Approximation error analysis of some deep backward schemes for nonlinear PDEs

Maximilien Germain, Huyen Pham, Xavier Warin

Deep backward multistep schemes for nonlinear PDEs and approximation error analysis *

Maximilien GERMAIN † Huyên PHAM ‡ Xavier WARIN §

September 13, 2021
to appear in SIAM Journal on Scientific Computing

Abstract

Recently proposed numerical algorithms for solving high-dimensional nonlinear partial differential equations (PDEs) based on neural networks have shown their remarkable performance. We review some of them and study their convergence properties. The methods rely on probabilistic representation of PDEs by backward stochastic differential equations (BSDEs) and their iterated time discretization. Our proposed algorithm, called deep backward multistep scheme (MDBDP), is a machine learning version of the LSMDP scheme of Gobet, Turkedjiev (Math. Comp. 85, 2016). It estimates simultaneously by backward induction the solution and its gradient by neural networks through sequential minimizations of suitable quadratic loss functions that are performed by stochastic gradient descent. Our main theoretical contribution is to provide an approximation error analysis of the MDBDP scheme as well as the deep splitting (DS) scheme for semilinear PDEs designed in Beck, Becker, Cheridito, Jentzen, Neufeld (2019). We also supplement the error analysis of the DBDP scheme of Huré, Pham, Warin (Math. Comp. 89, 2020). This yields notably convergence rate in terms of the number of neurons for a class of deep Lipschitz continuous GroupSort neural networks when the PDE is linear in the gradient of the solution for the MDBDP scheme, and in the semilinear case for the DBDP scheme. We illustrate our results with some numerical tests that are compared with some other machine learning algorithms in the literature.

1 Introduction

Let us consider the nonlinear parabolic partial differential equation (PDE) of the form

$$\begin{cases} \partial_t u + \mu \cdot D_x u + \frac{1}{2}\,\mathrm{Tr}(\sigma\sigma^\intercal D_x^2 u) \;=\; f(\cdot,\cdot,u,\sigma^\intercal D_x u) & \text{on } [0,T)\times\mathbb{R}^d \\ u(T,\cdot) \;=\; g & \text{on } \mathbb{R}^d, \end{cases} \tag{1.1}$$

with μ, σ functions defined on $[0,T]\times\mathbb{R}^d$, valued respectively in $\mathbb{R}^d$, and $\mathbb{M}^d$ (the set of $d\times d$ matrices), a nonlinear generator function f defined on $[0,T]\times\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d$, and a terminal function g defined on $\mathbb{R}^d$. Here, the operators D_x, D_x^2 refer respectively to the first and second order spatial derivatives, the symbol . denotes the scalar product, and $^\intercal$ is the transpose of vector or matrix.

A major challenge in the numerical resolution of such semilinear PDEs is the so-called "curse of dimensionality" making unfeasible the standard discretization of the state space in dimension greater than 3. Probabilistic mesh-free methods based on the Backward Stochastic Differential Equation (BSDE) representation of semilinear PDEs through the nonlinear Feynman-Kac formula were developed in [Zha04], [BT04], [HL+19], and (ii) on multilevel Picard methods, developed in [E+18] with algorithms based on Picard iterations, multi-level techniques and automatic differentiation. These methods permit to handle some PDEs with non linearity in u and its gradient $D_x u$, with convergence results as well as numerous numerical examples showing their efficiency in high dimension.

*This work is supported by FiME, Laboratoire de Finance des Marchés de l'Energie, and the "Finance and Sustainable Development" EDF - CACIB Chair.

†EDF R&D, LPSM, Université de Paris mgermain at lpsm.paris

‡LPSM, Université de Paris, FiME, CREST ENSAE pham at lpsm.paris

§EDF R&D, FiME xavier.warin at edf.fr

Over the last few years, machine learning methods have emerged since the pioneering papers by [HJE17] and [SS17], and have shown their efficiency for solving high-dimensional nonlinear PDEs by means of neural networks approximation. The work [HJE17] introduces a global machine learning resolution technique via a BSDE approach. The solution is represented by one feedforward neural network by time step, whose parameters are chosen as solutions of a single global optimization problem. It allows to solve PDEs in high dimension and a convergence study of Deep BSDE is conducted in [HL20]. The Deep Galerkin method of [SS17] proposes another global meshfree method with a random sampling of time and space points inside a bounded domain. A different point of view is proposed by [HPW20] with convergence results in L^2 for solving semilinear PDEs, where the solution and its gradient are estimated simultaneously by backward induction through the minimization of sequential loss functions. Similar idea also appears in [SS17] for linear PDEs. At the cost of solving multiple optimization problems, the Deep Backward scheme (DBDP) of [HPW20] verifies better stability and accuracy properties than the global method in [HJE17], as illustrated on several test cases. The recent paper [Bec+19] also introduces machine learning schemes based on local loss functions, called Deep Splitting (DS) method which estimates the PDE solution through backward explicit local optimization problems relying on a neural network regression method for the computation of conditional expectations.

In this paper, we propose machine learning schemes that use multistep methods introduced in [BD07] and [GT16]. The idea is to rely on the whole previously computed values of the discretized processes in the backward computations of the approximation as it is expected to yield a better propagation of regression errors. We shall develop this approach to the DBDP scheme of [HPW20], leading to the so-called deep backward multi-step scheme (MDBDP). This can be viewed as a machine learning version of the Multi-step Forward Dynamic Programming method studied by [GT16]. However, instead of solving at each time step two regression problems, our approach allows to consider only a single minimization as in the DBDP scheme. Compared to the latter, the multi-step consideration is expected to provide better accuracy by reducing the propagation of errors in the backward induction.Our main theoretical contribution is a detailed study of the approximation error of MDBDP scheme, through standard stability-type arguments for BSDEs (see e.g. Section 4.4 in [Zha17] for the continuous time case). The arguments can be adapted to obtain the convergence of the DS scheme introduced in [Bec+19]. Furthermore, by relying on recent approximation results for deep neural networks in [TSB21], we obtain a rate of convergence of our scheme in terms of the number of neurons, and supplement the convergence analysis of the DBDP scheme [HPW20].

We provide some numerical tests of our proposed algorithms, which show the benefit of multi-step schemes, and compare our results with the cited machine learning schemes. Notice that the GroupSort network is used for theoretical analysis but in the numerical implementation, we applied standard networks with tanh as activation function. The theoretical analysis of the convergence of methods relying on standard neural networks is left to future research. More numerical examples and tests are presented in the extended first arXiv version [GPW20] of this paper.

The plan of the paper is the following. In Section 2, we give a brief reminder on neural networks and notably on a specific class of deep network functions considered in [ALG19; TSB21] that yields an approximation result with rate of convergence for Lipschitz functions. We also review machine learning schemes for the numerical resolution of semilinear PDEs. We then describe in detail the MDBDP scheme. We state in Section 3 the convergence of the MDBDP, DS, and DBDP schemes, while Section 4 is devoted to the proof of these results. Section 5 gives some numerical tests for illustration.

2 BSDE Machine Learning Schemes for Semilinear PDEs

In this section, we review recent numerical schemes, and present our new scheme for the resolution of the semi-linear PDE (1.1) by approximations in the class of neural networks and relying on probabilistic representation of the solution to the PDE.

2.1 Neural Networks

We denote by

$$\mathcal{L}^{\rho}_{d_1,d_2} = \Big\{\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2} : \exists\, (\mathcal{W},\beta) \in \mathbb{R}^{d_2\times d_1}\times\mathbb{R}^{d_2},\ \phi(x) \ = \ \rho(\mathcal{W}x+\beta)\Big\},$$

the set of layer functions with input dimension d_1, output dimension d_2, and activation function $\rho : \mathbb{R}^{d_2} \to \mathbb{R}^{d_2}$. Usually, the activation is applied component-wise via a one-dimensional activation function, i.e., $\rho(x_1,\dots,x_{d_2}) = \big(\hat\rho(x_1),\dots,\hat\rho(x_{d_2})\big)$ with $\hat\rho : \mathbb{R} \mapsto \mathbb{R}$, to the affine map $x \in \mathbb{R}^{d_1} \mapsto \mathcal{W}x+\beta \in \mathbb{R}^{d_2}$, with a matrix $\mathcal{W}$ called weight, and vector β called bias. Standard examples of activation functions $\hat\rho$ are the sigmoid, the ReLU, the tanh. When ρ is the identity function, we simply write $\mathcal{L}_{d_1,d_2}$.

We then define

$$\mathcal{N}^{\rho}_{d_0,d',\ell,m} = \Big\{\varphi : \mathbb{R}^{d_0} \to \mathbb{R}^{d'} : \exists\phi_0 \in \mathcal{L}^{\rho_0}_{d_0,m_0},\ \exists\phi_i \in \mathcal{L}^{\rho_i}_{m_{i-1},m_i}, i=1,\dots,\ell-1,$$
$$\exists\phi_\ell \in \mathcal{L}_{m_{l-1},d'}, \varphi \ = \ \phi_\ell\circ\phi_{\ell-1}\circ\cdots\circ\phi_0\Big\},$$

as the set of feedforward neural networks with input layer dimension d_0, output layer dimension d', and ℓ hidden layers with m_i neurons per layer ($i = 0,\cdots,\ell-1$). These numbers d_0, d', ℓ, the sequence $m = (m_i)_{i=0,\dots,\ell-1}$, and sequence of activation functions $\rho = (\rho_i)_{i=0,\dots,\ell-1}$, form the architecture of the network. In the sequel, we shall mostly work with the case $d_0 = d$ (dimension of the state variable x).

A given network function $\varphi \in \mathcal{N}^{\rho}_{d_0,d',\ell,m}$ is determined by the weight/bias parameters $\theta = (\mathcal{W}_0,\beta_0,\dots,\mathcal{W}_\ell,\beta_\ell)$ defining the layer functions $\phi_0\dots,\phi_\ell$, and we shall sometimes write $\varphi = \varphi_\theta$.

We recall the fundamental result of [HSW89] that justifies the use of neural networks as function approximators, in the usual case of activation functions applied componentwise at each hidden layer.

Universal approximation theorem. The space $\bigcup_{i=0}^{\ell-1}\bigcup_{m_i=0}^{\infty}\mathcal{N}^{\rho}_{d_0,d',\ell,m}$ is dense in $L^2(\nu)$, the set of measurable functions $h : \mathbb{R}^{d_0} \to \mathbb{R}^{d'}$ s.t. $\int |h(x)|_2^2\nu(\mathrm{d}x) < \infty$, for any finite measure ν on $\mathbb{R}^{d_0}$, whenever ρ is continuous and non-constant.

This universal approximation theorem does not provide any rate of convergence, nor reveals even in theory how to achieve a given accuracy for a fixed number of neurons. There are few results in the literature that prove precise rates of convergence for approximation with deep neural networks, and most of them focus on single hidden layer. We mention the recent approximation theorem for (locally) Lipschitz continuous functions, which results from Proposition 6 combined with Proposition 1 (for $\alpha = 1$) or Section 2.5 (for $\alpha = 2$), in [Bac17], and motivates the introduction of the set of network functions $\mathcal{N}^{\alpha,R,\gamma}_{d,m,d'}$.

This universal approximation theorem does not provide any rate of convergence, nor reveals even in theory how to achieve a given accuracy for a fixed number of neurons. Some results give rates for the approximation of functions in Sobolev spaces [Pin99], for bounded convex subdifferentiable Lipschitz functions [BGS15] or bounded Lipschitz functions [Yar17], but here, we need a result related to (possibly unbounded) Lipschitz functions. The paper [Bac17] provides a possible answer in this direction, but we instead rely on a simpler approach in [TSB21], building on the GroupSort deep neural networks introduced by [ALG19]. Let $\kappa \in \mathbb{N}^*,\ \kappa \geq 2$, be a grouping size, dividing the number of neurons $m_i = \kappa n_i$, at each layer $i = 0,\cdots\ell-1$. $\sum_{i=0}^{\ell-1} m_i$ will be refered to as the width of the network and $\ell+1$ as its depth. The GroupSort networks correspond to classical deep feedforward neural networks in $\mathcal{N}^{\zeta_\kappa}_{d,1,\ell,m}$ with a specific sequence of activation function $\zeta_\kappa = (\zeta^i_\kappa)_{i=0,\dots,\ell-1}$, and one-dimensional output. Each nonlinear function ζ^i_κ divides its input into groups of size κ and sorts each group in decreasing order, see Figure 1. Moreover, by enforcing the parameters of the GroupSort to satisfy with the Euclidian norm $|\cdot|_2$ and the ℓ_∞ norm $|\cdot|_\infty$:

$$\sup_{|x|_2=1}|\mathcal{W}_0x|_\infty \leq 1,\ \ \sup_{|x|_\infty=1}|\mathcal{W}_ix|_\infty \leq 1,\ |\beta_j|_\infty \leq M,\ i=1,\cdots,l,\ j=0,\cdots,l$$

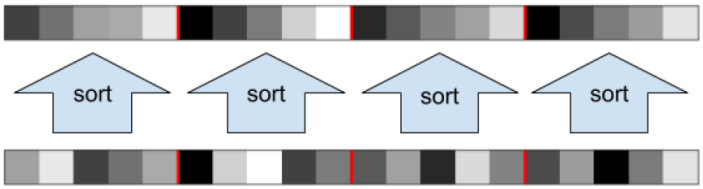

Figure 1: GroupSort activation function ζ_κ with grouping size $\kappa = 5$ and $m = 20$ neurons, figure from [ALG19].

for some $M > 0$, the related GroupSort neural networks from $\mathcal{N}^{\zeta_\kappa}_{d,d',\ell,m}$ are 1-Lipschitz. The space of such 1-Lipschitz GroupSort neural networks is called $\mathcal{S}^{\zeta_\kappa}_{d,\ell,m}$:

$$\mathcal{S}^{\zeta_\kappa}_{d,\ell,m} = \{\varphi_{(\mathcal{W}_0,\beta_0,\dots,\mathcal{W}_\ell,\beta_\ell)} \in \mathcal{N}^{\zeta_\kappa}_{d,1,\ell,m},\ \sup_{|x|_2=1} |\mathcal{W}_0 x|_\infty \leq 1,\ \sup_{|x|_\infty=1} |\mathcal{W}_i x|_\infty \leq 1,$$
$$|\beta_j|_\infty \leq M,\ i = 1,\cdots,l,\ j = 0,\cdots,l\}.$$

We then introduce the set $\mathcal{G}^{\zeta_\kappa}_{K,d,d',\ell,m}$ as

$$\mathcal{G}^{\zeta_\kappa}_{K,d,d',\ell,m} := \{\Psi = (\Psi_i)_{i=1,\dots,d'} : \mathbb{R}^d \mapsto \mathbb{R}^{d'},\ \Psi_i : x \in \mathbb{R}^d \mapsto K\beta_i \quad \phi_i\Big(\frac{x+\alpha_i}{\beta_i}\Big) \in \mathbb{R},$$
$$\phi_i \in \mathcal{S}^{\zeta_\kappa}_{d,\ell,m},\quad \text{for some } \alpha_i \in \mathbb{R}^d,\ \beta_i > 0\}.$$

Notice that these networks are $\sqrt{d'}K$-Lipschitz and that each of their components is K-Lipschitz. We rely on the the following quantitative approximation result which directly follows from [TSB21].

Proposition 2.1 (Slight extension of Tanielian, Sangnier, Biau [TSB21] : Approximation theorem for Lipschitz functions by Lipschitz GroupSort neural networks.)**.** *Let $f : [-R,R]^d \mapsto \mathbb{R}^{d'}$ be K-Lipschitz. Then, for all $\varepsilon > 0$, there exists a GroupSort neural network g in $\mathcal{G}^{\zeta_\kappa}_{K,d,d',\ell,m}$ verifying*

$$\sup_{x\in[-R,R]^d} |f(x) - g(x)|_2 \leq \sqrt{d'} 2RK\varepsilon$$

with g of grouping size $\kappa = \lceil \frac{2\sqrt{d}}{\varepsilon} \rceil$, depth $\ell + 1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((\frac{2\sqrt{d}}{\varepsilon})^{d^2-1})$ in the case $d > 1$. If $d = 1$, the same result holds with g of grouping size $\kappa = \lceil \frac{1}{\varepsilon} \rceil$, depth $\ell + 1 = 3$ and width $\sum_{i=0}^{\ell-1} m_i = O(\frac{1}{\varepsilon})$.

Proof. With f_i the i-th component of f, define

$$\widetilde{f_i} : z \in [0,1]^d \mapsto \frac{f_i(2R(z - 1/2))}{2RK}. \tag{2.1}$$

Then $\widetilde{f_i}$ is 1-Lipschitz and by Theorem 3 from [TSB21] if $d > 1$ (or Proposition 5 from [TSB21] if $d = 1$), there exists a 1-Lipschitz GroupSort neural network $g_i \in \mathcal{S}^{\zeta_\kappa}_{d,\ell,m}$ verifying

$$\sup_{z\in[0,1]^d} |\widetilde{f_i}(z) - g_i(z)| \leq \varepsilon$$

with g_i of grouping size $\kappa = O(\frac{2\sqrt{d}}{\varepsilon})$, depth $\ell+1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((\frac{2\sqrt{d}}{\varepsilon})^{d^2-1})$(respectively grouping size $\kappa = O(\frac{1}{\varepsilon})$, depth $\ell + 1 = 3$ and width $\sum_{i=0}^{\ell-1} m_i = O(\frac{1}{\varepsilon})$ if $d = 1$). Inverting (2.1) we have $f_i(x) = 2KR\widetilde{f_i}(\frac{x+R}{2R})$ hence

$$\sup_{x\in[-R,R]^d} \Big| f_i(x) - 2KRg_i\Big(\frac{x+R}{2R}\Big)\Big| \leq 2KR\varepsilon.$$

The result is proven by concatenating the d' K-Lipschitz GroupSort networks $x \mapsto 2KRg_i(\frac{x+R}{2R})$, $i = 1,\cdots,d'$. □

Remark 2.1. *As mentioned in [TSB21], GroupSort neural networks generalize the ReLU networks and, thanks to their Lipschitz continuity, offer better stability regarding noisy inputs and adversarial attacks. It also appears that GroupSort networks are more expressive than ReLU ones.*

2.2 Existing Schemes

We review recent machine learning schemes that will serve as benchmarks for our new scheme described in the next section. All these schemes rely on BSDE representation of the solution to the PDE, and differ according to the formulation of the time discretization of the BSDE.

For this purpose, let us introduce the diffusion process $\mathcal{X}$ in $\mathbb{R}^d$ associated to the linear part of the differential operator in the PDE (1.1), namely:

$$\mathcal{X}_t = \mathcal{X}_0 + \int_0^t \mu(s, \mathcal{X}_s)\mathrm{d}s + \int_0^t \sigma(s, \mathcal{X}_s)\mathrm{d}W_s, \quad 0 \le t \le T, \tag{2.2}$$

where W is a d-dimensional standard Brownian motion on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $\mathbb{F} = (\mathcal{F}_t)_t$, and $\mathcal{X}_0$ is an $\mathcal{F}_0$-measurable random variable valued in $\mathbb{R}^d$. Recall from [PP90] that the solution u to the PDE (1.1) admits a probabilistic representation in terms of the BSDE:

$$Y_t = g(\mathcal{X}_T) - \int_t^T f(s, \mathcal{X}_s, Y_s, Z_s)\mathrm{d}s - \int_t^T Z_s.\mathrm{d}W_s, \quad 0 \le t \le T, \tag{2.3}$$

via the Feynman-Kac formula $Y_t = u(t, \mathcal{X}_t)$, $0 \le t \le T$. When u is a smooth function, this BSDE representation is directly obtained by Itô's formula applied to $u(t, \mathcal{X}_t)$, and we have $Z_t = \sigma(t, \mathcal{X}_t)^\intercal D_x u(t, \mathcal{X}_t)$, $0 \le t \le T$.

Let π be a subdivision $\{t_0 = 0 < t_1 < \cdots < t_N = T\}$ with modulus $|\pi| := \sup_i \Delta t_i$, $\Delta t_i := t_{i+1} - t_i$, satisfying $|\pi| = O\left(\frac{1}{N}\right)$, and consider the Euler scheme

$$X_i = \mathcal{X}_0 + \sum_{j=0}^{i-1} \mu(t_j, X_j)\Delta t_j + \sum_{j=0}^{i-1} \sigma(t_j, X_j)\Delta W_j, \quad i = 0, \dots, N,$$

where $\Delta W_j := W_{t_{j+1}} - W_{t_j}$, $j = 0, \dots, N$. When the diffusion $\mathcal{X}$ cannot be simulated, we shall rely on the simulated paths of $(X_i)_i$ that act as training data in the setting of machine learning, and thus our training set can be chosen as large as desired.

The time discretization of the BSDE (2.3) is written in backward induction as

$$Y_i^\pi = Y_{i+1}^\pi - f(t_i, X_i, Y_i^\pi, Z_i^\pi)\Delta t_i - Z_i^\pi.\Delta W_i, \quad i = 0, \dots, N-1, \tag{2.4}$$

which also reads as conditional expectation formulae

$$\left\{ \begin{array}{rcl} Y_i^\pi & = & \mathbb{E}_i\Big[Y_{i+1}^\pi - f(t_i, X_i, Y_i^\pi, Z_i^\pi)\Delta t_i\Big] \\ Z_i^\pi & = & \mathbb{E}_i\Big[\frac{\Delta W_i}{\Delta t_i} Y_{i+1}^\pi\Big], \qquad i = 0, \dots, N-1, \end{array} \right. \tag{2.5}$$

where $\mathbb{E}_i$ denotes the conditional expectation w.r.t. $\mathcal{F}_{t_i}$. Alternatively, by iterating relations (2.4) together with the terminal relation $Y_N^\pi = g(X_N)$, we have

$$Y_i^\pi = g(X_N) - \sum_{j=i}^{N-1} \big[f(t_j, X_j, Y_j^\pi, Z_j^\pi)\Delta t_j + Z_j^\pi.\Delta W_j\big], \quad i = 0, \dots, N-1. \tag{2.6}$$

• Deep BSDE scheme [HJE17].

The idea of the method is to treat the backward equation (2.4) as a forward equation by approximating the initial condition Y_0 and the Z component at each time by networks functions of the X process, so as to match the terminal condition. More precisely, the problem is to minimize over network functions $\mathcal{U}_0 : \mathbb{R}^d \to \mathbb{R}$, and sequences of network functions $\boldsymbol{\mathcal{Z}} = (\mathcal{Z}_i)_i$, $\mathcal{Z}_i : \mathbb{R}^d \to \mathbb{R}^d$, $i = 0, \dots, N-1$, the global quadratic loss function

$$J_G(\mathcal{U}_0, \boldsymbol{\mathcal{Z}}) = \mathbb{E}\Big|Y_N^{\mathcal{U}_0, \boldsymbol{\mathcal{Z}}} - g(X_N)\Big|^2,$$

where $(Y_i^{\mathcal{U}_0,\boldsymbol{\mathcal{Z}}})_i$ is defined by forward induction as

$$Y_{i+1}^{\mathcal{U}_0,\boldsymbol{\mathcal{Z}}} = Y_i^{\mathcal{U}_0,\boldsymbol{\mathcal{Z}}} + f(t_i, X_i, Y_i^{\mathcal{U}_0,\boldsymbol{\mathcal{Z}}}, \mathcal{Z}_i(X_i))\Delta t_i + \mathcal{Z}_i(X_i).\Delta W_i, \quad i = 0, \dots, N-1,$$

starting from $Y_0^{\mathcal{U}_0,\boldsymbol{\mathcal{Z}}} = \mathcal{U}_0(\mathcal{X}_0)$. The output of this scheme, for the solution $(\widehat{\mathcal{U}}_0, \widehat{\boldsymbol{\mathcal{Z}}})$ to this global minimization problem, provides an approximation $\widehat{\mathcal{U}}_0$ of the solution $u(0,.)$ to the PDE at time 0, and approximations $Y_i^{\widehat{\mathcal{U}}_0,\widehat{\boldsymbol{\mathcal{Z}}}}$ of the solution to the PDE (1.1) at times t_i evaluated at $\mathcal{X}_{t_i}$, i.e., of $Y_{t_i} = u(t_i, \mathcal{X}_{t_i})$, $i = 0, \dots, N$.

• Deep Backward Dynamic Programming (DBDP) [HPW20].

The method relies on the backward dynamic programming relation (2.4) arising from the time discretization of the BSDE, and learns simultaneously at each time step t_i the pair (Y_{t_i}, Z_{t_i}) with neural networks trained with the forward process X and the Brownian motion W. The scheme has two versions:

1. *DBDP1.* Starting from $\widehat{\mathcal{U}}_N^{(1)} = g$, proceed by backward induction for $i = N-1, \dots, 0$, by minimizing over network functions $\mathcal{U}_i : \mathbb{R}^d \to \mathbb{R}$, and $\mathcal{Z}_i : \mathbb{R}^d \to \mathbb{R}^d$ the local quadratic loss function

$$J_i^{(B1)}(\mathcal{U}_i, \mathcal{Z}_i) = \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(1)}(X_{i+1}) - \mathcal{U}_i(X_i) - f(t_i, X_i, \mathcal{U}_i(X_i), \mathcal{Z}_i(X_i))\Delta t_i - \mathcal{Z}_i(X_i).\Delta W_i\Big|^2,$$

   and update $(\widehat{\mathcal{U}}_i^{(1)}, \widehat{\mathcal{Z}}_i^{(1)})$ as the solution to this local minimization problem.

2. *DBDP2.* Starting from $\widehat{\mathcal{U}}_N^{(2)} = g$, proceed by backward induction for $i = N-1, \dots, 0$, by minimizing over C^1 network functions $\mathcal{U}_i : \mathbb{R}^d \to \mathbb{R}$ the local quadratic loss function

$$\begin{aligned} J_i^{(B2)}(\mathcal{U}_i) &= \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \mathcal{U}_i(X_i) - f(t_i, X_i, \mathcal{U}_i(X_i), \sigma(t_i, X_i)^\intercal D_x\mathcal{U}_i(X_i))\Delta t_i \\ &\quad - D_x\mathcal{U}_i(X_i)^\intercal \sigma(t_i, X_i)\Delta W_i\Big|^2, \end{aligned}$$

   where $D_x\mathcal{U}_i$ is the automatic differentiation of the network function $\mathcal{U}_i$. Update $\widehat{\mathcal{U}}_i^{(2)}$ as the solution to this problem, and set $\widehat{\mathcal{Z}}_i^{(2)} = \sigma^\intercal(t_i,.)D_x\mathcal{U}_i^{(2)}$.

The output of DBDP provides an approximation $(\widehat{\mathcal{U}}_i, \widehat{\mathcal{Z}}_i)$ of the solution $u(t_i,.)$ and its gradient $\sigma^\intercal(t_i,.)D_xu(t_i,.)$ to the PDE (1.1) at times t_i, $i = 0, \dots, N-1$. The approximation error has been analyzed in [HPW20].

Remark 2.2. A machine learning scheme in the spirit of regression-based Monte-Carlo methods ([BT04], [GLW05]) for approximating condition expectations in the time discretization (2.5) of the BSDE, can be formulated as follows: starting from $\widehat{\mathcal{U}}_N = g$, proceed by backward induction for $i = N-1, \dots, 0$, in two regression problems:

(a) Minimize over network functions $\mathcal{Z}_i : \mathbb{R}^d \to \mathbb{R}^d$

$$J_i^{r,Z}(\mathcal{Z}_i) = \mathbb{E}\Big|\frac{\Delta W_i}{\Delta t_i}\widehat{\mathcal{U}}_{i+1}(X_{i+1}) - \mathcal{Z}_i(X_i)\Big|^2$$

and update $\widehat{\mathcal{Z}}_i$ as the solution to this minimization problem

(b) Minimize over network functions $\mathcal{U}_i : \mathbb{R}^d \to \mathbb{R}$

$$J_i^{r,Y}(\mathcal{U}_i) = \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}(X_{i+1}) - \mathcal{U}_i(X_i) - f(t_i, X_i, \mathcal{U}_i(X_i), \widehat{\mathcal{Z}}_i(X_i))\Delta t_i\Big|^2$$

and update $\widehat{\mathcal{U}}_i$ as the solution to this minimization problem.

Compared to these regression-based schemes, the DBDP scheme approximates simultaneously the pair component (Y, Z) via the minimization of the loss functions $J_i^{(B1)}(\mathcal{U}_i, \mathcal{Z}_i)$ (or $J_i^{(B2)}(\mathcal{U}_i)$ for the second version), $i = N-1, \dots, 0$. One advantage of this latter approach is that the accuracy of the DBDP scheme can be tested when computing at each time step the infimum of loss function, which should be equal to zero for the exact solution (up to the time discretization). In contrast, the infimum of the loss functions in the regression-based schemes is not known for the exact solution as it corresponds in theory to the residual of L^2-projection, and thus the accuracy of the scheme cannot be tested directly in-sample. Moreover, a variant where the automatic differentiation $D_x\mathcal{U}_i(X_i)$ is performed to estimate Z_{t_i} instead of using a second neural network $\widehat{\mathcal{Z}}_i$ (similarly as in the previous DBDP2 scheme) can also be considered. In this case, one only needs to solve for each time step the (b) optimization problem and not the (a) problem anymore. □

• **Deep Splitting (DS) scheme [Bec+19].**

This method also proceeds by backward induction as follows:

- Minimize over C^1 network functions $\mathcal{U}_N : \mathbb{R}^d \to \mathbb{R}$ the terminal loss function

$$J_N^S(\mathcal{U}_N) = \mathbb{E}\Big|g(X_N) - \mathcal{U}_N(X_N)\Big|^2,$$

  and denote by $\widehat{\mathcal{U}}_N$ as the solution to this minimization problem. If g is C^1, we can choose directly $\widehat{\mathcal{U}}_N = g$.

- For $i = N-1, \dots, 0$, minimize over C^1 network functions $\mathcal{U}_i : \mathbb{R}^d \to \mathbb{R}$ the loss function

$$\begin{aligned} & J_i^S(\mathcal{U}_i) \\ &= \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}(X_{i+1}) - \mathcal{U}_i(X_i) \\ &\qquad - f(t_i, X_{i+1}, \widehat{\mathcal{U}}_{i+1}(X_{i+1}), \sigma(t_i, X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}(X_{i+1}))\Delta t_i\Big|^2, \end{aligned} \tag{2.7}$$

  and update $\widehat{\mathcal{U}}_i$ as the solution to this minimization problem. Here D_x refers again to the automatic differentiation operator for network functions.

The DS scheme combines ideas of the DBDP2 and regression-based schemes where the current regression-approximation on Z is replaced by the automatic differentiation of the network function computed at the previous step. The current approximation of Y is then computed by a regression network-based scheme. In Section 3, we shall analyze the approximation error of the DS scheme. Please note that in (2.7) we consider a slight modification of the original DS scheme from [Bec+19]. In their loss function, the term $f(t_i, X_{i+1}, \widehat{\mathcal{U}}_{i+1}(X_{i+1}), \sigma(t_i, X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}(X_{i+1}))$ is replaced by $f(t_{i+1}, X_{i+1}, \widehat{\mathcal{U}}_{i+1}(X_{i+1}), \sigma(t_{i+1}, X_{i+1})^\intercal D_x\widehat{\mathcal{U}}_{i+1}(X_{i+1}))$.

2.3 Deep Backward Multi-step Scheme (MDBDP)

The starting point of the MDBDP scheme is the iterated representation (2.6) for the time discretization of the BSDE. This backward scheme is described as follows: for $i = N-1, \dots, 0$, minimize over network functions $\mathcal{U}_i : \mathbb{R}^d \to \mathbb{R}$, and $\mathcal{Z}_i : \mathbb{R}^d \to \mathbb{R}^d$ the loss function

$$\begin{aligned} J_i^{MB}(\mathcal{U}_i, \mathcal{Z}_i) = \mathbb{E}\Big|g(X_N) - \sum_{j=i+1}^{N-1} f(t_j, X_j, \widehat{\mathcal{U}}_j(X_j), \widehat{\mathcal{Z}}_j(X_j))\Delta t_j - \sum_{j=i+1}^{N-1} \widehat{\mathcal{Z}}_j(X_j).\Delta W_j \\ - \mathcal{U}_i(X_i) - f(t_i, X_i, \mathcal{U}_i(X_i), \mathcal{Z}_i(X_i))\Delta t_i - \mathcal{Z}_i(X_i).\Delta W_i\Big|^2 \end{aligned} \tag{2.8}$$

and update $(\widehat{\mathcal{U}}_i, \widehat{\mathcal{Z}}_i)$ as the solution to this minimization problem. This output provides an approximation $(\widehat{\mathcal{U}}_i, \widehat{\mathcal{Z}}_i)$ of the solution $u(t_i, .)$ to the PDE (1.1) at times t_i, $i = 0, \dots, N-1$. This approximation error will be analyzed in Section 3.

MDBDP is a machine learning version of the Multi-step Dynamic Programming method studied by [BD07] and [GT16]. Instead of solving at each time step two regression problems, our approach

allows to consider only a single minimization as in the DBDP scheme. Compared to the latter, the multi-step consideration is expected to provide better accuracy by reducing the propagation of errors in the backward induction.

Remark 2.3. *We could have also considered, as in the DBDP2 scheme, the automatic differentiation of $\widehat{\mathcal{U}}_i$ for the approximation of the gradient Z_{t_i}. However, as shown in the numerical tests of [HPW20], this approach leads to less accurate results than the DBDP1 algorithm which uses an additional neural network. Moreover, at least for theoretical analysis, it requires to optimize over C^1 neural networks, which is a restrictive assumption. Hence we focus on a DBDP1-type method.*

In the numerical implementation, the expectation defining the loss function J_i^{MB} in (2.8) is replaced by an empirical average leading to the so-called *generalization* (or estimation) error, largely studied in the statistical community, see [Gy02], and more recently [Hur+21], [BJK19] and the references therein. Moreover, recalling the parametrization $(\mathcal{U}^\theta, \mathcal{Z}^\theta)$ of neural network functions in $\mathcal{N}^\rho_{d,1,\ell,m} \times \mathcal{N}^\rho_{d,d,\ell,m}$, the minimization of the empirical average is amenable to stochastic gradient descent (SGD) extensively used in machine learning. More precisely, given a fixed time step $i = N-1, \dots, 0$, at each iteration of the SGD, we pick a sample $(X_j^k, \Delta W_j^k)_{j=i,\dots,N}$ of the Euler process and increment of Brownian motion $(X_j, \Delta W_j)_j$, $k = 1, \dots, K$, of mini-batch size K, and consider the empirical loss function:

$$\begin{aligned}
&\mathbb{J}_i^K(\theta) \\
&= \frac{1}{K}\sum_{k=1}^{K} \Big| g(X_N^k) - \sum_{j=i+1}^{N-1} f(t_j, X_j^k, \widehat{\mathcal{U}}_j(X_j^k), \widehat{\mathcal{Z}}_j(X_j^k))\Delta t_j - \sum_{j=i+1}^{N-1} \widehat{\mathcal{Z}}_j(X_j^k).\Delta W_j^k \\
&\qquad\qquad - \mathcal{U}^\theta(X_i^k) - f(t_i, X_i^k, \mathcal{U}^\theta(X_i^k), \mathcal{Z}^\theta(X_i^k))\Delta t_i - \mathcal{Z}^\theta(X_i^k).\Delta W_i^k \Big|^2, \qquad (2.9)
\end{aligned}$$

where $\widehat{\mathcal{U}}_j = \mathcal{U}_j^{\hat\theta_j}$, $\widehat{\mathcal{Z}}_j = \mathcal{Z}_j^{\hat\theta_j}$, and $\hat\theta_j$ is the resulting parameter from the SGD obtained at dates $j \in [\![i+1, N-1]\!]$. In practice, the number of iterations for SGD at the initial induction time $N-1$ should be large enough so as to learn accurately the value function $u(t_{N-1}, .)$ and its gradient $D_x u(t_{N-1}, .)$ via $\widehat{\mathcal{U}}^{\hat\theta_{N-1}}$ and $\widehat{\mathcal{Z}}^{\hat\theta_{N-1}}$. However, it is then expected that $(\widehat{\mathcal{U}}_j, \widehat{\mathcal{Z}}_j)$ does not vary a lot from $j = i+1$ to i, which means that at time i, one can design the SGD with initialization parameter equal to the resulting parameter from the previous SGD at time $i+1$, and then use few iterations to obtain accurate values of $\widehat{\mathcal{U}}_i$ and $\widehat{\mathcal{Z}}_i$. This observation allows to reduce significantly the computational time in (M)DBDP scheme when applying sequentially N SGD. The SGD algorithm for computing an approximate minimizer of the loss function induces the so-called *optimization* error, which has been extensively studied in the stochastic algorithm and machine learning communities, see [BM], [BF11], [BJK19], and the references therein.

Algorithm 1: MDBDP scheme.

Data: Initial parameter $\hat\theta_N$. A sequence of number of iterations $(S_i)_{i=0,\dots,N-1}$
for $i = N-1, \dots, 0$ **do**
  Initial parameter $\theta_i \leftarrow \hat\theta_{i+1}$
  Set $s = 1$
  while $s \le S_i$ **do**
    Pick a sample of $(X_j, \Delta W_j)_{j=i,\dots,N}$ of mini-batch size K
    Compute the gradient $\nabla \mathbb{J}_i^K(\theta)$ of $\mathbb{J}_i^K(\theta)$ defined in (2.9)
    Update $\theta_i \leftarrow \theta_i - \eta \nabla \mathbb{J}_i^K(\theta_i)$ with η learning rate
    $s \leftarrow s + 1$
  end
  Return $\hat\theta_i \leftarrow \theta_i$, $\widehat{\mathcal{U}}_i = \mathcal{U}^{\hat\theta_i}$, $\widehat{\mathcal{Z}}_i = \mathcal{Z}^{\hat\theta_i}$ `/* Update parameter, function and derivative */`
end

3 Convergence Analysis

This section is devoted to the approximation error and rate of convergence of the MDBDP, DS, and DBDP schemes described in Section 2.

We make the following standard assumptions on the coefficients of the forward-backward equation associated to semilinear PDE (1.1).

Assumption 3.1. *(i) $\mathcal{X}_0$ is square-integrable : $\mathcal{X}_0 \in L^2(\mathcal{F}_0, \mathbb{R}^d)$.*

(ii) The functions μ and σ are Lipschitz in $x \in \mathbb{R}^d$, uniformly in $t \in [0,T]$.

(iii) The generator function f is $1/2$-Hölder continuous in time and Lipschitz continuous in all other variables: $\exists\, [f]_{_L} > 0$ such that for all (t,x,y,z) and $(t',x',y',z') \in [0,T]\times\mathbb{R}^d\times\mathbb{R}\times\mathbb{R}^d$,

$$\begin{aligned} &|f(t,x,y,z) - f(t',x',y',z')| \\ &\leq\ [f]_{_L}\big(|t-t'|^{1/2} + |x-x'|_{_2} + |y-y'| + |z-z'|_{_2}\big). \end{aligned}$$

Moreover, $\sup_{t\in[0,T]} |f(t,0,0,0)| < \infty$.

(iv) The function g satisfies a linear growth condition.

Assumption 3.1 guarantees the existence and uniqueness of an adapted solution $(\mathcal{X}, Y, Z)$ to the forward-backward equation (2.2)-(2.3), satisfying

$$\mathbb{E}\Big[\sup_{0\leq t\leq T}|\mathcal{X}_t|_{_2}^2 + \sup_{0\leq t\leq T}|Y_t|^2 + \int_0^T |Z_t|_{_2}^2 \mathrm{d}t\Big] <\ \infty,$$

(see for instance Theorem 3.3.1, Theorem 4.2.1, Theorem 4.3.1 from [Zha17]). Given the time grid $\pi = \{t_i : i = 0,\dots,N\}$, let us introduce the L^2-regularity of Z:

$$\varepsilon^Z(\pi) :=\ \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} |Z_t - \bar{Z}_{t_i}|_{_2}^2 \mathrm{d}t\Big], \quad \text{with}\ \ \bar{Z}_{t_i}\ :=\ \frac{1}{\Delta t_i}\mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} Z_t \mathrm{d}t\Big].$$

Since $\bar{Z}$ is a L^2-projection of Z, we know that $\varepsilon^Z(\pi)$ converges to zero when $|\pi|$ goes to zero. Moreover, as shown in [Zha04], when g is also Lipschitz, we have

$$\varepsilon^Z(\pi) =\ O(|\pi|).$$

Here, the standard notation $O(|\pi|)$ means that $\limsup_{|\pi|\to 0}\ |\pi|^{-1}O(|\pi|) < \infty$.

Lemma 3.1. *Under Assumption 3.2 (ii), the following standard estimate for the Euler-Maruyama scheme holds when $\Delta t_i \to 0$*

$$\mathbb{E}|X_{i+1}^x - X_{i+1}^{x'}|_2^2 \leq (1 + C\Delta t_i)|x - x'|_2^2,$$

where $X_{i+1}^x := x + \mu(t_i,x)\Delta t_i + \sigma(t_i,x)\Delta W_i$.

Proof. By expanding the square, simply notice that the dominant terms when $\Delta t_i \to 0$ are of older Δt_i because the term of order $\sqrt{\Delta t_i}$, namely $(x-x')\cdot(\sigma(t_i,x) - \sigma(t_i,x'))\Delta W_i$ has a null expectation and all other terms are dominated by Δt_i. □

3.1 Convergence of the MDBDP Scheme

We fix classes of functions $\mathcal{N}_i$ and $\mathcal{N}_i'$ for the approximations respectively of the solution and its gradient, and define $(\widehat{\mathcal{U}}_i^{(1)}, \widehat{\mathcal{Z}}_i^{(1)})$ as the output of the MDBDP scheme at times t_i, $i = 0,\dots,N$.

Let us define (implicitly) the process

$$\begin{cases} V_i^{(1)} &=\ \mathbb{E}_i\Big[g(X_N) - f\big(t_i, X_i, V_i^{(1)}, \overline{\widehat{Z}}_i^{(1)}\big)\Delta t_i - \displaystyle\sum_{j=i+1}^{N-1} f\big(t_j, X_j, \widehat{\mathcal{U}}_j^{(1)}(X_j), \widehat{\mathcal{Z}}_j^{(1)}(X_j)\big)\Delta t_j\Big], \\ \overline{\widehat{Z}}_i^{(1)} &=\ \mathbb{E}_i\Big[\frac{g(X_N)\Delta W_i}{\Delta t_i} - \displaystyle\sum_{j=i+1}^{N-1} f\big(t_j, X_j, \widehat{\mathcal{U}}_j^{(1)}(X_j), \widehat{\mathcal{Z}}_j^{(1)}(X_j)\big)\frac{\Delta W_i \Delta t_j}{\Delta t_i}\Big],\ i = 0,\dots,N, \end{cases} \tag{3.1}$$

and notice by the Markov property of the discretized forward process $(X_i)_i$ that

$$V_i^{(1)} \;=\; v_i^{(1)}(X_i), \qquad \overline{\widehat{Z}}_i^{(1)} \;=\; \widehat{z}_i^{(1)}(X_i), \quad i = 0, \dots, N, \tag{3.2}$$

for some deterministic functions $v_i^{(1)}, \widehat{z}_i^{(1)}$. Let us then introduce

$$\varepsilon_i^{1,y} \;:=\; \inf_{\mathcal{U} \in \mathcal{N}_i} \mathbb{E}\big|v_i^{(1)}(X_i) - \mathcal{U}(X_i)\big|^2, \qquad \varepsilon_i^{1,z} \;:=\; \inf_{\mathcal{Z} \in \mathcal{N}_i'} \mathbb{E}\big|\widehat{z}_i^{(1)}(X_i) - \mathcal{Z}(X_i)\big|_2^2,$$

for $i = 0, \dots, N-1$, which represent the L^2-approximation errors of the functions $v_i^{(1)}, \widehat{z}_i^{(1)}$ in the classes $\mathcal{N}_i$ and $\mathcal{N}_i'$.

Theorem 3.1 (Approximation error of MDBDP)**.** *Under Assumption 3.1, there exists a constant $C > 0$ (depending only on the data μ, σ, f, g, d, T) such that in the limit $|\pi| \to 0$*

$$\begin{aligned} &\sup_{i \in [\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \big|Z_s - \widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2 \, \mathrm{d}s\Big] \\ &\leq\; C\Big(\mathbb{E}\big|g(\mathcal{X}_T) - g(X_N)\big|^2 + |\pi| + \varepsilon^Z(\pi) + \sum_{j=0}^{N-1} (\varepsilon_j^{1,y} + \Delta t_j \varepsilon_j^{1,z})\Big). \end{aligned} \tag{3.3}$$

Remark 3.1. The upper bound in (3.3) consists of four terms. The first three terms correspond to the time discretization of BSDE, similarly as in [Zha04], [BT04], namely (i) the strong approximation of the terminal condition (depending on the forward scheme and g), and converging to zero, as $|\pi|$ goes to zero, with a rate $|\pi|$ when g is Lipschitz, (ii) the strong approximation of the forward Euler scheme, and the L^2-regularity of Y, which gives a convergence of order $|\pi|$, (iii) the L^2-regularity of Z. Finally, the last term is the approximation error by the chosen class of functions. Note that the approximation error $\sum_{j=0}^{N-1}(\varepsilon_j^{1,y} + \Delta t_j \varepsilon_j^{1,z})$ in (3.3) is better than the one for the DBDP scheme derived in [HPW20], with an order $\sum_{j=0}^{N-1}(N\varepsilon_j^{1,y} + \varepsilon_j^{1,z})$. In the work [GT16] which introduced the multistep scheme with linear regression, the authors noticed the same improvement in the error propagation in comparison with the one-step classical scheme [GLW05]. □

We next study convergence for the approximation error of the MDBDP scheme, for a specific choice of functions classes $\mathcal{N}_i$ and $\mathcal{N}_i'$ and with the additional assumption that f does not depend on z.

Assumption 3.2. *The generator function f is independent of z. Namely, for all $(t, x, y, z, z') \in [0, T] \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$,*

$$f(t, x, y, z) = f(t, x, y, z').$$

Actually, if f is linear in z: $f(t, x, y, z) = \bar{f}(t, x, y) + \lambda(t, x).z$, one can boil down to Assumption 3.2 for $\bar{f}$ by incorporating the linearity in the drift function μ, namely with the modified drift: $\bar{\mu}(t, x) = \mu(t, x) - \sigma\lambda(t, x)$.

Proposition 3.1 (Rate of convergence of MDBDP)**.** *Let Assumption 3.1 and Assumption 3.2 hold, and assume that $\mathcal{X}_0 \in L^{2+\delta}(\mathcal{F}_0, \mathbb{R}^d)$, for some $\delta > 0$, and g is $[g]-$Lipschitz. Then, there exists a bounded sequence K_i (uniformly in i, N) such that for GroupSort neural networks classes $\mathcal{N}_i = \mathcal{G}^{\zeta_\kappa}_{K_i,d,1,\ell,m}$, and $\mathcal{N}_i' = \mathcal{G}^{\zeta_\kappa}_{\sqrt{\frac{d}{\Delta t_i}}K_i,d,d,\ell,m}$, we have*

$$\sup_{i \in [\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \big|Z_s - \widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2 \, \mathrm{d}s\Big] \;=\; O(1/N),$$

with a grouping size $\kappa = O(2\sqrt{d}N^2)$, depth $\ell + 1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((2\sqrt{d}N^2)^{d^2-1})$ in the case $d > 1$. If $d = 1$, take $\kappa = O(N^2)$, depth $\ell + 1 = 3$ and width $\sum_{i=0}^{\ell-1} m_i = O(N^2)$. Here, the constants in the $O(\cdot)$ term depend only on $\mu, \sigma, f, g,\ d, T, \mathcal{X}_0$.

3.2 Convergence of the DS Scheme

We consider classes $\mathcal{N}_i^{\gamma,\eta}$ of differentiable γ_i-Lipschitz functions with η_i-Lipschitz derivative for sequences $\gamma = (\gamma_i)_i$, $\eta = (\eta_i)_i$ and define $\widehat{\mathcal{U}}_i^{(2)}$ as the output of the DS scheme at times t_i, $i = 0, \dots, N.$.

Let us define the process

$$V_i^{(2)} = \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - f\big(t_i, X_i, \mathbb{E}_i[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})], \mathbb{E}_i[\sigma(t_i, X_i)^\intercal D_x[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})]\big)\Delta t_i\Big], \quad (3.4)$$

for $i \in [\![0, N-1]\!]$, and $V_N^{(2)} = \widehat{\mathcal{U}}_N^{(2)}(X_N)$. By the Markov property of $(X_i)_i$, we have $V_i^{(2)} = v_i^{(2)}(X_i)$, for some functions $v_i^{(2)} : \mathbb{R}^d \to \mathbb{R}$, $i \in [\![0, N-1]\!]$, and we introduce

$$\varepsilon_i^{\gamma,\eta} = \begin{cases} \inf_{\mathcal{U}\in\mathcal{N}_i^{\gamma,\eta}} \mathbb{E}\big|v_i^{(2)}(X_i) - \mathcal{U}(X_i)\big|^2, & i = 0, \dots, N-1, \\ \inf_{\mathcal{U}\in\mathcal{N}_i^{\gamma,\eta}} \mathbb{E}\big|g(X_N) - \mathcal{U}(X_N)\big|^2, & i = N. \end{cases}$$

the L^2-approximation error in the class $\mathcal{N}_i^{\gamma,\eta}$ of the functions $v_i^{(2)}$, $i = 0, \dots, N-1$, and g.

Theorem 3.2 (Approximation error of DS)**.** *Let Assumption 3.1 hold, and assume that $\mathcal{X}_0 \in L^4(\mathcal{F}_0, \mathbb{R}^d)$. Then, there exists a constant $C > 0$ (depending only on $\mu, \sigma, f, g, d, T, \mathcal{X}_0$) such that in the limit $|\pi| \to 0$*

$$\sup_{i\in[\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_i)\big|^2 \le C\Big(\mathbb{E}\big|g(X_N) - g(\mathcal{X}_T)\big|^2 + |\pi| + \varepsilon^Z(\pi) + \max_i\big[\gamma_i^2, \eta_i^2\big]|\pi| + \varepsilon_N^{\gamma,\eta} + N\sum_{i=0}^{N-1}\varepsilon_i^{\gamma,\eta}\Big). \quad (3.5)$$

Remark 3.2. We retrieve a similar error as in the analysis of the DBDP2 scheme derived in [HPW20]. Notice that when g is C^1, one can choose to initialize the DS scheme with $\widehat{\mathcal{U}}_N = g$, and the term $\varepsilon_N^{\gamma,\eta}$ is removed in (3.5). □

The GroupSort neural networks being only continuous but not differentiable, we are not able to express a convergence rate for the Deep Splitting scheme in terms of the architecture and number of neurons to choose, like in Propositions 3.1, 3.2. It would require a quantitative approximation result for C^1 neural networks with bounded Lipschitz gradient, and this is left to future research.

3.3 Convergence of the DBDP Scheme

We consider classes of functions $\mathcal{N}_i$ and $\mathcal{N}_i'$ for the approximations of the solution and its gradient, and define $(\widehat{\mathcal{U}}_i^{(3)}, \widehat{\mathcal{Z}}_i^{(3)})$ as the output of the DBDP scheme at times t_i, $i = 0, \dots, N$. Let us define (implicitly) the process

$$\begin{cases} V_i^{(3)} &= \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}) - f\big(t_i, X_i, V_i^{(3)}, \overline{\widehat{\mathcal{Z}}_i}^{(3)}\big)\Delta t_i\Big] \\ \overline{\widehat{\mathcal{Z}}_i}^{(3)} &= \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1})\frac{\Delta W_i}{\Delta t_i}\Big], \quad i = k, \dots, N-1. \end{cases}$$

and notice by the Markov property of the discretized forward process $(X_i)_i$ that

$$V_i^{(3)} = v_i^{(3)}(X_i), \qquad \overline{\widehat{\mathcal{Z}}_i} = \widehat{z}_i^{(3)}(X_i), \quad i = 0, \dots, N, \quad (3.6)$$

for some deterministic functions $v_i^{(3)}, \widehat{z}_i^{(3)}$. Let us then introduce

$$\varepsilon_i^{3,y} := \inf_{\mathcal{U}\in\mathcal{N}_i} \mathbb{E}\big|v_i^{(3)}(X_i) - \mathcal{U}(X_i)\big|^2, \qquad \varepsilon_i^{3,z} := \inf_{\mathcal{Z}\in\mathcal{N}_i'} \mathbb{E}\big|\widehat{z}_i^{(3)}(X_i) - \mathcal{Z}(X_i)\big|_2^2,$$

for $i = 0, \dots, N-1$, which represent the L^2-approximation errors of the functions $v_i^{(3)}, \widehat{z}_i^{(3)}$ in the classes $\mathcal{N}_i$ and $\mathcal{N}_i'$.

Theorem 3.3 (Huré, Pham, Warin [HPW20] : Approximation error of DBDP)**.** *Under Assumption 3.1, there exists a constant $C > 0$ (depending only on the data μ, σ, f, g, d, T) such that in the limit $|\pi| \to 0$*

$$\sup_{i\in[\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(3)}(X_i)\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_s - \widehat{\mathcal{Z}}_i^{(3)}(X_i)\big|_2^2 \,\mathrm{d}s\Big]$$
$$\leq\ C\Big(\mathbb{E}\big|g(\mathcal{X}_T) - g(X_N)\big|^2 + |\pi| + \varepsilon^Z(\pi) + N\sum_{j=0}^{N-1}(\varepsilon_j^{3,y} + \Delta t_j \varepsilon_j^{3,z})\Big). \tag{3.7}$$

We next study convergence rate for the approximation error of the DBDP scheme, and need to specify the class of network functions $\mathcal{N}_i$ and $\mathcal{N}_i'$.

Proposition 3.2 (Rate of convergence of DBDP)**.** *Let Assumption 3.1 hold, and assume that $\mathcal{X}_0 \in L^{2+\delta}(\mathcal{F}_0, \mathbb{R}^d)$, for some $\delta > 0$, and g is $[g]-$Lipschitz. Then, there exists a bounded sequence K_i (uniformly in i, N) such that for $\mathcal{N}_i = \mathcal{G}^{\zeta_\kappa}_{K_i,d,1,\ell,m}$, and $\mathcal{N}_i' = \mathcal{G}^{\zeta_\kappa}_{\sqrt{\frac{d}{\Delta t_i}}K_i,d,d,\ell,m}$, we have*

$$\sup_{i\in[\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(3)}(X_i)\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_s - \widehat{\mathcal{Z}}_i^{(3)}(X_i)\big|_2^2 \,\mathrm{d}s\Big] = O(1/N),$$

with a grouping size $\kappa = O(2\sqrt{d}N^3)$, depth $\ell + 1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((2\sqrt{d}N^3)^{d^2-1})$ in the case $d > 1$. If $d = 1$, take $\kappa = O(N^3)$, depth $\ell + 1 = 3$ and width $\sum_{i=0}^{\ell-1} m_i = O(N^3)$. Here, the constants in the $O(\cdot)$ term depend only on $\mu, \sigma, f, g,\ d, T, \mathcal{X}_0$.

4 Proof of the Main Theoretical Results

4.1 Proof of Theorem 3.1

Let us introduce the processes $(\bar{V}_i, \bar{Z}_i)_i$ arising from the time discretization of the BSDE (2.3), and defined by the *implicit* backward Euler scheme:

$$\left\{\begin{array}{rcl} \bar{V}_i^{(1)} &=& \mathbb{E}_i\Big[\bar{V}_{i+1}^{(1)} - f\big(t_i, X_i, \bar{V}_i^{(1)}, \bar{Z}_i^{(1)}\big)\Delta t_i\Big] \\ \bar{Z}_i^{(1)} &=& \mathbb{E}_i\Big[\bar{V}_{i+1}^{(1)}\frac{\Delta W_i}{\Delta t_i}\Big], \quad i = 0, \ldots, N-1, \end{array}\right. \tag{4.1}$$

starting from $\bar{V}_N^{(1)} = g(X_N)$. We recall from [Zha04] the time discretization error:

$$\sup_{i\in[\![0,N]\!]} \mathbb{E}\big|Y_{t_i} - \bar{V}_i^{(1)}\big|^2 + \mathbb{E}\Big[\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \big|Z_s - \bar{Z}_i^{(1)}\big|_2^2 \,\mathrm{d}s\Big]$$
$$\leq C\Big(\mathbb{E}\big|g(\mathcal{X}_T) - g(X_N)\big|^2 + |\pi| + \varepsilon^Z(\pi)\Big), \tag{4.2}$$

for some constant C depending only on the coefficients satisfying Assumption 3.1.

Let us introduce the auxiliary process

$$\hat{V}_i^{(1)} = \mathbb{E}_i\Big[g(X_N) - \sum_{j=i}^{N-1} f\big(t_j, X_j, \widehat{\mathcal{U}}_j^{(1)}(X_j), \widehat{\mathcal{Z}}_j^{(1)}(X_j)\big)\Delta t_j\Big], \quad i = 0, \ldots, N, \tag{4.3}$$

and notice by the tower property of conditional expectations that we have the recursive relations:

$$\hat{V}_i^{(1)} = \mathbb{E}_i\Big[\hat{V}_{i+1}^{(1)} - f\big(t_i, X_i, \widehat{\mathcal{U}}_i^{(1)}(X_i), \widehat{\mathcal{Z}}_i^{(1)}(X_i)\big)\Delta t_i\Big], \quad i = 0, \ldots, N-1. \tag{4.4}$$

Observe also that $\overline{\widehat{Z}}_i^{(1)}$ defined in (3.1) satisfies

$$\overline{\widehat{Z}}_i^{(1)} = \mathbb{E}_i\Big[\hat{V}_{i+1}^{(1)}\frac{\Delta W_i}{\Delta t_i}\Big], \quad i = 0, \ldots, N-1. \tag{4.5}$$

We now decompose the approximation error, for $i \in [\![0, N-1]\!]$, into

$$\begin{aligned}
&\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2 \\
&\leq\ 4\Big(\mathbb{E}\big|Y_{t_i} - \bar{V}_i^{(1)}\big|^2 + \mathbb{E}\big|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}\big|^2 + \mathbb{E}\big|\hat{V}_i^{(1)} - V_i^{(1)}\big|^2 + \mathbb{E}\big|V_i^{(1)} - \hat{\mathcal{U}}_i^{(1)}(X_i)\big|^2\Big) \\
&=:\ 4(I_i^1 + I_i^2 + I_i^3 + I_i^4),
\end{aligned} \tag{4.6}$$

and analyze each of these contribution terms. In the sequel, C denotes a generic constant independent of π that may vary from line to line, and depending only on the coefficients satisfying Assumption 3.1. Notice that the first contribution term is the time discretization error for BSDE given by (4.2), and we shall study the three other terms in the following steps.

Step 1. Fix $i \in [\![0, N-1]\!]$. From the definition (3.1) of $V_i^{(1)}$ and by the martingale representation theorem, there exists a square integrable process $\{\widehat{Z}_s^{(1)}, t_i \leq s \leq T\}$ s.t.

$$\begin{aligned}
&g(X_N) - f\big(t_i, X_i, V_i^{(1)}, \overline{\widehat{Z}_i^{(1)}}\big)\Delta t_i - \sum_{j=i+1}^{N-1} f\big(t_j, X_j, \widehat{\mathcal{U}}_j^{(1)}(X_j), \widehat{\mathcal{Z}}_j^{(1)}(X_j)\big)\Delta t_j \\
&=\ V_i + \int_{t_i}^{t_N} \widehat{Z}_s^{(1)}.\mathrm{d}W_s.
\end{aligned} \tag{4.7}$$

From the definition (3.1) of $\overline{\widehat{Z}}_i^{(1)}$, and by Itô isometry, we then have

$$\overline{\widehat{Z}}_i^{(1)} \ =\ \frac{\mathbb{E}_i\big[\int_{t_i}^{t_{i+1}} \widehat{Z}_s^{(1)}\mathrm{d}s\big]}{\Delta t_i}, \ \text{ i.e. } \ \mathbb{E}_i\Big[\int_{t_i}^{t_{i+1}} \big(\widehat{Z}_s^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big)\mathrm{d}s\Big] \ =\ 0. \tag{4.8}$$

Plugging (4.7) into (2.8), we see that the loss function of the MDBDP scheme can be rewritten as

$$\begin{aligned}
&\mathcal{J}_i^{MB}(\mathcal{U}_i, \mathcal{Z}_i) \\
&=\ \mathbb{E}\Big|V_i^{(1)} - \mathcal{U}_i(X_i) + \Delta t_i\big[f\big(t_i, X_i, V_i^{(1)}, \overline{\widehat{Z}}_i^{(1)}\big) - f\big(t_i, X_i, \mathcal{U}_i(X_i), \mathcal{Z}_i(X_i)\big)\big] \\
&\qquad + \sum_{j=i+1}^{N-1} \int_{t_j}^{t_{j+1}} \big[\widehat{Z}_s^{(1)} - \widehat{\mathcal{Z}}_j(X_j)\big].\mathrm{d}W_s + \int_{t_i}^{t_{i+1}} \big[\widehat{Z}_s^{(1)} - \mathcal{Z}_i(X_i)\big].\mathrm{d}W_s\Big|^2 \\
&= \widetilde{\mathcal{J}}_i^{MB}(\mathcal{U}_i, \mathcal{Z}_i) + \mathbb{E}\Big[\sum_{j=i}^{N-1} \int_{t_j}^{t_{j+1}} \big|\widehat{Z}_s^{(1)} - \overline{\widehat{Z}}_j^{(1)}\big|_2^2 \mathrm{d}s\Big] \\
&\quad + \sum_{j=i+1}^{N-1} \Delta t_j \mathbb{E}\big|\overline{\widehat{Z}}_j^{(1)} - \widehat{\mathcal{Z}}_j(X_j)\big|_2^2,
\end{aligned} \tag{4.9}$$

where we use (4.8), and

$$\begin{aligned}
&\widetilde{\mathcal{J}}_i^{MB}(\mathcal{U}_i, \mathcal{Z}_i) \\
&:= \mathbb{E}\Big|V_i^{(1)} - \mathcal{U}_i(X_i) + \Delta t_i\big[f\big(t_i, X_i, V_i^{(1)}, \overline{\widehat{Z}}_i^{(1)}\big) - f\big(t_i, X_i, \mathcal{U}_i(X_i), \mathcal{Z}_i(X_i)\big)\big]\Big|^2 \\
&\qquad +\ \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)} - \mathcal{Z}_i(X_i)\big|_2^2.
\end{aligned}$$

It is clear by Lipschitz continuity of f in Assumption 3.1 that

$$\widetilde{\mathcal{J}}_i^{MB}(\mathcal{U}_i, \mathcal{Z}_i) \leq\ C\Big(\mathbb{E}\big|V_i^{(1)} - \mathcal{U}_i(X_i)\big|^2 + \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)} - \mathcal{Z}_i(X_i)\big|_2^2\Big). \tag{4.10}$$

On the other hand, by the Young inequality: $(1-\beta)a^2 + \big(1 - \frac{1}{\beta}\big)b^2 \leq (a+b)^2 \leq (1+\beta)a^2 +$

$\big(1+\frac{1}{\beta}\big)b^2$, for all $(a,b)\in\mathbb{R}^2$, and $\beta>0$, we have

$$
\begin{aligned}
&\widetilde{\mathcal{J}}_i^{MB}(\mathcal{U}_i,\mathcal{Z}_i)\\
&\geq\ (1-\beta)\mathbb{E}\big|V_i^{(1)}-\mathcal{U}_i(X_i)\big|^2+\ \Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\mathcal{Z}_i(X_i)\big|_2^2\\
&+\qquad\Big(1-\frac{1}{\beta}\Big)|\Delta t_i|^2\mathbb{E}\big|f\big(t_i,X_i,\mathcal{U}_i(X_i),\mathcal{Z}_i(X_i)\big)-f\big(t_i,X_i,V_i^{(1)},\overline{\widehat{Z}}_i^{(1)}\big)\big|^2\\
&\geq\ (1-\beta)\mathbb{E}\big|V_i^{(1)}-\mathcal{U}_i(X_i)\big|^2+\ \Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\mathcal{Z}_i(X_i)\big|_2^2\\
&\quad-\frac{2[f]_L^2}{\beta}|\Delta t_i|^2\Big(\mathbb{E}\big|\mathcal{U}_i(X_i)-V_i^{(1)}\big|^2+\mathbb{E}\big|\mathcal{Z}_i(X_i)-\overline{\widehat{Z}}_i^{(1)}\big|_2^2\Big)\\
&\geq\ \Big(1-\big(4[f]_L^2+\frac{1}{2}\big)\Delta t_i\Big)\mathbb{E}\big|V_i^{(1)}-\mathcal{U}_i(X_i)\big|^2+\frac{1}{2}\Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\mathcal{Z}_i(X_i)\big|_2^2, \qquad (4.11)
\end{aligned}
$$

where we use the Lipschitz continuity of f in the second inequality, and choose explicitly $\beta=4[f]_L^2\Delta t_i$ (<1 for Δt_i small enough) in the last one. By applying inequality (4.11) to $(\mathcal{U}_i,\mathcal{Z}_i)=(\widehat{\mathcal{U}}_i^{(1)},\widehat{\mathcal{Z}}_i^{(1)})$, which is a minimizer of $\tilde{\mathcal{J}}_i^{MB}$ by (4.9), and combining with (4.10), this yields for Δt_i small enough and for all functions $\mathcal{U}_i$, $\mathcal{Z}_i$:

$$
\begin{aligned}
&\mathbb{E}\big|V_i^{(1)}-\widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2+\Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2\\
&\leq\ C\Big(\mathbb{E}\big|V_i-\mathcal{U}_i(X_i)\big|^2+\Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i-\mathcal{Z}_i(X_i)\big|_2^2\Big).
\end{aligned}
$$

By minimizing over $\mathcal{U}_i,\mathcal{Z}_i$ in the right hand side, we get the approximation error in the classes $\mathcal{N}_i,\mathcal{N}_i'$ of the regressed functions $V_i^{(1)}$, $\overline{\widehat{Z}}_i^{(1)}$:

$$
\mathbb{E}\big|V_i^{(1)}-\widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2+\Delta t_i\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2\leq\ C(\varepsilon_i^{1,y}+\Delta t_i\varepsilon_i^{1,z}). \qquad (4.12)
$$

Step 2. From the expressions of $V_i^{(1)}$ and $\hat{V}_i^{(1)}$ in (3.1), (4.3), and by Lipschitz continuity of f, we have by (4.12):

$$
\begin{aligned}
\mathbb{E}\big|\hat{V}_i^{(1)}-V_i^{(1)}\big|^2=\ &\Delta t_i^2\mathbb{E}\Big|\mathbb{E}_i\big[f\big(t_i,X_i,V_i^{(1)},\overline{\widehat{Z}}_i^{(1)}\big)-f\big(t_i,X_i,\widehat{\mathcal{U}}_i^{(1)}(X_i),\widehat{\mathcal{Z}}_i^{(1)}(X_i)\big)\big]\Big|^2\\
&\leq\ 2[f]_L^2|\Delta t_i|^2\Big(\mathbb{E}\big|V_i^{(1)}-\widehat{\mathcal{U}}_i^{(1)}(X_i)\big|^2+\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2\Big)\\
&\leq\ C\Delta t_i(\varepsilon_i^{1,y}+\Delta t_i\varepsilon_i^{1,z}),\qquad i=0,\dots,N. \qquad (4.13)
\end{aligned}
$$

Step 3. From the recursive expressions of $\bar{V}_i^{(1)}$, $\hat{V}_i^{(1)}$ in (4.1), (4.4), and applying the Young, the Cauchy-Schwarz inequalities, together with the Lipschitz condition of f, we get for $\beta>0$:

$$
\begin{aligned}
&\mathbb{E}\big|\bar{V}_i^{(1)}-\hat{V}_i^{(1)}\big|^2\\
&\leq\ (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)}-\hat{V}_{i+1}^{(1)}\big]\Big|^2+2[f]_L^2\Big(1+\frac{1}{\beta}\Big)|\Delta t_i|^2\Big(\mathbb{E}\big|\bar{V}_i^{(1)}-\hat{\mathcal{U}}_i^{(1)}(X_i)\big|^2+\mathbb{E}\big|\bar{Z}_i^{(1)}-\widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2\Big)\\
&\leq\ (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)}-\hat{V}_{i+1}^{(1)}\big]\Big|^2+2[f]_L^2\Big(1+\frac{1}{\beta}\Big)|\Delta t_i|^2\big(3\mathbb{E}|\bar{V}_i^{(1)}-\hat{V}_i^{(1)}|^2+2\mathbb{E}\big|\bar{Z}_i^{(1)}-\overline{\widehat{Z}}_i^{(1)}\big|_2^2\big)\\
&+\ 2[f]_L^2\Big(1+\frac{1}{\beta}\Big)|\Delta t_i|^2\Big(3\mathbb{E}|\hat{V}_i^{(1)}-V_i^{(1)}|^2+3\mathbb{E}|V_i^{(1)}-\widehat{\mathcal{U}}_i^{(1)}(X_i)|^2+2\mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)}-\widehat{\mathcal{Z}}_i(X_i)\big|_2^2\Big)\\
&\leq\ (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)}-\hat{V}_{i+1}^{(1)}\big]\Big|^2+(1+\beta)\frac{2[f]_L^2|\Delta t_i|^2}{\beta}\big(3\mathbb{E}|\bar{V}_i^{(1)}-\hat{V}_i^{(1)}|^2+2\mathbb{E}\big|\bar{Z}_i^{(1)}-\overline{\widehat{Z}}_i^{(1)}\big|_2^2\big)\\
&+\ C[f]_L^2\Big(1+\frac{1}{\beta}\Big)\Delta t_i(\varepsilon_i^{1,y}+\Delta t_i\varepsilon_i^{1,z}), \qquad (4.14)
\end{aligned}
$$

where we use (4.12), (4.13) in the last inequality. Moreover, by (4.1), (4.5), we have

$$\begin{aligned}\Delta t_i\big(\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big) &= \mathbb{E}_i\Big[\Delta W_i\big(\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big)\Big] \\ &= \mathbb{E}_i\Big[\Delta W_i\Big(\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)} - \mathbb{E}_i\big[\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big]\Big)\Big],\end{aligned}$$

and thus by the Cauchy-Schwarz inequality

$$\Delta t_i \mathbb{E}\big|\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big|_2^2 \le d\Big(\mathbb{E}\big|\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big]\Big|^2\Big). \tag{4.15}$$

Plugging into (4.14), and choosing $\beta = 4d[f]_L^2 \Delta t_i$, gives

$$\begin{aligned}&(1 - C\Delta t_i)\mathbb{E}\big|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}\big|^2 \\ &\le (1 + C\Delta t_i)\mathbb{E}\big|\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big|^2 + (1 + C\Delta t_i)\big(\varepsilon_i^{1,y} + \Delta t_i \varepsilon_i^{1,z}\big)\end{aligned}$$

By discrete Gronwall lemma, and recalling that $\bar{V}_N^{(1)} = \hat{V}_N^{(1)}$ $(= g(X_N))$, we then obtain

$$\sup_{i \in [\![0,N]\!]} \mathbb{E}\big|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}\big|^2 \le C \sum_{i=0}^{N-1} \big(\varepsilon_i^{1,y} + \Delta t_i \varepsilon_i^{1,z}\big). \tag{4.16}$$

The required bound for the approximation error on Y follows by plugging (4.2), (4.12), (4.13), and (4.16) into (4.6).

Step 4. We decompose the approximation error for the Z component into three terms

$$\begin{aligned}&\mathbb{E}\Big[\sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \big|Z_s^{(1)} - \widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2 \, \mathrm{d}s\Big] \\ &\le 3 \sum_{i=0}^{N-1} \Big(\mathbb{E}\Big[\int_{t_i}^{t_{i+1}} \big|Z_s^{(1)} - \bar{Z}_i^{(1)}\big|_2^2 \, \mathrm{d}s\Big] + \Delta t_i \mathbb{E}\big|\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big|_2^2 + \Delta t_i \mathbb{E}\big|\overline{\widehat{Z}}_i^{(1)} - \widehat{\mathcal{Z}}_i^{(1)}(X_i)\big|_2^2\Big)\end{aligned} \tag{4.17}$$

By summing the inequality (4.15) (recalling that $\bar{V}_N^{(1)} = \hat{V}_N^{(1)}$), and using (4.14), we have for $\beta \in (0,1)$:

$$\begin{aligned}&\sum_{i=0}^{N-1} \Delta t_i \mathbb{E}|\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}|^2 \\ &\le d \sum_{i=0}^{N-1} \Big(\mathbb{E}\big|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}\big|^2 - \mathbb{E}\big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big]\big|^2\Big) \\ &\le d \sum_{i=0}^{N-1} \Big(\beta \mathbb{E}\Big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big]\Big|^2 + \big(1 + \frac{1}{\beta}\big)\big(2[f]_L^2 |\Delta t_i|^2\big)\big(3\mathbb{E}|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}|^2 + 2\mathbb{E}\big|\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big|_2^2\big) \\ &\qquad + C[f]_L^2\big(1 + \frac{1}{\beta}\big)\Delta t_i(\varepsilon_i^{1,y} + \Delta t_i \varepsilon_i^{1,z})\Big) \\ &\le d \sum_{i=0}^{N-1} \Big(\frac{8d[f]_L^2 \Delta t_i}{1 - 8d[f]_L^2 \Delta t_i} \mathbb{E}\big|\mathbb{E}_i\big[\bar{V}_{i+1}^{(1)} - \hat{V}_{i+1}^{(1)}\big]\big|^2 + \frac{3}{4d}\Delta t_i \mathbb{E}|\bar{V}_i^{(1)} - \hat{V}_i^{(1)}|^2 + \frac{C}{8d}(\varepsilon_i^{1,y} + \Delta t_i \varepsilon_i^{1,z})\Big) \\ &\qquad + \frac{1}{2}\sum_{i=0}^{N-1} \Delta t_i \mathbb{E}\big|\bar{Z}_i^{(1)} - \overline{\widehat{Z}}_i^{(1)}\big|_2^2,\end{aligned} \tag{4.18}$$

by choosing explicitly $\beta = \frac{8d[f]_L^2 \Delta t_i}{1 - 8d[f]_L^2 \Delta t_i} = O(\Delta t_i)$ for Δt_i small enough. Plugging (4.2), (4.12), (4.16), and (4.18) (using the Jensen inequality) into (4.17), this proves the required bound for the approximation error on Z, and completes the proof. □

4.2 Proof of Proposition 3.1

Let us introduce the flow of the Euler scheme (X_i) by:

$$X_{j+1}^{k,x} := X_j^{k,x} + \mu(t_j, X_j^{k,x})\Delta t_j + \sigma(t_j, X_j^{k,x})\Delta W_j,\ j = k, \dots, N,$$

starting from $X_k^{k,x} = x$ at time step $j = k \in \mathbb{N}^*$. Under Assumption 3.2, f does not depend on z so by slight abuse of notation we write $f(t,x,y) = f(t,x,y,z)$. Define

$$\begin{cases} V_{i,1}^{k,x} &= \mathbb{E}_i\Big[g(X_N^{k,x}) - f\big(t_i, X_i^{k,x}, V_{i,1}^{k,x}\big)\Delta t_i - \sum_{j=i+1}^{N-1} f\big(t_j, X_j^{k,x}, \widehat{\mathcal{U}}_j^{(1)}(X_j^{k,x})\big)\Delta t_j\Big], \\ \hat{V}_{i,1}^{k,x} &= \mathbb{E}_i\Big[g(X_N^{k,x}) - \sum_{j=i}^{N-1} f\big(t_j, X_j^{k,x}, \widehat{\mathcal{U}}_j^{(1)}(X_j^{k,x})\big)\Delta t_j\Big], \quad i = k, \dots, N, \\ \overline{\widehat{Z}}_{i,1}^{k,x} &= \mathbb{E}_i\Big[V_{i+1,1}^{k,x}\frac{\Delta W_i}{\Delta t_i}\Big],\ i = k, \dots, N, \end{cases}$$

and observe that we have the recursive relations:

$$\begin{aligned} \hat{V}_{i,1}^{k,x} &= \mathbb{E}_i\Big[\hat{V}_{i+1,1}^{k,x} - f\big(t_i, X_i^{k,x}, \widehat{\mathcal{U}}_i^{(1)}(X_i^{k,x})\big)\Delta t_i\Big],\ i = k, \dots, N, \\ V_{i,1}^{k,x} &= \mathbb{E}_i\Big[\hat{V}_{i+1,1}^{k,x} - f\big(t_i, X_i^{k,x}, V_{i,1}^{k,x}\big)\Delta t_i\Big],\ i = k, \dots, N. \end{aligned}$$

Notice by the Markov property of the discretized forward process $(X_i^{k,x})_i$ that

$$V_{j,1}^{k,x} = v_j^{(1)}(X_j^{k,x}),\ \hat{V}_{j,1}^{k,x} = \hat{v}_j^{(1)}(X_j^{k,x}),\ \overline{\widehat{Z}}_{j,1}^{k,x} = \hat{z}_j^{(1)}(X_j^{k,x}),\ j = k, \dots, N$$

for some deterministic function $v_j^{(1)}, \hat{v}_j^{(1)}, \hat{z}_j^{(1)}$ which do not depend on k. Notably $v_j^{(1)}, \hat{z}_j^{(1)}$ are the same functions as in (3.2).

Step 1. We first estimate the evolution of the Lipschitz constant of $\hat{v}_i^{(1)}$ when i varies. Let $x' \in \mathbb{R}^d$. By the Cauchy-Schwarz inequality

$$\Delta t_k \mathbb{E}\Big|\overline{\widehat{Z}}_{k,1}^{k,x} - \overline{\widehat{Z}}_{k,1}^{k,x'}\Big|^2 \le \frac{1}{\Delta t_k}\mathbb{E}\Big|\mathbb{E}_k\Big[(\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'})\Delta W_k\Big]\Big|^2 \le d\ \mathbb{E}\Big|\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big|^2. \quad (4.19)$$

Moreover, assuming that $\widehat{\mathcal{U}}_k^{(1)}$ is $[\widehat{\mathcal{U}}_k^{(1)}]-$Lipschitz yields

$$\begin{aligned} \mathbb{E}\Big|\hat{V}_{k,1}^{k,x} - \hat{V}_{k,1}^{k,x'}\Big| &\le \mathbb{E}\Big|\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big| + \Delta t_i \mathbb{E}\Big|\{f\big(t_k, x', \widehat{\mathcal{U}}_k^{(1)}(x')\big) - f\big(t_k, x, \widehat{\mathcal{U}}_k^{(1)}(x)\big)\}\Big| \\ &\le \mathbb{E}\Big|\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big| + [f]\Delta t_i(1 + [\widehat{\mathcal{U}}_k^{(1)}])|x - x'|_2. \end{aligned}$$

Step 2. Then for the $v_k^{(1)}$ function, the Young inequality gives

$$\begin{aligned} \mathbb{E}\Big|V_{k,1}^{k,x} - V_{k,1}^{k,x'}\Big|^2 &\le (1 + \gamma\Delta t_k)\mathbb{E}\Big|\mathbb{E}_k\Big[\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big]\Big|^2 \\ &\quad + (1 + \frac{1}{\gamma\Delta t_k})\Delta t_k^2 \mathbb{E}\Big|\{f\big(t_k, x', V_{k,1}^{k,x'}\big) - f\big(t_k, x, V_{k,1}^{k,x}\big)\}\Big|^2 \\ &\le (1 + \gamma\Delta t_k)\mathbb{E}\Big|\mathbb{E}_k\Big[\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big]\Big|^2 \\ &\quad + 2[f]^2(1 + \frac{1}{\gamma\Delta t_k})\Delta t_k^2 \mathbb{E}\big[|x - x'|_2^2 + |V_{k,1}^{k,x} - V_{k,1}^{k,x'}|^2\big]. \end{aligned}$$

Therefore by choosing $\gamma = 2[f]^2$ for Δt_k small enough

$$\mathbb{E}\Big|V_{k,1}^{k,x} - V_{k,1}^{k,x'}\Big|^2 \le (1 + (\gamma+3)\Delta t_k)\mathbb{E}\Big|\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big|^2 + (1 + (\gamma+3)\Delta t_i)\Delta t_k|x - x'|_2^2.$$

Hence assuming $\hat{v}_{k+1}^{(1)}$ is $[\hat{v}_{k+1}^{(1)}]-$Lipchitz we obtain with Lemma 3.1

$$|v_k^{(1)}(x) - v_k^{(1)}(x')|^2 = \mathbb{E}\Big|V_{k,1}^{k,x} - V_{k,1}^{k,x'}\Big|^2 \tag{4.20}$$
$$\leq (1 + (\gamma+3)\Delta t_k)((1 + C\Delta t_k)[\hat{v}_{k+1}^{(1)}]^2 + \Delta t_k)|x - x'|_2^2$$
$$\leq (1 + \tilde{C}\Delta t_k)([\hat{v}_{k+1}^{(1)}]^2 + \Delta t_k)|x - x'|_2^2 := [v_k^{(1)}]^2|x - x'|_2^2, \tag{4.21}$$

for Δt_k small enough and another constant $\tilde{C}$.
Step 3. Let $\epsilon > 0$, $\kappa \in \mathbb{N}$, $\ell \in \mathbb{N}$, $m \in \mathbb{R}^\ell$ to be chosen after. Recursively, we choose $\mathcal{N}_k = \mathcal{G}^{\zeta_\kappa}_{[v_k^{(1)}],d,1,\ell,m}$ (with $[v_{N-1}^{(1)}]^2 = (1 + \tilde{C}\Delta t_{N-1})([g]^2 + \Delta t_{N-1})$ by (4.21)) to approximate $v_k^{(1)}$ by $[v_k^{(1)}]-$Lipschitz GroupSort neural networks with uniform error $2[v_k]R\epsilon$ on $[-R,R]^d$, see Proposition 2.1. Therefore, by Lemma 3.1, estimations (4.20) and the definition of $[v_k^{(1)}]$ in (4.21), for Δt_k small enough

$$|\hat{v}_k^{(1)}(x) - \hat{v}_k^{(1)}(x')| \leq \mathbb{E}\Big|\hat{V}_{k+1,1}^{k,x} - \hat{V}_{k+1,1}^{k,x'}\Big| + [f]\Delta t_k(1 + [\widehat{\mathcal{U}}_k^{(1)}])|x - x'|_2$$
$$\leq (1 + (C + 2[f])\Delta t_k)[\hat{v}_{k+1}^{(1)}]|x - x'|_2 + [f](1 + C\Delta t_k)\Delta t_k|x - x'|_2.$$

Thus $\hat{v}_k^{(1)}$ is $[\hat{v}_k^{(1)}]$ Lipschitz with

$$[\hat{v}_k^{(1)}] \leq (1 + \hat{C}\Delta t_k)[\hat{v}_{k+1}^{(1)}] + [f](1 + C\Delta t_k)\Delta t_i$$

for a constant $\hat{C}$. By discrete Gronwall lemma over $k = N-1, \dots, 0$,

$$[\hat{v}_i^{(1)}]^2 \leq K,\ [v_i^{(1)}]^2 \leq K,$$

uniformly in i, N for some constant K. By (4.19) and Proposition 2.1, we choose $\mathcal{N}_k' = \mathcal{G}^{\zeta_\kappa}_{\sqrt{\frac{d}{\Delta t_i}}[v_k^{(1)}],d,d,\ell,m}$ to approximate $\widehat{z_k}^{(1)}$ by GroupSort neural networks with uniform error $2\frac{d}{\sqrt{\Delta t_k}}[v_k]R\epsilon$ on $[-R,R]^d$. Thus $\sqrt{\Delta t_k}\widehat{z_k}^{(1)}, \sqrt{\Delta t_k}\mathcal{Z}_k^{(1)}$ are dK Lipschitz, uniformly.
Step 4. The regression errors $\varepsilon_i^{1,y}$ verify from, localization of X_i on $B_2(R)$, the Hölder inequality, and the Markov inequality, the approximation error of $v_i^{(1)}$, $i \in [\![0, N-1]\!]$, by the class of GroupSort neural networks (Proposition 2.1)

$$\begin{aligned}
\sqrt{\varepsilon_i^{1,y}} &= \inf_{\mathcal{U}\in\mathcal{G}_{[v_i],d,1}} \big\|v_i^{(1)}(X_i) - \mathcal{U}(X_i)\big\|_2 \\
&\leq \inf_{\mathcal{U}\in\mathcal{G}_{[v_i],d,1}} \Big\|\big(v_i^{(1)}(X_i) - \mathcal{U}(X_i)\big)\mathbf{1}_{X_i\in B_2(R)}\Big\|_2 + \Big\|\big(v_i^{(1)}(X_i) - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big)\mathbf{1}_{|X_i|_2\geq R}\Big\|_2 \\
&\leq 2KR\epsilon + \mathbb{E}\Big|\big(v_i^{(1)}(X_i) - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big)^{2q}\Big|^{1/2q}\mathbb{E}\Big|\mathbf{1}_{|X_i|_2\geq R}^{\frac{2q}{2q-1}}\Big|^{\frac{2q-1}{2q}} \\
&= 2KR\epsilon + \mathbb{E}\Big|\big(v_i^{(1)}(X_i) - \widehat{\mathcal{U}}_i^{(1)}(X_i)\big)^{2q}\Big|^{1/2q}\mathbb{E}[\mathbf{1}_{|X_i|_2\geq R}]^{\frac{2q-1}{2q}} \\
&\leq 2KR\epsilon + \frac{\big(\big\|v_i^{(1)}(X_i) - v_i^{(1)}(0)\big\|_{2q} + \big\|\widehat{\mathcal{U}}_i^{(1)}(X_i) - v_i^{(1)}(0)\big\|_{2q}\big)\big\|X_i\big\|_{\frac{2q}{2q-1}}}{R},
\end{aligned} \tag{4.22}$$

for $q > 1$ and $2q = 2 + \delta$ with δ as in the statement of the Proposition and by noticing that $(v_i^{(1)}(X_i) - \widehat{\mathcal{U}}_i^{(1)}(X_i)) = (v_i^{(1)}(X_i) - v_i^{(1)}(0) - (\widehat{\mathcal{U}}_i^{(1)}(X_i) - v_i^{(1)}(0)))$. Now, by Lipschitz continuity of $v_i^{(1)}$,$\widehat{\mathcal{U}}^{(1)}$ and because $0 \in B_2(R)$ we have

$$\begin{aligned}
&\big\|\widehat{\mathcal{U}}_i^{(1)}(X_i) - v_i^{(1)}(0)\big\|_{2q} + \big\|v_i^{(1)}(X_i) - v_i^{(1)}(0)\big\|_{2q} \\
&\leq \big\|\widehat{\mathcal{U}}_i^{(1)}(0) - v_i^{(1)}(0)\big\|_{2q} + \big\|\widehat{\mathcal{U}}_i^{(1)}(X_i) - \widehat{\mathcal{U}}_i^{(1)}(0)\big\|_{2q} + \big\|v_i^{(1)}(X_i) - v_i^{(1)}(0)\big\|_{2q} \\
&\leq 2KR\epsilon + 2K\big\|X_i\big\|_{2q}.
\end{aligned} \tag{4.23}$$

Recalling the standard estimate $\|X_i\|_{2q} \le C(1+\|\mathcal{X}_0\|_{2q})$, $i=0,\dots,N$, we then have

$$\varepsilon_i^{1,y} \le C\Big\{R^2\epsilon^2 + \frac{1+R^2\epsilon^2}{R^2}\Big\},$$

for some constant $C(d,\mathcal{X}_0)$ independent of N,R,ϵ. Similarly, repeating (4.22) and (4.23) by replacing respectively $\widehat{\mathcal{U}}_i^{(1)}$ by $\widehat{\mathcal{Z}}_i^{(1)}$ and $v_i^{(1)}$ by $\hat{z}_i^{(1)}$ and recalling that $\sqrt{\Delta t_k}\widehat{z}_k^{(1)}, \sqrt{\Delta t_k}\mathcal{Z}_k^{(1)}$ are dK Lipschitz uniformly w.r.t N, we obtain

$$\Delta t_i \varepsilon_i^{1,z} \le C\{R^2\epsilon^2 + \frac{1+R^2\epsilon^2}{R^2}\},$$

Then to obtain a convergence rate of $O(1/N)$ in (3.3), it suffices to choose R,ϵ such that

$$NR^2\epsilon^2 = O(1/N),\ N\frac{1+R^2\epsilon^2}{R^2} = O(1/N),$$

which is verified with if $d>1$ with $R=O(N)$, $\epsilon = O(\frac{1}{N^2})$. Then by Proposition 2.1, we can choose the previously GroupSort neural networks with grouping size $\kappa = O(2\sqrt{d}N^2)$, depth $\ell+1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((2\sqrt{d}N^2)^{d^2-1})$ if $d>1$. If $d=1$, we can take $\kappa = O(N^2)$, depth $\ell+1=3$ and width $\sum_{i=0}^{\ell-1} m_i = O(N^2)$.

4.3 Proof of Theorem 3.2

Let us introduce the *explicit* backward Euler scheme of the BSDE (2.3):

$$\left\{\begin{array}{rcl} \bar{V}_i^{(2)} & = & \mathbb{E}_i\Big[\bar{V}_{i+1}^{(2)} - f\big(t_i, X_i, \bar{V}_{i+1}^{(2)}, \bar{Z}_i^{(2)}\big)\Delta t_i\Big] \\ \bar{Z}_i^{(2)} & = & \mathbb{E}_i\Big[\bar{V}_{i+1}^{(2)}\frac{\Delta W_i}{\Delta t_i}\Big], \quad i=0,\dots,N-1, \end{array}\right. \tag{4.24}$$

starting from $\bar{V}_N^{(2)} = g(X_N)$, and which is also known to converge with the same time discretization error (4.2) than the implicit backward scheme.

We decompose the approximation error into three terms:

$$\mathbb{E}\big|Y_{t_i} - \widehat{\mathcal{U}}_i^{(2)}(X_i)\big|^2 \le 3\Big(\mathbb{E}\big|Y_{t_i} - \bar{V}_i^{(2)}\big|^2 + \mathbb{E}\big|\bar{V}_i^{(2)} - V_i^{(2)}\big|^2 + \mathbb{E}\big|V_i^{(2)} - \widehat{\mathcal{U}}_i^{(2)}(X_i)\big|^2\Big). \tag{4.25}$$

The first term is the classical time discretization error, and the rest of the proof is devoted to the analysis of the second and third terms.

Step 1. Fix $i \in [\![0, N-1]\!]$. By definition of $V_i^{(2)}$ in (3.4) and the martingale representation theorem, there exists a square integrable process $\{\widehat{Z}_s^{(2)}, t_i \le s \le t_{i+1}\}$ such that

$$\begin{aligned} &\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - f\big(t_i, X_i, \mathbb{E}_i[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})], \mathbb{E}_i\big[\sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big]\big)\Delta t_i \\ &= V_i + \int_{t_i}^{t_{i+1}} \widehat{Z}_s^{(2)}.\mathrm{d}W_s. \end{aligned}$$

It follows that the quadratic loss function of the DS scheme in (2.7) is written as

$$\begin{aligned} &J_i^S(\mathcal{U}_i) \\ &:= \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \mathcal{U}_i(X_i) - f\big(t_i, X_{i+1}, \widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}), \sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big)\Delta t_i\Big|^2 \\ &= \tilde{J}_i^S(\mathcal{U}_i) + \mathbb{E}\Big[\int_{t_i}^{t_{i+1}} |\widehat{Z}_s^{(2)}|_2^2 \mathrm{d}s\Big], \end{aligned} \tag{4.26}$$

where

$$\begin{aligned} \tilde{J}_i^S(\mathcal{U}_i) &:= \mathbb{E}\Big|V_i^{(2)} - \mathcal{U}_i(X_i) + \Delta f_i \Delta t_i\Big|^2 \\ \text{with}\quad \Delta f_i &:= f\big(t_i, X_i, \mathbb{E}_i[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})], \mathbb{E}_i[\sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})]\big) \\ &\quad - f\big(t_i, X_{i+1}, \widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}), \sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big). \end{aligned}$$

A direct application of the Young inequality in the form $(a+b)^2 \geq \frac{1}{2}a^2 - b^2$ leads to

$$\tilde{J}_i^S(\mathcal{U}_i) + |\Delta t_i|^2 \mathbb{E}\big|\Delta f_i\big|^2 \geq \frac{1}{2}\mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i(X_i)\big|^2. \tag{4.27}$$

On the other hand, by Lipschitz continuity of f, we have

$$\begin{aligned}
&\tilde{J}_i^S(\mathcal{U}_i) + |\Delta t_i|^2 \mathbb{E}\big|\Delta f_i\big|^2 \\
&\leq\ 2\mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i(X_i)\big|^2 + 3|\Delta t_i|^2 \mathbb{E}\big|\Delta f_i\big|^2 \\
&\leq\ 2\mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i(X_i)\big|^2 + 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}|X_{i+1} - X_i|_2^2 \\
&\quad +\ 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \mathbb{E}_i[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})]\Big|^2 \\
&\quad +\ 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}\Big|\sigma(t_i, X_i)^\intercal D_x \widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \mathbb{E}_i\big[\sigma(t_i, X_i)^\intercal D_x \widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big]\Big|_2^2 \\
&\leq\ 2\ \mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i^{(2)}(X_i)\big|^2 + 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}|X_{i+1} - X_i|_2^2 \\
&\quad +\ 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \widehat{\mathcal{U}}_{i+1}^{(2)}(X_i)\Big|^2 \\
&\quad +\ 9|\Delta t_i|^2 [f]_L^2 \mathbb{E}\Big[|\sigma(t_i, X_i)|_2^2 \mathbb{E}_i\big|D_x \widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - D_x \widehat{\mathcal{U}}_{i+1}^{(2)}(X_i)\big|_2^2\Big],
\end{aligned} \tag{4.28}$$

where we use the definition of conditional expectation $\mathbb{E}_i[.]$, and the tower property of conditional expectation in the last inequality. Recall that $\widehat{\mathcal{U}}_{i+1} \in \mathcal{N}_i^{\gamma,\eta}$ is Lipschitz on $\mathbb{R}^d$. Actually, we have

$$\big|\widehat{\mathcal{U}}_{i+1}(x) - \widehat{\mathcal{U}}_{i+1}(x')\big| \leq\ \gamma_i |x - x'|_2, \quad \forall x, x' \in \mathbb{R}^d\,.$$

By the Cauchy-Schwarz inequality, we then have

$$\begin{aligned}
\mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \widehat{\mathcal{U}}_{i+1}^{(2)}(X_i)\Big|^2 &\leq\ C\gamma_i^2 \big\|X_{i+1} - X_i\big\|_4^2 \\
&\leq\ C\gamma_i^2 \Delta t_i
\end{aligned}$$

for Δt_i small enough, $R \geq 1$, and we used again the standard estimate: $\|X_i\|_{2p} \leq C(1 + \|\mathcal{X}_0\|_{2p})$, $\|X_{i+1} - X_i\|_{2p} \leq C(1 + \|\mathcal{X}_0\|_{2p})\sqrt{\Delta t_i}$, for $p \geq 1$. By using also the Lipschitz condition on $D_x\widehat{\mathcal{U}}_{i+1}$, and plugging into (4.28), we get

$$\tilde{J}_i^S(\mathcal{U}_i) + |\Delta t_i|^2 \mathbb{E}\big|\Delta f_i\big|^2 \leq\ 2\mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i(X_i)\big|^2 + C(d)\max\big[\gamma_i^2, \eta_i^2\big]\Big(1 + \big\|\mathcal{X}_0\big\|_4^2\Big)^2 |\Delta t_i|^3. \tag{4.29}$$

By applying inequality (4.27) to $\mathcal{U}_i = \widehat{\mathcal{U}}_i^{(2)}$, which is a minimizer of $\tilde{\mathcal{J}}_i^S$ by (4.26), and combining with (4.29), this yields for all functions $\mathcal{U}_i$ in $\mathcal{N}_i^{\gamma,\eta}$:

$$\mathbb{E}\big|V_i^{(2)} - \widehat{\mathcal{U}}_i^{(2)}(X_i)\big|^2 \leq\ C\Big(\mathbb{E}\big|V_i^{(2)} - \mathcal{U}_i(X_i)\big|^2 + (1 + \|\mathcal{X}_0\|_4^2)^2 |\Delta t_i|^3 \max\big[\gamma_i^2, \eta_i^2\big]\Big),$$

and thus by minimizing over $\mathcal{U}_i$ in the right hand side

$$\mathbb{E}\big|V_i^{(2)} - \widehat{\mathcal{U}}_i^{(2)}(X_i)\big|^2 \leq\ C\Big(\varepsilon_i^{\gamma,\eta} + (1 + \|\mathcal{X}_0\|_4^2)^2 |\Delta t_i|^3 \max\big[\gamma_i^2, \eta_i^2\big]\Big). \tag{4.30}$$

Step 2.

From the expressions of $V_i^{(2)}$, and $\bar{V}_i^{(2)}$ in (3.4) and (4.24), and by applying the Young, the Cauchy-Schwarz inequalities, we get with $\beta \in (0,1)$

$$
\begin{aligned}
&\mathbb{E}\big|\bar V_i^{(2)} - V_i^{(2)}\big|^2 \\
&\le\ (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big]\Big|^2 \\
&+\ \big(1+\frac{1}{\beta}\big)|\Delta t_i|^2\mathbb{E}\Big|f\big(t_i, X_i, \bar V_{i+1}^{(2)}, \bar Z_i^{(2)}\big) \\
&\qquad\qquad -\ f\big(t_i, X_i, \mathbb{E}_i[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})], \mathbb{E}_i[\sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})]\big)\Big|^2 \\
&\le\ (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big]\Big|^2 \\
&+\ 2[f]_{\scriptscriptstyle L}^2\big(1+\frac{1}{\beta}\big)|\Delta t_i|^2\Big(\mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big|^2 + \mathbb{E}\Big|\mathbb{E}_i\big[\sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big] - \bar Z_i^{(2)}\Big|_2^2\Big) \qquad (4.31)
\end{aligned}
$$

Now, recalling the expression of $\bar Z_i$ in (4.24), and by a standard integration by parts argument (see e.g. Lemma 2.1 in [FTW11]), we have

$$
\begin{aligned}
&\mathbb{E}_i\big[\sigma(t_i,X_i)^\intercal D_x\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1})\big] - \bar Z_i^{(2)} \\
&=\ \mathbb{E}_i\Big[\big(\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big)\frac{\Delta W_i}{\Delta t_i}\Big] \\
&=\ \mathbb{E}_i\Big[\Big(\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)} - \mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big]\Big)\frac{\Delta W_i}{\Delta t_i}\Big].
\end{aligned}
$$

By plugging into (4.31), we then obtain by the Cauchy-Schwarz inequality

$$
\begin{aligned}
&\mathbb{E}\big|\bar V_i^{(2)} - V_i^{(2)}\big|^2 \\
&\le (1+\beta)\mathbb{E}\Big|\mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big]\Big|^2 + 2[f]_{\scriptscriptstyle L}^2(1+\beta)\frac{|\Delta t_i|^2}{\beta}\Big\{\mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big|^2 \\
&+\ \frac{d}{\Delta t_i}\Big[\mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big|^2 - \mathbb{E}\Big|\mathbb{E}_i\big[\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big]\Big|^2\Big]\Big\} \\
&\le\ (1+C\Delta t_i)\mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - \bar V_{i+1}^{(2)}\big|^2, \qquad (4.32)
\end{aligned}
$$

by choosing explicitly $\beta = 2d[f]_{\scriptscriptstyle L}^2\Delta t_i$ for Δt_i small enough. By using again the Young inequality on the r.h.s. of (4.32), and since $\Delta t_i = O(1/N)$, we then get

$$
\mathbb{E}\big|\bar V_i^{(2)} - V_i^{(2)}\big|^2 \le\ (1+C\Delta t_i)\mathbb{E}\big|\bar V_{i+1}^{(2)} - V_{i+1}^{(2)}\big|^2 + CN\mathbb{E}\big|\widehat{\mathcal{U}}_{i+1}^{(2)}(X_{i+1}) - V_{i+1}^{(2)}\big|^2.
$$

By discrete Gronwall lemma, and recalling that $\bar V_N^{(2)} = g(X_N)$, $V_N^{(2)} = \widehat{\mathcal{U}}_N(X_N)$, we deduce with (4.30) that

$$
\sup_{i\in[\![0,N]\!]}\mathbb{E}\big|\bar V_i^{(2)} - V_i^{(2)}\big|^2 \le\ C\varepsilon_N^{\gamma,\eta} + CN\sum_{i=1}^{N-1}\Big(\varepsilon_i^{\gamma,\eta} + (1+\|\mathcal{X}_0\|_4^2)^2|\Delta t_i|^3\max\big[\gamma_i^2,\eta_i^2\big]\Big). \quad (4.33)
$$

The required bound (3.5) for the approximation error on Y follows by plugging (4.2), (4.30) and (4.33) into (4.25). □

4.4 Proof of Proposition 3.2

For $x\in\mathbb{R}^d$, we define the processes $X_{j+1}^{j,x}$, $j=0,\dots,N$,

$$
X_{j+1}^{j,x} := x + \mu(t_j,x)\Delta t_j + \sigma(t_j,x)\Delta W_j, \quad j=0,\dots,N-1.
$$

Define also

$$
\left\{\begin{array}{rcl} V_{i,3}^x &=& \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - f\big(t_i,x,V_{i,3}^x,\overline{\widehat{Z}}_{i,3}^x\big)\Delta t_i\Big] = v_i^{(3)}(x) \\ \overline{\widehat{Z}}_{i,3}^x &=& \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x})\frac{\Delta W_i}{\Delta t_i}\Big] = \widehat{z}_i^{(3)}(x)\end{array}\right.
$$

with $v_i^{(3)}, \widehat{z}_i^{(3)}$ as in (3.6) by Markov property.
Step 1. Let $x' \in \mathbb{R}^d$. By the Cauchy-Schwarz inequality, we have the standard estimate

$$\begin{aligned}
&\Delta t_i \mathbb{E}\Big|\overline{\widehat{Z}}_{i,3}^{x} - \overline{\widehat{Z}}_{i,3}^{x'}\Big|_2^2 \\
&= \frac{1}{\Delta t_i}\mathbb{E}\Big|\mathbb{E}_i\Big[\{\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'}) - \mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big]\}\Delta W_i\Big]\Big|^2 \\
&\leq d\,\Big(\mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big|^2 - \mathbb{E}\Big|\mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big]\Big|^2\Big). \qquad (4.34)
\end{aligned}$$

We then apply the Young inequality to see that

$$\begin{aligned}
&\mathbb{E}\Big|V_{i,3}^{x} - V_{i,3}^{x'}\Big|^2 \\
&\leq (1+\gamma\Delta t_i)\mathbb{E}\Big|\mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big]\Big|^2 \\
&+ (1+\frac{1}{\gamma\Delta t_i})\Delta t_i^2\mathbb{E}\Big|\{f\big(t_i, x', V_{i,3}^{x'}, \overline{\widehat{Z}}_i^{3,x'}\big) - f\big(t_i, x, V_{i,3}^{x}, \overline{\widehat{Z}}_{i,3}^{x}\big)\}\Big|^2 \\
&\leq (1+\gamma\Delta t_i)\mathbb{E}\Big|\mathbb{E}_i\Big[\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big]\Big|^2 \\
&+ 3[f]^2(1+\frac{1}{\gamma\Delta t_i})\Delta t_i^2\mathbb{E}\{|x-x'|_2^2 + |V_{i,3}^{x} - V_{i,3}^{x'}|^2 + |\overline{\widehat{Z}}_{i,3}^{x} - \overline{\widehat{Z}}_{i,3}^{x'}|_2^2\}.
\end{aligned}$$

Hence for $\gamma = 3[f]^2 d$ and Δt_i small enough, using (4.34) we obtain

$$\begin{aligned}
\mathbb{E}\Big|V_{i,3}^{x} - V_{i,3}^{x'}\Big|^2 &\leq (1+(\gamma+3d)\Delta t_i)\mathbb{E}\Big|\widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x}) - \widehat{\mathcal{U}}_{i+1}^{(3)}(X_{i+1}^{i,x'})\Big|^2 \\
&\quad + (1+(\gamma+3d)\Delta t_i)\Delta t_i\mathbb{E}|x-x'|_2^2.
\end{aligned}$$

Therefore, with Lemma 3.1

$$\begin{aligned}
|v_{N-1}^{(3)}(x) - v_{N-1}^{(3)}(x')|^2 &= \mathbb{E}\Big|V_{N-1,3}^{x} - V_{N-1,3}^{x'}\Big|^2 \\
&\leq (1+(\gamma+3d)\Delta t_{N-1})((1+C\Delta t_{N-1})[g]^2 + \Delta t_i)|x-x'|_2^2 \\
&\leq (1+\hat{C}\Delta t_{N-1})([g]^2 + \Delta t_i)|x-x'|_2^2,
\end{aligned}$$

for some constant $\hat{C}$. Similarly, assuming $\widehat{\mathcal{U}}_{i+1}^{(3)}$ is $[\widehat{\mathcal{U}}_{i+1}^{(3)}]-$Lipschitz, $v_i^{(3)}$ is Lipschitz with constant $[v_i^{(3)}]$ verifying

$$[v_i^{(3)}]^2 \leq (1+\hat{C}\Delta t_i)([\widehat{\mathcal{U}}_{i+1}^{(3)}]^2 + \Delta t_i).$$

Step 2. Let $\epsilon > 0,\ \kappa \in \mathbb{N},\ \ell \in \mathbb{N},\ m \in \mathbb{R}^\ell$ to be chosen after. Recursively, we approximate $v_i^{(3)}$ by a $[v_i^{(3)}]$-Lipschitz GroupSort neural network $\mathcal{U}_i^{(3)}$ in $\mathcal{N}_i = \mathcal{G}^{\zeta_\kappa}_{[v_i^{(3)}],d,1,\ell,m}$ with uniform error $2[v_i]R\epsilon$ on $[-R,R]^d$ (Proposition 2.1). Then by discrete Gronwall inequality

$$[\mathcal{U}_i^{(3)}]^2 \leq K,\ [v_i^{(3)}]^2 \leq K,$$

uniformly in i, N for some constant K. Thus $v_i^{(3)}$,$\mathcal{U}_i^{(3)}$ are K Lipschitz, uniformly. Then we approximate by (4.34) $\widehat{z}_i^{(3)}$ by a $\sqrt{\frac{d}{\Delta t_i}}[v_i^{(3)}]$-Lipschitz GroupSort neural network $\mathcal{Z}_i$ in $\mathcal{N}_i' = \mathcal{G}^{\zeta_\kappa}_{\sqrt{\frac{d}{\Delta t_i}}[v_i^{(3)}],d,d,\ell,m}$ with uniform error $2\frac{d}{\sqrt{\Delta t_i}}[v_i^{(3)}]R\varepsilon$ on $[-R,R]^d$ thanks to Proposition 2.1. Thus $\sqrt{\Delta t_i}\widehat{z}_i^{(3)}, \sqrt{\Delta t_i}\mathcal{Z}_i^{(3)}$ are dK Lipschitz, uniformly.
Step 3. The regression errors $\varepsilon_i^{3,y}$ verify from, localization of X_i on $B_2(R)$, the Hölder inequality, and the Markov inequality, the approximation error of $v_i^{(3)}$, $i \in [\![0, N-1]\!]$, by the class of GroupSort

neural networks (Proposition 2.1)

$$\begin{aligned}
\sqrt{\varepsilon_i^{3,y}} &= \inf_{\mathcal{U}\in\mathcal{G}_{[v_k],d,1}} \big\|v_i^{(3)}(X_i) - \mathcal{U}(X_i)\big\|_2 \\
&\le \inf_{\mathcal{U}\in\mathcal{G}_{[v_k],d,1}} \Big\|\big(v_i^{(3)}(X_i) - \mathcal{U}(X_i)\big)\mathbf{1}_{X_i\in B_2(R)}\Big\|_2 + \Big\|\big(v_i^{(3)}(X_i) - \widehat{\mathcal{U}}_i^{(3)}(X_i)\big)\mathbf{1}_{|X_i|_2\ge R}\Big\|_2 \\
&\le 2KR\epsilon + \mathbb{E}\Big|\big(v_i^{(3)}(X_i) - \widehat{\mathcal{U}}_i^{(3)}(X_i)\big)^{2q}\Big|^{1/2q}\mathbb{E}\Big|\mathbf{1}_{|X_i|_2\ge R}^{\frac{2q}{2q-1}}\Big|^{\frac{2q-1}{2q}} \\
&= 2KR\epsilon + \mathbb{E}\Big|\big(v_i^{(3)}(X_i) - \widehat{\mathcal{U}}_i^{(3)}(X_i)\big)^{2q}\Big|^{1/2q}\mathbb{E}[\mathbf{1}_{|X_i|_2\ge R}]^{\frac{2q-1}{2q}} \\
&\le 2KR\epsilon + \frac{\big(\big\|v_i^{(3)}(X_i) - v_i^{(3)}(0)\big\|_{2q} + \big\|\widehat{\mathcal{U}}_i^{(3)}(X_i) - v_i^{(3)}(0)\big\|_{2q}\big)\big\|X_i\big\|_{\frac{2q}{2q-1}}}{R}, \qquad (4.35)
\end{aligned}$$

by noticing that $(v_i^{(3)}(X_i) - \widehat{\mathcal{U}}_i^{(3)}(X_i)) = (v_i^{(3)}(X_i) - v_i^{(3)}(0) - (\widehat{\mathcal{U}}_i^{(3)}(X_i) - v_i^{(3)}(0)))$ for $q > 0$ and $2q = 2+\delta$ with δ as in the statement of the Proposition. Now, by Lipschitz continuity of $v_i^{(3)}$,$\widehat{\mathcal{U}}^{(3)}$ and because $0 \in B_2(R)$ we have

$$\begin{aligned}
&\big\|\widehat{\mathcal{U}}_i^{(3)}(X_i) - v_i^{(3)}(0)\big\|_{2q} + \big\|v_i^{(3)}(X_i) - v_i^{(3)}(0)\big\|_{2q} \\
&\le \big\|\widehat{\mathcal{U}}_i^{(3)}(0) - v_i^{(3)}(0)\big\|_{2q} + \big\|\widehat{\mathcal{U}}_i^{(3)}(X_i) - \widehat{\mathcal{U}}_i^{(3)}(0)\big\|_{2q} + \big\|v_i^{(3)}(X_i) - v_i^{(3)}(0)\big\|_{2q} \\
&\le 2KR\epsilon + 2K\big\|X_i\big\|_{2q}. \qquad (4.36)
\end{aligned}$$

Recalling the standard estimate $\|X_i\|_{2q} \le C(1 + \|\mathcal{X}_0\|_{2q})$, $i = 0,\dots,N$, we then have

$$\varepsilon_i^{3,y} \le C\Big\{R^2\epsilon^2 + \frac{1+R^2\epsilon^2}{R^2}\Big\},$$

for some constant $C(d,\mathcal{X}_0)$ independent of N, R, ϵ. Similarly repeating (4.35) and (4.36) by replacing respectively $\widehat{\mathcal{U}}_i^{(3)}$ by $\widehat{\mathcal{Z}}_i^{(3)}$ and $v_i^{(3)}$ by $\hat{z}_i^{(3)}$ and recalling that $\sqrt{\Delta t_i}\widehat{z}_i{}^{(3)}, \sqrt{\Delta t_i}\mathcal{Z}_i^{(3)}$ are dK Lipschitz uniformly w.r.t. N, we obtain

$$\Delta t_i\varepsilon_i^{3,z} \le C\Big\{R^2\epsilon^2 + \frac{1+R^2\epsilon^2}{R^2}\Big\}.$$

Then to obtain a convergence rate of $O(1/N)$ in (3.7), it suffices to choose R, ε such that

$$N^2R^2\epsilon^2 = O(1/N), \quad N^2\frac{1+R^2\epsilon^2}{R^2} = O(1/N),$$

which is verified with $R = O(N^{3/2})$, $\epsilon = O(\frac{1}{N^3})$. Then by Proposition 2.1, if $d > 1$ we can choose the previously GroupSort neural networks with grouping size $\kappa = O(\lceil 2\sqrt{d}N^3\rceil)$, depth $\ell + 1 = O(d^2)$ and width $\sum_{i=0}^{\ell-1} m_i = O((2\sqrt{d}N^3)^{d^2-1})$. If $d = 1$, we can take $\kappa = O(N^3)$, depth $\ell + 1 = 3$ and width $\sum_{i=0}^{\ell-1} m_i = O(N^3)$.

5 Numerical Tests

We test our different algorithms and the cited ones in this paper on some examples and by varying the state space dimension. In each example we use tanh as activation function, and an architecture composed of 2 hidden layers with $d + 10$ neurons. We apply Adam gradient descent [KB14] with a decreasing learning rate, using the Tensorflow library. Each numerical experiment is conducted using a node composed of 2 Intel® Xeon® Gold 5122 Processors, 192 Gb of RAM, and 2 GPU nVidia® Tesla® V100 16Gb. We use a batch size of 1000. We do not implement the GroupSort network because even if it is useful for theoretical analysis, it would be costly to use in practice: on the one hand, it will induce a cost of order $O(n \ln n)$ where n is the batch size, compared to a linear cost $O(n)$ for standard activation function; on the other hand, it requires to track the Lipschitz constant of the functions and adapt the networks architecture accordingly. Whereas theoretical

results suggest to take deep neural networks with depth increasing with the dimension, we observe that two hidden layers are enough to obtain a good accuracy. According to our experience tanh activation function provides the best results. ReLU or Elu being not bounded, some explosion tends to appear when the learning rates are not small enough.
We consider examples from [HPW20] to compare its DBDP scheme with the DS and MDBDP schemes. The three first lines of the tables below are taken from [HPW20]. For each test, the two best results are highlighted in boldface. We use 5000 gradient descent iterations by time step except 20000 for the projection of the final condition. The execution of the multistep algorithm approximately takes between 8000 s. and 16000 s. (depending on the dimension) for a resolution with $N = 120$. More numerical examples and tests are presented in the extended version [GPW20] of this paper, and the codes at: `https://github.com/MaxGermain/MultistepBSDE`.

5.1 PDE with Bounded Solution and Simple Structure

We take the parameters: $\mu = \frac{0.2}{d}$, $\sigma = \frac{I_d}{\sqrt{d}}$, terminal condition $g(x) = \cos(\overline{x})$, with $\overline{x} = \sum_{i=1}^{d} x_i$, and generator

$$\begin{aligned} & f(x,y,z) \\ = \quad & -\Big(\cos(\overline{x}) + 0.2\sin(\overline{x})\Big)e^{\frac{T-t}{2}} + \frac{1}{2}(\sin(\overline{x})\cos(\overline{x})e^{T-t})^2 - \frac{1}{2d}(y(1_d \cdot z))^2. \end{aligned}$$

so that the PDE solution is given by $u(t,x) = \cos(\overline{x})\exp\left(\frac{T-t}{2}\right)$.

We fix $T = 1$, and increase the dimension d. The results are reported in Figure 2 for $d = 10$, in Figure 3 for $d = 20$, and in Figure 4 for $d = 50$. It is observed that all the schemes DBDP, DBSDE and MDBDP provide quite accurate results with smallest standard deviation for MDBDP, and largely outperforms the DS scheme.

	Averaged value	Standard deviation	Relative error (%)
[HPW20] (DBDP1)	- 1.3895	0.0015	0.44
[HPW20] (DBDP2)	- 1.3913	**0.0006**	0.57
[HJE17] (DBSDE)	**- 1.3880**	0.0016	**0.33**
[Bec+19] (DS)	- 1.4097	0.0173	1.90
MDBDP	**-1.3887**	**0.0006**	**0.38**

Figure 2: Estimate of $u(0, x_0)$ in the case (5.1), where $d = 10, x_0 = 1\ \mathbb{1}_{10}, T = 1$ with 120 time steps. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is -1.383395.

	Averaged value	Standard deviation	Relative error (%)
[HPW20] (DBDP1)	0.6760	0.0027	0.47
[HPW20] (DBDP2)	**0.6710**	0.0056	**0.27**
[HJE17] (DBSDE)	0.6869	**0.0024**	2.09
[Bec+19] (DS)	0.6944	0.0201	3.21
MDBDP	**0.6744**	**0.0005**	**0.24**

Figure 3: Estimate of $u(0, x_0)$ in the case (5.1), where $d = 20, x_0 = 1\ \mathbb{1}_{20}, T = 1$ with 120 time steps. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 0.6728135.

	Averaged value	Standard deviation	Relative error (%)
[HPW20] (DBDP1)	**1.5903**	**0.0063**	**0.04**
[HPW20] (DBDP2)	1.5876	0.0068	0.21
[HJE17] (DBSDE)	1.5830	0.0361	0.50
[Bec+19] (DS)	1.6485	0.0140	3.62
MDBDP	**1.5924**	**0.0005**	**0.09**

Figure 4: Estimate of $u(0, x_0)$ in the case (5.1), where $d = 50, x_0 = 1\ \mathbb{1}_{50}, T = 1$ with 120 time steps. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.5909.

5.2 PDE with Unbounded Solution and more Complex Structure

We consider a toy example with solution given by

$$u(t,x) = \frac{T-t}{d}\sum_{i=1}^{d}(\sin(x_i)1_{x_i<0} + x_i 1_{x_i\geq 0}) + \cos\Big(\sum_{i=1}^{d} i x_i\Big).$$

Therefore we take the parameters

$$\mu = 0,\ \sigma = \frac{I_d}{\sqrt{d}},\ \ T = 1, \quad f(t,x,y,z) = k(t,x) - \frac{y}{\sqrt{d}}(1_d \cdot z) - \frac{y^2}{2} \tag{5.2}$$

with $k(t,x) = \partial_t u + \frac{1}{2d}\operatorname{Tr}(D_x^2 u) + \frac{u}{\sqrt{d}}\sum_i D_{x_i} u + \frac{u^2}{2}$.

We start with tests in dimension $d = 1$. The results are reported in Figure 5.

	Averaged value	Standard deviation	Relative error (%)
[HPW20] (DBDP1)	1.3720	0.0030	0.41
[HPW20] (DBDP2)	**1.3736**	0.0022	**0.29**
[HJE17] (DBSDE)	1.3724	**0.0005**	0.38
[Bec+19] (DS)	1.3630	0.0079	1.06
MDBDP	**1.3735**	**0.0003**	**0.30**

Figure 5: Estimate of $u(0, x_0)$ in the case (5.2), where $d = 1, x_0 = 0.5, T = 1$ with 120 time steps. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.3776.

We next increase the dimension to $d = 8$, and report the results in the following figure. The accuracy is not so good as in the previous section with simple structure of the solution, but we notice that the MDBDP scheme yields the best performance (above dimension $d = 10$, all the schemes do not give good approximation results).

	Averaged value	Standard deviation	Relative error (%)
[HPW20] (DBDP1)	**1.1694**	0.0254	**0.78**
[HPW20] (DBDP2)	1.0758	**0.0078**	7.28
[HJE17] (DBSDE)	NC	NC	NC
[Bec+19] (DS)	1.2283	**0.0113**	5.86
MDBDP	**1.1654**	0.0379	**0.47**

Figure 6: Estimate of $u(0, x_0)$ in the case (5.2), where $d = 8, x_0 = 0.5\ \mathbb{1}_8, T = 1$ with 120 time steps. Average and standard deviation observed over 10 independent runs are reported. The theoretical solution is 1.1603.

References

[ALG19] C. Anil, J. Lucas, and R. Grosse. "Sorting Out Lipschitz Function Approximation". In: *Proceedings of the 36th ICML*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. 2019, pp. 291–301.

[Bac17] F. Bach. "Breaking the Curse of Dimensionality with Convex Neural Networks". In: *Journal of Machine Learning Research* 18.19 (2017), pp. 1–53.

[BD07] C. Bender and R. Denk. "A forward scheme for backward SDEs". In: *Stochastic Processes and their Applications* 117.12 (2007), pp. 1793 –1812.

[Bec+19] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. "Deep splitting method for parabolic PDEs". In: *arXiv:1907.03452* (July 2019).

[BF11] B. Bercu and J.C. Fort. "Generic stochastic gradient methods". In: *Wiley Encyclopedia of Opera- tions Research and Management Science*. 2011, pp. 1–8.

[BGS15] Gabor Balazs, András György, and Csaba Szepesvari. "Near-optimal max-affine estimators for convex regression". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Guy Lebanon and S. V. N. Vishwanathan. Vol. 38. 2015, pp. 56–64.

[BJK19] C. Beck, A. Jentzen, and B. Kuckuck. "Full error analysis for the training of deep neural networks". In: *arXiv:1910.00121v2* (2019).

[BM] F. Bach and E. Moulines. "Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n)." In: *Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13*, pp. 773–781.

[BT04] B. Bouchard and N. Touzi. "Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations". In: *Stoch. Proc. Appl.* 111.2 (2004), pp. 175 –206.

[E+18] W. E, M. Hutzenthaler, A. Jentzen, and T. Kruse. "On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations". In: *to appear in Journal of Scientific Computing* (2018).

[FTW11] A. Fahim, N. Touzi, and X. Warin. "A probabilistic numerical method for fully nonlinear parabolic PDEs". In: *Ann. Appl. Probab.* 21.4 (Aug. 2011), pp. 1322–1364.

[GLW05] E. Gobet, J-P. Lemor, and X. Warin. "A regression-based Monte Carlo method to solve backward stochastic differential equations". In: *Ann. Appl. Probab.* 15.3 (2005), pp. 2172–2202.

[GPW20] M. Germain, H. Pham, and X. Warin. "Deep backward multistep schemes for nonlinear PDEs and approximation error analysis". In: *arXiv:2006.01496v1* (2020).

[GT16] E. Gobet and P. Turkedjiev. "Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions". In: *Math. Comp.* 85 (Mar. 2016).

[Gy02] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Series in Statistics, Springer-Verlag, 2002.

[HJE17] J. Han, A. Jentzen, and W. E. "Solving high-dimensional partial differential equations using deep learning". In: *Proc. Natl. Acad. Sci.* 115 (July 2017).

[HL+19] P. Henry-Labordère, N. Oudjane, X. Tan, N. Touzi, and X. Warin. "Branching diffusion representation of semilinear PDEs and Monte Carlo approximation". In: *Ann. Inst. H. Poincaré Probab. Statist.* 55.1 (Feb. 2019), pp. 184–210.

[HL20] J. Han and J. Long. "Convergence of the Deep BSDE Method for Coupled FBSDEs". In: *Probability, Uncertainty and Quantitative Risk* 5.1 (2020), pp. 1–33.

[HPW20] C. Huré, H. Pham, and X. Warin. "Deep backward schemes for high-dimensional nonlinear PDEs". In: *Mathematics of Computation* 89.324 (July 2020), pp. 1547–1580.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Netw.* 2.5 (July 1989), pp. 359–366. ISSN: 0893-6080.

[Hur+21] C. Huré, H. Pham, A. Bachouch, and N. Langrené. "Deep neural networks algorithms for stochastic control problems on finite horizon, part I: convergence analysis". In: *SIAM Journal on Numerical Analysis* 59.1 (2021), pp. 525–557.

[KB14] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference for Learning Representations*. 2014.

[Pin99] Allan Pinkus. "Approximation theory of the MLP model". In: *Acta Numerica* 8 (1999), pp. 143–195.

[PP90] E. Pardoux and S. Peng. "Adapted solution of a backward stochastic differential equation". In: *Systems & Control Letters* 14.1 (1990), pp. 55 –61. ISSN: 0167-6911.

[SS17] J. Sirignano and K. Spiliopoulos. "DGM: A deep learning algorithm for solving partial differential equations". In: *J. Computational Phys.* 375 (Aug. 2017).

[TSB21] U. Tanielian, M. Sangnier, and G. Biau. "Approximating Lipschitz continuous functions with GroupSort neural networks". In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS). PMLR: Volume 130, 2021.

[Yar17] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

[Zha04] J. Zhang. "A numerical scheme for BSDEs". In: *Ann. Appl. Probab.* 14.1 (2004), pp. 459–488.

[Zha17] J. Zhang. *Backward stochastic differential equations: from linear to fully nonlinear theory*. Vol. 86. Probability theory and stochastic modelling. Springer, 2017.