|E[f^\delta_d(\bar{X}^{d,\lambda,x,\delta}_t)\mathcal{M}^{m,\delta}_{d,\lambda}(t,x,B^d_t)] - \frac{1}{M}\sum_{\ell=1}^M f^\delta_d(\bar{X}^{x,\delta,(\ell)}_t(\omega_{\varepsilon,d}))\mathcal{M}^{m,\delta}_{d}(t,x,B^{(\ell)}_t(\omega_{\varepsilon,d}))\right| \le \varepsilon. \ \square \tag{4.44}$$

Lemma 4. *For $d \in \mathbb{N}$, $t \in (0,T]$ and $M \in \mathbb{N}$, let $B^{d,(\ell)}_t$, $\ell = 1,\dots,M$ be independent identically distributed random variables such that $B^{d,(\ell)}_t \overset{\text{law}}{=} B^d_t$. There exist $\{\phi_{\varepsilon,d}\}_{\varepsilon\in(0,1),d\in\mathbb{N}} \subset \mathcal{N}$ and $c > 0$ (which depends only on a,b,C,m,κ,t and λ) such that for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, we have $\mathcal{C}(\phi_{\varepsilon,d}) \le c\varepsilon^{-c}d^c$, and for a realization $\omega_{\varepsilon,d} \in \Omega^d$ given in Lemma 3, it holds that*

$$\sup_{x\in[a,b]^d}\left|\frac{1}{M}\sum_{\ell=1}^M f^\delta_d(\bar{X}^{d,\lambda,x,\delta,(\ell)}_t(\omega_{\varepsilon,d}))\mathcal{M}^{m,\delta}_{d}(t,x,B^{d,(\ell)}_t(\omega_{\varepsilon,d})) - \mathcal{R}(\phi_{\varepsilon,d})(x)\right| \le \varepsilon, \tag{4.45}$$

where $\delta = O(\varepsilon^{c_1}d^{-c_1})$ and $M = O(\varepsilon^{-c_2}d^{c_2})$ with the constants c_1 and c_2 in Lemma 2 and Lemma 3.

Proof of Lemma 4. In the proof, we use a generic constant $c > 0$ which depends only on a, b, C, m, κ, t and λ. Let $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, $\ell = 1, \ldots, M$, let $\delta = O(\varepsilon^{c_1} d^{-c_1})$, $M = O(\varepsilon^{-c_2} d^{c_2})$ where c_1 and c_2 are the constants appearing in Lemma 2 and Lemma 3, let $\omega_{\varepsilon,d}$ be a realization given in Lemma 3, and let $b^{d,(\ell)} = B_t^{d,(\ell)}(\omega_{\varepsilon,d})$. Since there exists $\eta_{\delta,d}^{(\ell)} \in \mathcal{N}$ such that $\mathcal{R}(\eta_{\delta,d}^{(\ell)})(x) = x + \lambda \mathcal{R}(\psi_{\delta,d}^{V_0})(x) t + \lambda \sum_{i=1}^d \mathcal{R}(\psi_{\delta,d}^{V_i})(x) b^{d,(\ell),i}$ for $x \in \mathbb{R}^d$ and $\mathcal{C}(\eta_{\delta,d}^{(\ell)}) = O(\delta^{-c} d^c)$ (by Lemma 9 in Appendix B), there exists $\psi_{1,(\ell)}^{\delta,d} \in \mathcal{N}$ such that $\mathcal{R}(\psi_{1,(\ell)}^{\delta,d})(x) = \mathcal{R}(\psi_{\delta,d}^{f})(\mathcal{R}(\eta_{\delta,d}^{(\ell)})(x)) = f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta}(\omega_{\varepsilon,d}))$ for $x \in \mathbb{R}^d$ and $\mathcal{C}(\psi_{1,(\ell)}^{\delta,d}) = O(\delta^{-c} d^c)$ (by Lemma 10 in Appendix B). Next, we recall that by (4.31), the weight $\mathcal{M}_{d,\lambda}^{m,\delta}(t, x, b^{d,(\ell)})$, $x \in \mathbb{R}^d$ has the form $\mathcal{M}_{d,\lambda}^{m,\delta}(t, x, b^{d,(\ell)}) = 1 + \sum_{e \leq n(m)} \lambda^{p(e)} g_e(t) h_e^\delta(x) \mathrm{Poly}_e(b^{d,(\ell)})$ constructed by some products of $A_{d,\delta}^{-1}$, $\{V_{d,i}^\delta\}_{0 \leq i \leq d}$ and $\{V_{d,i,\alpha}^\delta\}_{0 \leq i \leq d, \alpha \in \{1,\ldots,d\}^\ell, \ell \leq 2m}$ in Assumption 2. Using Lemma 12, Lemma 9 in Appendix B and Assumption 2, there is a neural network $\psi_{2,(\ell)}^{\varepsilon,d} \in \mathcal{N}$ such that $\sup_{x \in [a,b]^d} |\mathcal{M}_{d,\lambda}^{m,\delta}(t, x, b^{d,(\ell)}) - \mathcal{R}(\psi_{2,(\ell)}^{\varepsilon,d})(x)| \leq \varepsilon/2$ and $\mathcal{C}(\psi_{2,(\ell)}^{\varepsilon,d}) = O(\varepsilon^{-c} d^c)$. Hence, we have

$$\sup_{x \in [a,b]^d} |f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta,(\ell)}(\omega_{\varepsilon,d})) \mathcal{M}_{d,\lambda}^{m,\delta}(t, x, b^{d,(\ell)}) - \mathcal{R}(\psi_{1,(\ell)}^{\delta,d})(x) \mathcal{R}(\psi_{2,(\ell)}^{\varepsilon,d})(x)| \leq \varepsilon/2. \tag{4.46}$$

We again use Lemma 12 in Appendix B to see that there exists $\Psi_{(\ell)}^{\varepsilon,d} \in \mathcal{N}$ such that

$$|\mathcal{R}(\psi_{1,(\ell)}^{\delta,d})(x) \mathcal{R}(\psi_{2,(\ell)}^{\varepsilon,d})(x) - \mathcal{R}(\Psi_{(\ell)}^{\varepsilon,d})(x)| \leq \varepsilon/2, \tag{4.47}$$

for all $x \in [a,b]^d$, and $\mathcal{C}(\Psi_{(\ell)}^{\varepsilon,d}) = O(\varepsilon^{-c} d^c)$. Finally, applying Lemma 9 gives the desired result, i.e. there exist $\{\phi_{\varepsilon,d}\}_{\varepsilon \in (0,1), d \in \mathbb{N}} \subset \mathcal{N}$ and $c > 0$ such that for all $\varepsilon \in (0,1)$, $d \in \mathbb{N}$, we have $\mathcal{C}(\phi^{\varepsilon,d}) \leq c \varepsilon^{-c} d^c$, and for a realization $\omega_{\varepsilon,d} \in \Omega^d$ given in Lemma 3, it holds that

$$\sup_{x \in [a,b]^d} \Big| \frac{1}{M} \sum_{\ell=1}^M f_d^\delta(\bar{X}_t^{d,\lambda,x,\delta,(\ell)}(\omega_{\varepsilon,d})) \mathcal{M}_d^{m,\delta}(x, B_t^{d,(\ell)}(\omega_{\varepsilon,d})) - \mathcal{R}(\phi_{\varepsilon,d})(x) \Big| \leq \varepsilon. \quad \square \tag{4.48}$$

Proof of Theorem 1. The first assertion (in (3.19)) follows from (4.29). The second assertion (in (3.20)) is obtained by combining Lemma 2, Lemma 3 and Lemma 4. □

5 Deep learning implementation

We briefly provide the implementation scheme for the approximation in Theorem 1. Let ξ be a uniformly distributed random variable, i.e. $\xi \in U([a,b]^d)$, and define $\mathbb{X}_t^\xi = \xi + \lambda \sum_{i=0}^d V_{i,d}(\xi) B_t^{i,d}$, $t \geq 0$. For $t > 0$, the m-th order asymptotic expansion of Theorem 1 can be represented by

$$u^m(t, \cdot) = \mathrm{argmin}_{\psi \in C([a,b]^d)} E[|\psi(\xi) - f(\mathbb{X}_t^\xi) \mathcal{M}_{d,\lambda}^m(t, \xi, B_t^d)|^2], \tag{5.1}$$

which is obtained by Theorem 1 of this paper combining with Proposition 2.2 of Beck et al. (2021) [2]. We construct a deep neural network $u^{\mathcal{NN},\theta^*}(t, \cdot)$ to approximate the function $u^m(t, \cdot)$ given by for a depth $L \in \mathbb{N}$ and $N_0, N_1, \ldots, N_L \in \mathbb{N}$,

$$u^{\mathcal{NN},\theta}(t, x) = \mathcal{A}_{W_L^\theta, B_L^\theta} \circ \varrho_{N_{L-1}} \circ \mathcal{A}_{W_{L-1}^\theta, B_{L-1}^\theta} \circ \cdots \circ \varrho_{N_1} \circ \mathcal{A}_{W_1^\theta, B_1^\theta}(x), \ x \in \mathbb{R}^d, \tag{5.2}$$

where $\mathcal{A}_{W_k^\theta, B_k^\theta}(x) = W_k^\theta x + B_k^\theta$, $x \in \mathbb{R}^{N_{k-1}}$, $k = 1, \ldots, L$ with $((W_1^\theta, B_1^\theta), \ldots, (W_L^\theta, B_L^\theta)) \in \mathcal{N}_L^{N_0, N_1, \ldots, N_L}$ given by

$$\mathcal{A}_{W_k^\theta, B_k^\theta}(x) = \begin{pmatrix} \theta^{q+1} & \cdots & \theta^{q+N_{k-1}} \\ \vdots & \ddots & \vdots \\ \theta^{q+(N_k-1)N_{k-1}+1} & \cdots & \theta^{q+N_k N_{k-1}} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{N_{k-1}} \end{pmatrix} + \begin{pmatrix} \theta^{q+N_k N_{k-1}+1} \\ \vdots \\ \theta^{q+N_k N_{k-1}+N_k} \end{pmatrix}, \tag{5.3}$$

and the optimized parameter θ^* obtained by the following minimization problem:

$$\theta^* = \mathrm{argmin}_{\theta \in \mathbb{R}^{\sum_{\ell=1}^{L} N_\ell(N_{\ell-1}+1)}} E[|u^{\mathcal{NN},\theta}(t,\xi) - f(\mathbb{X}_t^\xi)\mathcal{M}_{d,\lambda}^m(t,\xi,B_t^d)|^2]. \tag{5.4}$$

In the implementation of the deep neural network approximation, we use stochastic gradient descent method and the Adam optimizer [20] as in Section 3 and 4 of Beck et al. (2021) [2]. In Appendix C, we list the sample code of the scheme for a high-dimensional PDE with a nonlinear coefficient in Section 6.2 (which includes linear coefficient case).

6 Numerical examples

In the section, we perform numerical experiments in order to demonstrate the accuracy of our scheme. We compare the deep learning method of Beck et al. (2021) [2] where the Euler-Maruyama scheme is used with the stochastic gradient descent method with the Adam optimizer. All experiments are performed in Google Colaboratory using Tensorflow.

6.1 High-dimensional Black-Scholes model

6.1.1 Uncorrelated case

First, we examine our scheme for a high-dimensional Black-Scholes model (geometric Brownian motion) whose corresponding PDE is given by

$$\partial_t u_\lambda^d(t,x) = \lambda \sum_{i=1}^d \mu x_i \frac{\partial}{\partial x_i} u_\lambda^d(t,x) + \frac{\lambda^2}{2} \sum_{i=1}^d c_i^2 x_i^2 \frac{\partial^2}{\partial x_i^2} u_\lambda^d(t,x), \quad u_\lambda^d(0,x) = f_d(x), \tag{6.1}$$

where $f_d(x) = \max\{\max\{x_1 - K\}, \dots, \max\{x_d - K\}\}$. Let $d = 100$, $t = 1.0$, $a = 99.0$, $b = 101.0$, $K = 100.0$, $\lambda = 0.3$, $\mu = 1/30$ (or $r := \lambda \times \mu = 0.01$), $c_i = 1.0$ (or $\sigma_i := \lambda \times c_i = 0.3$), $i = 1, \dots, 100$. We approximate the function $u_\lambda^d(t,\cdot)$ (or the maximum option price $e^{-rt}u_\lambda^d(t,\cdot)$ in financial mathematics) on $[a,b]^d$ by constructing a deep neural network (1 input layer with d-neurons, 2 hidden layers with $2d$-neurons each and 1 output layer with 1-neuron) based on Theorem 1 with $m = 1$ and Section 5. For the experiment, we use the batch size $M = 1,024$, the number of iteration steps $J = 5,000$ and the learning rate $\gamma(j) = 10^{-1}\mathbf{1}_{[0,0.3J]}(j) + 10^{-2}\mathbf{1}_{(0.3J,0.6J]}(j) + 10^{-3}\mathbf{1}_{(0.6J,J]}(j)$, $j \le J$ for the stochastic gradient descent method. After we estimate the function $u_\lambda^d(t,\cdot)$, we input $x_0 = (100.0, \dots, 100.0) \in [a,b]^d$ to check the accuracy. We compute the mean of 10 independent trials and estimate the relative error, i.e. $|(u_\lambda^{deep,d}(t,x_0) - u_\lambda^{ref,d}(t,x_0))/u_\lambda^{ref,d}(t,x_0)|$ where the reference value $u_\lambda^{ref,d}(t,x_0)$ is computed by the Itô formula with Monte-Carlo method with 10^7-paths. The same experiment is applied to the method of Beck et al. (2021) [2]. Table 1 provides the numerical results (the relative errors and the runtimes) for AE $m = 1$ and the method in Beck et al. (2021) [2] with the Euler-Maruyama discretization $n = 16$, 32 (Beck et al. $n = 16$, Beck et al. $n = 32$ in the table).

Table 1: Comparison in deep learning methods for $d = 100$

	AE $m = 1$	Beck et al. $n = 16$	Beck et al. $n = 32$
Relative error	0.0048	0.0056	0.0017
Runtime	75.49s	217.79s	352.79s

6.1.2 Correlated case

We next provide a numerical example for a Black-Scholes model with correlated noise in high-dimension. Let us consider the following PDE:

$$\partial_t u_\lambda^d(t,x) = \lambda \sum_{i=1}^d \mu x_i \frac{\partial}{\partial x_i} u_\lambda^d(t,x) + \frac{\lambda^2}{2} \sum_{i,j,k=1}^d \sigma_k^i \sigma_k^j x_i x_j \frac{\partial^2}{\partial x_i \partial x_j} u_\lambda^d(t,x), \quad u_\lambda^d(0,x) = f_d(x), \tag{6.2}$$

where $f_d(x) = \max\{K - \frac{1}{d}\sum_{i=1}^d x_i, 0\}$ and $\sigma = [\sigma_k^j]_{k,j} \in \mathbb{R}^{d\times d}$ satisfies $\sigma_{ij} = 0$ for $i < j$, $\sigma_{ii} > 0$ for $i = 1, \ldots, d$ and

$$\sigma\sigma^\top = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \rho & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & 1 \end{pmatrix} \in \mathbb{R}^{d\times d}. \tag{6.3}$$

Let $d = 100$, $t = 1.0$, $a = 99.0$, $b = 101.0$, $K = 90.0$, $\lambda = 0.3$, $\mu = 0.0$, $\rho = 0.5$. We approximate the function $u_\lambda^d(t,\cdot)$ (the basket option price in financial mathematics) on $[a,b]^d$ by constructing a deep neural network (1 input layer with d-neurons, 2 hidden layers with $2d$-neurons each and 1 output layer with 1-neuron) based on Theorem 1 ($m = 1$) with the expansion technique of the basket option price given in Section 3.1 of Takahashi (1999) [32] and Section 5. For the experiment, we use the batch size $M = 1,024$, the number of iteration steps $J = 5,000$ and the learning rate $\gamma(j) = 5.0\times 10^{-2}\mathbf{1}_{[0,0.3J]}(j) + 5.0\times 10^{-3}\mathbf{1}_{(0.3J,0.6J]}(j) + 5.0\times 10^{-4}\mathbf{1}_{(0.6J,J]}(j)$, $j \le J$ for the stochastic gradient descent method. After we estimate the function $u_\lambda^d(t,\cdot)$, we input $x_0 = (100.0,\ldots,100.0) \in [a,b]^d$ to check the accuracy. We compute the mean of 10 independent trials and estimate the relative error, i.e. $|(u_\lambda^{deep,d}(t,x_0) - u_\lambda^{ref,d}(t,x_0))/u_\lambda^{ref,d}(t,x_0)|$ where the reference value $u_\lambda^{ref,d}(t,x_0)$ is computed by the Itô formula with Monte-Carlo method with 10^7-paths. The same experiment is applied to the method of Beck et al. (2021) [2]. Table 2 provides the numerical results (the relative errors and the runtimes) for AE $m = 1$ and the method in Beck et al. (2021) [2] with the Euler-Maruyama discretization $n = 32$, 64 (Beck et al. $n = 32$, Beck et al. $n = 64$ in the table).

Table 2: Comparison in deep learning methods for $d = 100$

	AE $m = 1$	Beck et al. $n = 32$	Beck et al. $n = 64$
Relative error	0.0039	0.0042	0.0035
Runtime	83.56s	470.73s	848.43s

6.2 High-dimensional CEV model (nonlinear volatility case)

We consider a Kolmogorov PDE with nonlinear diffusion coefficients whose corresponding stochastic process is called the CEV model:

$$\partial_t u_\lambda^d(t,x) = \lambda\sum_{i=1}^d \mu x_i \frac{\partial}{\partial x_i}u_\lambda^d(t,x) + \frac{\lambda^2}{2}\sum_{i=1}^d \gamma_i^2 c_i^2 x_i^{2\beta_i}\frac{\partial^2}{\partial x_i^2}u_\lambda^d(t,x), \quad u_\lambda^d(0,x) = f_d(x), \tag{6.4}$$

where $f_d(x) = \max\{\max\{x_1 - K\},\ldots,\max\{x_d - K\}\}$. Let $d = 100$, $t = 1.0$, $a = 99.0$, $b = 101.0$, $K = 100.0$, $\lambda = 0.3$, $\mu = 1/30$ (or $r := \lambda\times\mu = 0.01$), $\beta_i = 0.5$, $\gamma_i = K^{1-\beta_i}$, $c_i = 1.0$ (or $\sigma_i := \lambda\times c_i = 0.3$), $i = 1,\ldots,d$. We approximate the function $u_\lambda^d(t,\cdot)$ (or the maximum option price $e^{-rt}u_\lambda^d(t,\cdot)$) on $[a,b]^d$ by constructing a deep neural network (1 input layer with d-neurons, 2 hidden layers with $2d$-neurons each and 1 output layer with 1-neuron,) based on Theorem 1 with $m = 1$. For the experiment, we use the batch size $M = 1,024$, the number of iteration steps $J = 5,000$ and the learning rate $\gamma(j) = 5.0\times 10^{-1}\mathbf{1}_{[0,0.3J]}(j) + 5.0\times 10^{-2}\mathbf{1}_{(0.3J,0.6J]}(j) + 5.0\times 10^{-3}\mathbf{1}_{(0.6J,J]}(j)$, $j \le J$ for the stochastic gradient descent method. After we estimate the function $u_\lambda^d(t,\cdot)$, we input $x_0 = (100.0,\ldots,100.0) \in [a,b]^d$ to check the accuracy. We compute the mean of 10 independent trials and estimate the relative error, i.e. $|(u_\lambda^{deep,d}(t,x_0) - u_\lambda^{ref,d}(t,x_0))/u_\lambda^{ref,d}(t,x_0)|$ where the reference value $u_\lambda^{ref,d}(t,x_0)$ is computed by Monte-Carlo method with the Euler-Maruyama scheme with time-steps 2^{10} and 10^7-paths. The same experiment is applied to the method of Beck et al. (2021) [2]. Table 3 provides the numerical results (the relative errors and the runtimes) for AE $m = 1$ and the method in Beck et al. (2021) [2] with the Euler-Maruyama discretization $n = 32$, 64 (Beck et al. $n = 32$, Beck et al. $n = 64$ in the table).

Table 3: Comparison in deep learning methods for $d = 100$

	AE $m = 1$	Beck et al. $n = 64$	Beck et al. $n = 128$
Relative error	0.0006	0.0019	0.0006
Runtime	83.09s	764.76s	1265.26s

6.3 High-dimensional Heston model

We finally show an example for a small time asymptotic expansion for a high-dimensional Heston model:

$$\partial_t u_\lambda^{2d}(t,x) = \mathcal{L}^{2d,\lambda} u_\lambda^{2d}(t,x), \quad u_\lambda^{2d}(0,x) = f_{2d}(x), \tag{6.5}$$

where $f_{2d}(x) = \max\{\max\{x_1 - K\}, \ldots, \max\{x_{2d-1} - K\}\}$ and $\mathcal{L}^{2d,\lambda}$ is a generator given by

$$\begin{aligned}\mathcal{L}^{2d,\lambda} =& \lambda \sum_{i=1}^{d} [\kappa_i(\theta_i - x_{2i}) \frac{\partial}{\partial x_{2i}}] \\ &+ \lambda^2 \sum_{i=1}^{d} [\frac{1}{2} x_{2i} x_{2i-1}^2 \frac{\partial^2}{\partial x_{2i-1}^2} + \rho_i \nu_i x_{2i-1} x_{2i} \frac{\partial^2}{\partial x_{2j-1} \partial x_{2i}} + \frac{1}{2} \nu_i^2 x_2 \frac{\partial^2}{\partial x_{2i}^2}].\end{aligned} \tag{6.6}$$

Let $d = 25$ ($2d = 50$), $t = 0.5$, $a = 99.0$, $b = 101.0$, $a' = 0.035$, $b' = 0.045$, $K = 100.0$, $\lambda = 1.0$, $\kappa_i = 1.0$, $\theta_i = 0.04$, $\nu_i = 0.1$, $\rho_i = -0.5$, $i = 1, \ldots, d$. We approximate the function $u_\lambda^d(t, \cdot)$ on $[a, b]^d$ by constructing a deep neural network (1 input layer with $2d$-neurons, 2 hidden layers with $4d$-neurons each and 1 output layer with 1-neuron) based on Theorem 1 with $m = 1$ and Section 5. For the experiment, we use the batch size $M = 1,024$, the number of iteration steps $J = 5,000$ and the learning rate $\gamma(j) = 5.0 \times 10^{-2} \mathbf{1}_{[0,0.3J]}(j) + 5.0 \times 10^{-3} \mathbf{1}_{(0.3J,0.6J]}(j) + 5.0 \times 10^{-4} \mathbf{1}_{(0.6J,J]}(j)$, $j \le J$ for the stochastic gradient descent method. After we estimate the function $u_\lambda^d(t, \cdot)$, we input $x_0 = (100.0, 0.04, \ldots, 100.0, 0.04) \in ([a, b] \times [a', b'])^d$ to check the accuracy. We compute the mean of 10 independent trials and estimate the relative error, i.e. $|(u_\lambda^{deep,d}(t, x_0) - u_\lambda^{ref,d}(t, x_0))/u_\lambda^{ref,d}(t, x_0)|$ where the reference value $u_\lambda^{ref,d}(t, x_0)$ is computed by Monte-Carlo method with the Euler-Maruyama scheme with time-steps 2^{10} and 10^7-paths. The same experiment is applied to the method of Beck et al. (2021) [2]. Table 4 provides the numerical results (the relative errors and the runtimes) for AE $m = 1$ and the method in Beck et al. (2021) [2] with the Euler-Maruyama discretization $n = 16$, 32 (Beck et al. $n = 16$, Beck et al. $n = 32$ in the table).

Table 4: Comparison in deep learning methods for $2d = 50$

	AE $m = 1$	Beck et al. $n = 16$	Beck et al. $n = 32$
Relative error	0.0006	0.0034	0.0007
Runtime	46.96s	119.37s	201.61s

7 Conclusion

In the paper, we introduced a new spatial approximation for solving high-dimensional PDEs without the curse of dimensionality, where an asymptotic expansion method with a deep learning-based algorithm is effectively applied. The mathematical justification for the spatial approximation was provided using Malliavin calculus and ReLU calculus. We checked the effectiveness of our method through numerical examples for high-dimensional Kolmogorov PDEs.

More accurate deep learning-based implementations based on the method of the paper should be studied as a next research topic. We believe that higher order asymptotic expansion or higher order weak approximation (discretization) will give robust computation schemes without the curse of dimensionality, which should be proved mathematically in the future work. Also, applying our method to nonlinear problems as in [14][15] will be a challenging and important task.

A Malliavin calculus

In the following, we provide precise estimates of Wiener functionals, which are useful for proving and computing the deep learning-based approximation with our asymptotic expansion.

Lemma 5. *Let $d \in \mathbb{N}$, $F \in (\mathbb{D}^\infty(\Omega^d))^d$ be a non-degenerate Wiener functional, $G \in \mathbb{D}^\infty(\Omega^d)$, $\alpha = (\alpha_1, \dots, \alpha_\ell) \in \{1, \dots, d\}^\ell$ with length $\ell \in \mathbb{N}$. For $k \in \mathbb{N} \cup \{0\}$ and $p \geq 1$, there exist $c = c(k,p) > 0$, $q_1 = q_1(k,p) > 1$, $q_2 = q_2(k,p,d) > 1$, $q_3 = q_3(k,p) > 1$ and $r = r(k) \in \mathbb{N}$ such that*

$$\|H_\alpha(F,G)\|_{k,p} \leq cd^c \| \det(\sigma^F)^{-1}\|_{q_1}^r \|DF\|_{k+|\alpha|,q_2,H^d}^{2dr-|\alpha|} \|G\|_{k+|\alpha|,q_3}. \tag{A.1}$$

Proof of Lemma 5. For $i \in \{1, \dots, d\}$, we have

$$\|H_{(i)}(F,G)\|_{k,p} \leq \sum_{j=1}^d \|\delta([\sigma^F]_{ij}^{-1} DF^j G)\|_{k,p} \leq c_{k,p} \sum_{j=1}^d \|[\sigma^F]_{ij}^{-1} DF^j G\|_{k+1,p,H^d}, \tag{A.2}$$

for some universal constant $c_{k,p} > 0$. Let p_1 and p_2 be real numbers such that $p_1^{-1} + p_2^{-1} = p^{-1}$. Hereafter, we use a generic constant $C > 0$ such that $C = cd^c$ for some $c > 0$ depending on k and p, whose value varies from line to line. Since it holds that

$$\|[\sigma^F]_{ij}^{-1} DF^j\|_{k+1,p_1,H^d} \leq C \| \det(\sigma^F)^{-1}\|_{2(k+2)p_1}^e \|DF\|_{k+1,2(2d(k+2)-1)p_1,H^d}^{2de-1}, \tag{A.3}$$

for some $e \in \mathbb{N}$ depending on k, we have

$$\|H_{(i)}(F,G)\|_{k,p} \leq C \| \det(\sigma^F)^{-1}\|_{2(k+2)p_1}^e \|DF\|_{k+1,2(2d(k+2)-1)p_1,H^d}^{2de-1} \|G\|_{k+1,p_2}. \tag{A.4}$$

For $\alpha = (\alpha_1, \dots, \alpha_\ell) \in \{1, \dots, d\}^\ell$, we have

$$\begin{aligned} &\|H_{(\alpha_1,\dots,\alpha_\ell)}(F,G)\|_{k,p} = \|H_{(\alpha_\ell)}(F, H_{(\alpha_1,\dots,\alpha_{\ell-1})}(F,G))\|_{k,p} \\ &\leq C \| \det(\sigma^F)^{-1}\|_{2(k+2)p_1}^e \|DF\|_{k+1,2(2d(k+2)-1)p_1,H^d}^{2de-1} \|H_{(\alpha_1,\dots,\alpha_{\ell-1})}(F,G)\|_{k+1,p_2}. \end{aligned} \tag{A.5}$$

Then, iterating this procedure, we have that for $k \in \mathbb{N} \cup \{0\}$ and $p \geq 1$, there exist $q_1, q_2, q_3 > 1$ and $r \in \mathbb{N}$ such that

$$\|H_\alpha(F,G)\|_{k,p} \leq C \| \det(\sigma^F)^{-1}\|_{q_1}^r \|DF\|_{k+|\alpha|,q_2,H^d}^{2dr-|\alpha|} \|G\|_{k+|\alpha|,q_3}. \quad \square \tag{A.6}$$

Lemma 6. *For $d \in \mathbb{N}$, $i = 1, 2$, let $\{G_t^{d,x,i}\}_{t \in (0,T], x \in \mathbb{R}^d} \subset \mathbb{D}^\infty(\Omega^d)$ satisfy that for $k \geq 1$ and $p \in [1, \infty)$, there exist $c_i, s_i > 0$ independent of d such that $\sup_{x \in [a,b]^d} \|G_t^{d,x,i}\|_{k,p} \leq c_i d^{c_i} t^{s_i/2}$ for all $t \in (0,T]$. Then, we have that for $k \geq 1$ and $p \in [1, \infty)$, there exists c independent of d such that for all $t \in (0,T]$, $\sup_{x \in [a,b]^d} \| \prod_{i=1}^2 G_t^{d,x,i}\|_{k,p} \leq rd^r t^{(s_1+s_2)/2}$ and $\sup_{x \in [a,b]^d} \| \sum_{i=1}^2 G_t^{d,x,i}\|_{k,p} \leq cd^c t^{\min\{s_1,s_2\}/2}$.*

Proof of Lemma 6. We only prove the former case. By Proposition 1.5.6 of Nualart [29], for $k \geq 1$ and $p \in [1, \infty)$, $\|\prod_{i=1}^2 G_t^{d,x,i}\|_{k,p} \leq c_{k,p} \|G_t^{d,x,1}\|_{k,p_1} \|G_t^{d,x,2}\|_{k,p_2}$ for some constant $c_{k,p} > 0$ depending only on k and p, where $p_1, p_2 > 1$ satisfies $1/p_1 + 1/p_2 = 1/p$. Then, by the assumptions, $\sup_{x \in [a,b]^d} \| \prod_{i=1}^2 G_t^{d,x,i}\|_{k,p} \leq rd^r t^{(s_1+s_2)/2}$. $\square$

Lemma 7. *For $d \in \mathbb{N}$, let $\{u_t^{d,x}\}_{t \in (0,T], x \in \mathbb{R}^d} \subset \mathbb{D}^\infty(\Omega^d)$ satisfy that for $t \in (0,T]$, $x \in \mathbb{R}^d$, $j = 1, \dots, d$, $\int_0^t u_s^{d,x} dB_s^{d,j} \in \mathbb{D}^\infty(\Omega^d)$ and that for $k \geq 1$ and $p \in [1, \infty)$, there exist $q, \nu > 0$ independent of d such that $\sup_{x \in [a,b]^d} \|u_t^{d,x}\|_{k,p} \leq qd^q t^{\nu/2}$ for all $t \in (0,T]$. Then, for $k \geq 1$ and $p \in [1, \infty)$, there exists $c > 0$ independent of d such that for all $t \in (0,T]$, $\sup_{x \in [a,b]^d} \| \int_0^t u_s^{d,x} dB_s^{d,0}\|_{k,p} \leq cd^c t^{(\nu+2)/2}$ and for $j = 1, \dots, d$, $\sup_{x \in [a,b]^d} \| \int_0^t u_s^{d,x} dB_s^{d,j}\|_{k,p} \leq cd^c t^{(\nu+1)/2}$.*

Proof of Lemma 7. We only prove the latter case. Note that for $r = 1, \dots, k$, $D^r \int_0^t u_s^{d,x} dB_s^{d,j} = D^{r-1} u_\cdot^{d,x} + \int_0^t D^r u_s^{d,x} dB_s^{d,j}$. Then, it holds that $E[\|D^r \int_0^t u_s^{d,x} dB_s^{d,j}\|^p_{(H^d)^{\otimes r}}] = E[\|D^{r-1} u_\cdot^{d,x}\|^p_{(H^d)^{\otimes r}}] + E[\| \int_0^t D^r u_s^{d,x} dB_s^{d,j}\|^p_{(H^d)^{\otimes r}}]$. Here, $E[\|D^{r-1} u_\cdot^{d,x}\|^p_{(H^d)^{\otimes r}}] \leq \eta d^\eta t^{p-1} \int_0^t E[\|D^{r-1} u_s^{d,x}\|^p_{(H^d)^{\otimes (r-1)}}] ds$ for some η (independent of d) and $E[\|\int_0^t D^r u_s^{d,x} dB_s^{d,j}\|^p_{(H^d)^{\otimes r}}] \leq c_p t^{p/2-1} \int_0^t E[\|D^r u_s^{d,x}\|^p_{(H^d)^{\otimes r}}] ds$ for some $c_p > 0$ (independent of d) by Hölder inequality and Burkholder-Davis-Gundy inequality. By the assumptions, $\sup_{x \in [a,b]^d} E[\|D^{r-1} u_\cdot^{d,x}\|^p_{(H^d)^{\otimes r}}] \leq \eta d^\eta t^{p-1} \int_0^t q^p d^{pq} s^{p\nu/2} ds \leq c d^c t^{p(\nu/2+1)}$ and $\sup_{x \in [a,b]^d} E[\|\int_0^t D^r u_s^{d,x} dB_s^{d,j}\|^p_{(H^d)^{\otimes r}}] \leq c_p t^{p/2-1} \int_0^t q^p d^{pq} s^{p\nu/2} ds \leq c d^c t^{p(\nu+1)/2}$. Then, we have $\sup_{x \in [a,b]^d} \| \int_0^t u_s^{d,x} dB_s^{d,j}\|_{k,p} \leq c d^c t^{(\nu+1)/2}$. □

B ReLU calculus

Appendix B gives some results on ReLU calculus which are basic in the analysis of our paper. We prepare the following result from Lemma A.7 of [5].

Lemma 8. *Let $n, d, L \in \mathbb{N}$ and for $i = 1, \dots, n$, let $d_i \in \mathbb{N}$ and $\phi_i \in \mathcal{N}$ with $\mathcal{L}(\phi_i) = L$, $\dim_{\text{in}}(\phi_i) = d$ and $\dim_{\text{out}}(\phi_i) = d_i$. Then, there exists $\psi \in \mathcal{N}$ such that $\mathcal{L}(\psi) = L$, $\mathcal{C}(\psi) \leq \sum_{i=1}^n \mathcal{C}(\phi_i)$, $\dim_{\text{in}}(\psi) = d$ and $\dim_{\text{out}}(\psi) = \sum_{i=1}^n d_i$ and*

$$\mathcal{R}(\psi)(x) = (\mathcal{R}(\phi_1)(x), \dots, \mathcal{R}(\phi_n)(x)), \quad x \in \mathbb{R}^d. \tag{B.1}$$

Also, we list Lemma 5.1 in [12] and Lemma 5.3 in [6].

Lemma 9. *Let $L, n, N_0, N_L \in \mathbb{N}$, $\{a_\ell\}_{\ell=1}^n \subset \mathbb{R}$ and $\{\phi_\ell\}_{\ell=1}^n \subset \mathcal{N}$ be DNNs such that $\mathcal{L}(\phi_\ell) = L$, $\dim_{\text{in}}(\phi_\ell) = N_0$ and $\dim_{\text{out}}(\phi_\ell) = N_L$ for $\ell = 1, \dots, n$. Then, there exists $\psi \in \mathcal{N}$ such that $\mathcal{L}(\psi) = L$, $\mathcal{C}(\psi) \leq n^2 \mathcal{C}(\phi_1)$ and*

$$\mathcal{R}(\psi)(x) = \sum_{\ell=1}^n a_\ell \mathcal{R}(\phi_\ell)(x), \quad x \in \mathbb{R}^{N_0}. \tag{B.2}$$

Lemma 10. *Let $L_1, L_2, N_0^1, N_0^2, N_{L_1}^1, N_{L_2}^2 \in \mathbb{N}$ and $\phi_1, \phi_2 \in \mathcal{N}$ be DNNs such that $\mathcal{L}(\phi_1) = L_1$, $\mathcal{L}(\phi_2) = L_2$, $\dim_{\text{in}}(\phi_1) = N_0^1$, $\dim_{\text{out}}(\phi_1) = N_{L_1}^1$, $\dim_{\text{in}}(\phi_2) = N_0^2$, $\dim_{\text{out}}(\phi_2) = N_{L_2}^2$ and $N_{L_2}^2 = N_0^1$. Then, there exists $\psi \in \mathcal{N}$ such that $\mathcal{L}(\psi) = L_1 + L_2$, $\mathcal{C}(\psi) \leq 2(\mathcal{C}(\phi_1) + \mathcal{C}(\phi_2))$ and*

$$\mathcal{R}(\psi)(x) = \mathcal{R}(\phi_1)(\mathcal{R}(\phi_2)(x)), \quad x \in \mathbb{R}^{N_0^2}. \tag{B.3}$$

The following result of Theorem 6.3 of [6] is useful.

Lemma 11. *Let $M \in \mathbb{N} \cap [2, \infty)$ and $D \in [1, \infty)$. There exist DNNs $\{\psi_\varepsilon\}_{\varepsilon \in (0,1)} \subset \mathcal{N}$ and a constant $c > 0$ (independent of M and D) such that for all $\varepsilon \in (0,1)$, $\mathcal{C}(\psi_\varepsilon) \leq cM(|\log(\varepsilon)| + M \log(D) + \log(M))$ and*

$$\sup_{x_1, \dots, x_M \in [-D, D]} |\mathcal{R}(\psi_\varepsilon)(x_1, \dots, x_M) - \prod_{i=1}^M x_i| \leq \varepsilon. \tag{B.4}$$

In our analysis, the next result will be applied.

Lemma 12. *Let $a \in \mathbb{R}$, $b \in (a, \infty)$, $c > 0$, $m \in \mathbb{N} \cap [2, \infty)$, $d, L \in \mathbb{N}$ and $\{\phi_\ell\}_{\ell=1}^m \subset \mathcal{N}$ be DNNs such that for $i \in \{1, \dots, m\}$, $\mathcal{L}(\phi_i) = L$, $\dim_{\text{in}}(\phi_i) = d$, $\dim_{\text{out}}(\phi_i) = 1$, $\mathcal{C}(\phi_i) \leq c d^c$ and $\sup_{x \in [a,b]^d} |\mathcal{R}(\phi_i)(x)| \leq c d^c$. Then, there exist $\{\psi^{\varepsilon,d}\}_{\varepsilon \in (0,1), d \in \mathbb{N}} \subset \mathcal{N}$ and $\kappa > 0$ (independent of d) such that for all $\varepsilon \in (0,1)$ and $d \in \mathbb{N}$, we have $\mathcal{C}(\psi^{\varepsilon,d}) \leq \kappa \varepsilon^{-1} d^\kappa$ and*

$$\sup_{x \in [a,b]^d} \left| \mathcal{R}(\psi^{\varepsilon,d})(x) - \prod_{\ell=1}^m \mathcal{R}(\phi_\ell)(x) \right| \leq \varepsilon. \tag{B.5}$$

Proof of Lemma 12. First we use Lemma 11. Let $\varphi(d) := cd^c$. Then, there exist a set of DNNs $\{\Psi_{\varphi(d),\varepsilon}\}_{\varepsilon\in(0,1)} \subset \mathcal{N}$ and a constant $c' > 0$ (independent of m and $\varphi(d)$) such that for all $\varepsilon \in (0,1)$, $\mathcal{C}(\Psi_{\varphi(d),\varepsilon}) \le c' m^2 \varepsilon^{-1} d^c$ and

$$|\mathcal{R}(\Psi_{\varphi(d),\varepsilon})(\mathcal{R}(\phi_1)(x), \dots, \mathcal{R}(\phi_m)(x)) - \prod_{\ell=1}^{m} \mathcal{R}(\phi_\ell)(x)| \le \varepsilon, \tag{B.6}$$

for any $x \in [a,b]^d$. By Lemma 8, there exists $\Phi \in \mathcal{N}$ such that $\mathcal{C}(\Phi) \le mcd^c$ and

$$\mathcal{R}(\Phi)(x) = (\mathcal{R}(\phi_1)(x), \dots, \mathcal{R}(\phi_m)(x)), \quad x \in \mathbb{R}^d. \tag{B.7}$$

By Lemma 10, there exist $\{\psi^{\varepsilon,d}\}_{\varepsilon\in(0,1),d\in\mathbb{N}} \subset \mathcal{N}$ and $\kappa > 0$ such that for all $\varepsilon \in (0,1)$ and $d \in \mathbb{N}$, we have $\mathcal{C}(\psi^{\varepsilon,d}) \le \kappa \varepsilon^{-1} d^\kappa$,

$$\mathcal{R}(\psi^{\varepsilon,d})(x) = \mathcal{R}(\Psi_{\varphi(d),\varepsilon})(\mathcal{R}(\Phi)(x)), \quad x \in \mathbb{R}^d, \tag{B.8}$$

and

$$\sup_{x\in[a,b]^d} \left| \mathcal{R}(\psi^{\varepsilon,d})(x) - \prod_{\ell=1}^{m} \mathcal{R}(\phi_\ell)(x) \right| \le \varepsilon. \quad \square \tag{B.9}$$

C Sample code

We show the sample Python code used in the numerical computation in Section 6.2.

Listing 1: model.py

```python
import tensorflow as tf
from tensorflow. contrib. layers . python . layers import initializers
from tensorflow. python . training. moving_averages \
import assign_moving_average
from tensorflow. contrib. layers . python . layers import utils

import time

import numpy as np
import math
from scipy.stats import multivariate_normal as normal
from tensorflow.python.ops import control_flow_ops
from tensorflow import random_normal_initializer as norm_init
from tensorflow import random_uniform_initializer as unif_init
from tensorflow import constant_initializer as con


def neural_net(y, neurons, name, is_training,
               reuse=tf.AUTO_REUSE, decay =0.9, dtype=tf. float32):
    def batch_normalization(x):
        beta = tf. get_variable( 'beta', [x. get_shape()[ -1]], dtype,
            tf. zeros_initializer ())
        gamma = tf. get_variable(
               'gamma', [x. get_shape()[ -1]], dtype,
            tf. ones_initializer ())
        mv_mean = tf. get_variable(
            'mv_mean', [x. get_shape()[ -1]], dtype=dtype,
            initializer=tf. zeros_initializer (), trainable= False)
        mv_var = tf. get_variable(
            'mv_var', [x. get_shape()[ -1]], dtype =dtype,
        initializer=tf. ones_initializer(), trainable= False)
        mean, variance = tf.nn. moments(x, [0], name='moments')
        tf. add_to_collection(
            tf. GraphKeys. UPDATE_OPS,
            assign_moving_average(mv_mean, mean, decay,
```

```
                                          zero_debias= True ))
        tf. add_to_collection(
            tf. GraphKeys. UPDATE_OPS,
            assign_moving_average(mv_var, variance, decay,
            zero_debias= False ))
        mean, variance = utils. smart_cond( is_training,
                                            lambda : (mean, variance),
                                            lambda : (mv_mean, mv_var ))
        return tf.nn. batch_normalization(x, mean, variance,
                                            beta, gamma, 1e-6)
    def layer(x, out_size, activation):
        w = tf. get_variable(
                'weights', [x. get_shape(). as_list()[ -1], out_size],
                dtype, initializers. xavier_initializer ())
        return activation( batch_normalization(tf. matmul (x, w )))
    with tf. variable_scope(name, reuse = reuse ):
        y = batch_normalization(y)
        for i in range (len( neurons) - 1):
            with tf. variable_scope('layer_%i_' % (i + 1)):
                y = layer (y, neurons[i], tf.nn. relu)
        with tf. variable_scope('layer_%i_' % len( neurons)):
            return layer (y, neurons[ -1], tf. identity)

def nn_model(XT, Xini, weight, K, f, neurons, dtype=tf. float32):

    nn = neural_net(Xini, neurons, 'v', True, dtype= dtype )
    loss = (nn - tf. stop_gradient(f(K,XT)*weight) ) ** 2

    return tf. reduce_mean(loss)

def simulate(Simtype, T, n, d, X_min, X_max, X_valid, K, SDE, f, neurons,
    train_steps, batch_size, lr_boundaries, lr_values, epsilon=1e-8):

    tf. reset_default_graph ()

    Xini = tf.random_uniform((batch_size, d), minval=X_min, maxval=X_max)
    XT, weight = SDE(Xini, T, d, n, Simtype)

    loss = nn_model(XT, Xini, weight, K, f, neurons)

    global_step = tf. get_variable(
        'global_step', [], tf.int32,
        tf. zeros_initializer(), trainable= False )

    learning_rate = tf. train . piecewise_constant(
        global_step, lr_boundaries, lr_values)
    update_ops = tf. get_collection(
        tf. GraphKeys. UPDATE_OPS, 'v')
    with tf. control_dependencies( update_ops):
        train_op = tf. train . AdamOptimizer(
            learning_rate, epsilon= epsilon). minimize(
            loss, global_step= global_step)

    with tf. Session() as sess:

        sess. run(tf. global_variables_initializer ())
        var_list_n = tf. get_collection(
        tf. GraphKeys. GLOBAL_VARIABLES, 'v')

        for _ in range(train_steps):
            sess. run(train_op)

        v = sess.run(neural_net(tf.cast(X_valid, tf.float32), neurons, 'v',
            False))

        return np.reshape(v, [-1])
```

Listing 2: CEV.py

```
from model import simulate
import numpy as np
import time
import tensorflow as tf

def f(K, x):
    return tf.exp(-r*T)* tf.maximum(tf.reduce_max(x, 1, keepdims = True) -K, 0.0)

def SDE(Xini, T, d, n, Simtype):

    X = Xini
    Weight = 1.0

    if Simtype == 'Euler-Maruyama':
        for _n in range (n):
            dW = tf. random_normal(( batch_size , d), stddev =np. sqrt(T/n))
            X = X + r*X*T/n + sigma *K**(1.0-beta)*X**beta*dW
            X = tf.maximum(X, 0.0)

    elif Simtype == 'AE':
        dW = tf. random_normal(( batch_size, d), stddev =np. sqrt(T))
        Weight = M_weight(X, T, dW)
        X = X + r*X*T + sigma *K**(1.0-beta)*X**beta*dW
        X = tf.maximum(X, 0.0)

    return X, Weight

def M_weight(x, T, dW):

    inv = 1.0/(sigma *K**(1.0-beta)*x**beta)
    L0V0 = r**2*x
    L0Vi = r*beta*sigma*K**(1.0-beta)*x**beta + 1.0/2.0*beta*(beta-1.0)*sigma**3*K**(3.0*(1.0-beta))*x**(3.0*beta-2.0)
    LiV0 = r*sigma*K**(1.0-beta)*x**beta
    LiVi = beta*sigma**2*K**(2.0*(1.0-beta))*x**(2.0*beta-1.0)
    w11 = dW*dW-T
    w001 = dW*T**2.0
    w011 = w11*T
    w111 = dW**3-3.0*dW*T

    A = 1.0 / (2.0 * T) * tf.reduce_sum(inv * ( L0V0 * w001 + L0Vi * w011 + LiV0 * w011 + LiVi * w111 ) ,1 ,keepdims=True)

    return 1.0 + A

T, d, K = 1.0, 100, 100.0
r, sigma, beta = 0.01, 0.3, 0.5
X_min, X_max = 99.0, 101.0

grid = 10
X_valid = np.ones((1,d))*np.expand_dims(np.linspace(X_min, X_max, grid+1), axis=1)

batch_size = 1024
train_steps = 5000
neurons = [2*d, 2*d, 1]
lr_values = [0.5 , 0.05, 0.005]
lr_boundaries = [train_steps // 10 * 3 ,train_steps // 10 * 6]

for Simtype in ['Euler-Maruyama', 'AE']:
    if Simtype == 'Euler-Maruyama':
        n_range = [1,2,4,8,16,32,64,128]
    else:
        n_range = [1]
```

```
    for n in n_range:
        print ('batch size, train steps, lr_values 1, lr_values 2,
            lr_values 3, d, x, K, T, n, value, time, Simtype')
        t_0 = time. time ()
        vv = simulate(Simtype, T, n, d, X_min, X_max, X_valid, K, SDE, f,
            neurons, train_steps, batch_size,lr_boundaries, lr_values)
        t_1 = time. time ()

        for i in range(grid+1):
            print ('%i, %i, %.1f, %.2f, %.3f, %i, %.1f, %.1f, %.1f, %i, %.6f
                , %.2f, %s' %(batch_size, train_steps, lr_values[0],
                lr_values[1], lr_values[2], d, X_valid[i,0], K, T, n, vv[i],
                 t_1 - t_0, Simtype))
        print('')
```

Acknowledgements

This work is supported by JST PRESTO (Grant Number JPMJPR2029), Japan. The authors thank Riu Naito for his great help in numerical computation of the proposed method. We also thank two anonymous reviewers for valuable comments and suggestions.

References

[1] V. Bally, An elementary introduction to Malliavin calculus, INRIA (2003)

[2] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen, Solving the Kolmogorov PDE by Means of Deep Learning, *Journal of Scientific Computing*, 88, Article number: 73 (2021)

[3] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, arXiv (2020)

[4] J. Berner, P. Grohs, and A. Jentzen, Analysis of the Generalization Error: Empirical Risk Minimization over Deep Artificial Neural Networks Overcomes the Curse of Dimensionality in the Numerical Approximation of Black-Scholes Partial Differential Equations, *SIAM Journal on Mathematics of Data Science*, 2(3), 631-657 (2020)

[5] D. Elbrächter, D. Perekrestenko, P. Grohs and H. Bölcskei, Deep Neural Network Approximation Theory, arXiv (2021)

[6] D. Elbrächter, P. Grohs, A. Jentzen and C. Schwab, DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing, *Constructive Approximation* (2021)

[7] E. Fournié, J.M. Lasry, J. Lebuchoux, P.L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance, *Finance and Stochastics*, 3(4), 391-412 (1999)

[8] M. Fujii, A. Takahashi, and M. Takahashi, Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs, *Asia-Pacific Financial Markets*, (2019)

[9] M. Germain, H. Pham, and X. Warin, Approximation Error Analysis of Some Deep Backward Schemes for Nonlinear PDEs, *SIAM Journal on Scientific Computing*, 44(1) (2022)

[10] E. Gobet and R. Munos, Sensitivity analysis using Itô-Malliavin calculus and martingales, and application to stochastic optimal control, *SIAM Journal on Control and Optimization*, 43(5), 1676-1713 (2005)

[11] L. Gonon, P. Grohs, A Jentzen, D. Kofler, and D. Siska, Uniform error estimates for artificial neural network approximations for heat equations, *IMA Journal of Numerical Analysis* (2021)

[12] P. Grohs, F. Hornung, A. Jentzen, and P. Wurstemberger, A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations, *Mem. Amer. Math. Soc.* (2021)

[13] C. Huré, H. Pham, and X. Warin, Deep backward schemes for high-dimensional nonlinear PDEs, *Mathematics of Computation*, 89, 1547-1579 (2020)

[14] 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 Part. Differ. Equ. Appl.*, 1, 1-34 (2020)

[15] M. Hutzenthaler, A. Jentzen, T. Kruse, T.A. Nguyen, and P. von Wurstemberger, Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations, *Proc. Roy. Soc. A*, (2020)

[16] N. Ikeda and S. Watanabe, Stochastic Differential Equations and Diffusion Processes, 2nd ed., North-Holland, Amsterdam, Kodansha, Tokyo (1989)

[17] Y. Iguchi and T. Yamada, A second order discretization for degenerate systems of stochastic differential equations, *IMA Journal of Numerical Analysis* (to appear) (2020)

[18] Y. Iguchi and T. Yamada, Operator splitting around Euler-Maruyama scheme and high order discretization of heat kernels, *ESAIM: Mathematical Modelling and Numerical Analysis*, vol 55, 323-367 (2021)

[19] A. Jentzen, D. Salimova, and T. Welti, A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients, *Communications in Mathematical Sciences* (2021)

[20] D. Kingma and J. Ba, *Adam*: a method for stochastic optimization, *Proceedings of the International Conference on Learning Representations* (ICLR), May (2015)

[21] P.E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer (1992)

[22] S. Kusuoka and D. Stroock, Applications of the Malliavin calculus Part I, *Stochastic Analysis* (Katata/Kyoto 1982), 271-306 (1984)

[23] N. Kunitomo and A. Takahashi, The asymptotic expansion approach to the valuation of interest rate contingent claims, *Mathematical Finance*, 11, 117-151 (2001)

[24] N. Kunitomo and A. Takahashi, On validity of the asymptotic expansion approach in contingent claim analysis, *Annals of Applied Probability*, 13(3), 914-952 (2003)

[25] P. Malliavin, *Stochastic Analysis*, Springer (1997)

[26] P. Malliavin and A. Thalmaier, *Stochastic Calculus of Variations in Mathematical Finance*, Springer (2006)

[27] R. Naito and T. Yamada, A third-order weak approximation of multidimensional Itô stochastic differential equations, *Monte Carlo Methods and Applications*, vol 25 (2), 97-120 (2019)

[28] R. Naito and T. Yamada, A higher order weak approximation of McKean-Vlasov type SDEs, *BIT Numerical Mathematics*, 62, 521-559 (2021)

[29] D. Nualart, *The Malliavin Calculus and Related Topics*, Springer (2006)

[30] Y. Okano and T. Yamada, A control variate method for weak approximation of SDEs via discretization of numerical error of asymptotic expansion, *Monte Carlo Methods and Applications*, 25(3) (2019)

[31] C. Reisinger and Y. Zhang, Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems, *Analysis and Applications*, 18(6), 951-999 (2020)

[32] A. Takahashi, An asymptotic expansion approach to pricing financial contingent claims, *Asia-Pacific Financial Markets*, 6(2), 115-151 (1999)

[33] A. Takahashi, Asymptotic expansion approach in finance, *Large Deviations and Asymptotic Methods in Finance* (P. Friz, J. Gatheral, A. Gulisashvili, A. Jacquier and J. Teichmann ed.), Springer Proceedings in Mathematics & Statistics (2015)

[34] A. Takahashi, Y. Tsuchida, and T. Yamada, A new efficient approximation scheme for solving high-dimensional semilinear PDEs: control variate method for Deep BSDE solver, *Journal of Computational Physics*, Vol.454, 110956 (2022)

[35] A. Takahashi and T. Yamada, An asymptotic expansion with push-down of Malliavin weights, *SIAM Journal on Financial Mathematics*, 3, 95-136 (2012)

[36] A. Takahashi and T. Yamada, A remark on approximation of the solutions to partial differential equations in finance, *Recent Advances in Financial Engineering 2011*, 133-181 (2012)

[37] A. Takahashi and T. Yamada, On error estimates for asymptotic expansions with Malliavin weights: Application to stochastic volatility model, *Mathematics of Operations Research*, 40(3), 513-551 (2015)

[38] A. Takahashi and T. Yamada, A weak approximation with asymptotic expansion and multidimensional Malliavin weights, *Annals of Applied Probability*, 26(2), 818-856 (2016)

[39] A. Takahashi and N. Yoshida, Monte Carlo simulation with asymptotic method, *Journal of the Japan Statistical Society*, 35(2), 171-203 (2005)

[40] S. Watanabe, Analysis of Wiener functionals (Malliavin calculus) and its applications to heat kernels, *Annals of Probability*, 15, 1-39 (1987)

[41] T. Yamada, An arbitrary high order weak approximation of SDE and Malliavin Monte Carlo: application to probability distribution functions, *SIAM Journal on Numerical Analysis*, 57(2), 563-591 (2019)

[42] T. Yamada and K.Yamamoto, Second order discretization of Bismut-Elworthy-Li formula: application to sensitivity analysis, *SIAM/ASA Journal on Uncertainty Quantification*, 7(1), 143-173 (2019)

[43] N. Yoshida, Asymptotic expansions of maximum likelihood estimators for small diffusions via the theory of Malliavin-Watanabe, *Probability Theory and Related Fields*, 92, 275-311 (1992)