Discrete and Continuous Dynamical Systems - Series B
Vol. 30, No. 4, April 2025, pp. 1083-1105
doi:10.3934/dcdsb.2024121

SOLVING HIGH DIMENSIONAL FBSDE WITH DEEP SIGNATURE TECHNIQUES WITH APPLICATION TO NONLINEAR OPTIONS PRICING

Hui Sun[✉*1] and Feng Bao[✉2]

[1]Citibank, Markets and Global Function, Wilmington DE 19801, USA

[2]Department of Mathematics, Florida State University, FL 32306, USA

(Communicated by Xiaoying Han)

Abstract. We report two methods for solving FBSDEs of path-dependent types in high-dimensions. Inspired by the work from [31], [32], and [20], we propose deep learning frameworks for solving such problems using path signatures as underlying features. Our two methods (forward/backward) demonstrate comparable/better accuracy and efficiency compared to the state of the art [14], [13], and [18]. More importantly, leveraging the techniques developed in [5], we are able to solve the problem of high dimension (100 and above), which is a limitation in [14] and [13]. We also provide convergence proofs for both methods with the proof of the the forward method following similar lines to [14], and the backward methods inspired by [16] in the Markovian case.

1. **Introduction.** Ever since the seminal paper [32], solving high dimensional PDEs attracted a lot of attention, and researchers invested a large amount of effort in designing numerical schemes to solve such problems (c.f.[2], [17] and [18]). The workhorse of almost all such schemes is the combination of the usage of BSDEs/FBSDEs and deep neural networks. The BSDEs/FBSDEs framework provides a probabilistic interpretation for the PDEs of interest, which makes the simulation method possible, while deep learning techniques provide the uniform approximation power, which most importantly is not sensitive to dimensions.

The classical decoupled forward-backward SDE takes the form

$$\begin{cases} dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \\ dY_t = -f(t, X_t, Y_t, Z_t)dt + Z_t dW_t \\ X_0 = x, Y_T = g(X_T) \end{cases} \tag{1}$$

where $(b, \sigma) : [0, T] \times \mathbb{R}^{d_1} \to \mathbb{R}^{d_1} \times \mathbb{R}^{d_1 \times d}$, f is the driver of the BSDE $f : [0, T] \times \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \mathbb{R}^{d_2 \times d} \to \mathbb{R}^{d_2}$, and $g : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$. We have a Markovian system when the drift and diffusion of the SDE, the driver of the BSDE, and the terminal

2020 *Mathematics Subject Classification.* Primary: 65C30, 60H35; Secondary: 65M75.

Key words and phrases. Forward-backward stochastic differential equations, signature, Dimension reduction, option pricing.

Disclaimer: the content of the paper is of the first author's own research interest and does not represent any of the corporate opinion.

*Corresponding author: Hui Sun.

function are only dependent on the current state X_t. By the non-linear Feynman-Kac formula, the corresponding PDEs are of the semi-parabolic type.

On the other hand, of particular interest is the PDE of path dependent types (PPDE). The notion of nonlinear path-dependent PDEs was first proposed by Peng [24], and the classical solution of the semi-linear PDEs are given in [25]. The notion of viscosity solutions to the PPDEs were studied in [9], and the viscosity solutions of fully nonlinear parabolic path dependent PDEs are studied in [11] and [12]. This notion of viscosity solutions generalize that of viscosity solutions to PDEs developed in the 80's, and it can be used to characterize the value function of non-Markovian stochastic control problems. For an overview of the theory of viscosity solutions for PPDEs and some applications, see [28] for a survey.

In a general context, PPDEs can be formulated in terms of non-Markovian FBSDEs, where the coefficients (f, g etc.) in the system can depend on the entire path of the stochastic process. However, allowing the variables to be path-dependent can lead to challenges both theoretically and numerically.

On the other hand, solving the PPDE/FBSDE is of great interest since they arise naturally in various financial context, see [33],[23],[15],[26]. Recently, some attention has been given to finding the numerical solutions of the PPDEs [29], where LSTM together with path signatures are used, and in [30], the LSTM method is used with the deep Galerkin method. On the other hand, we note that the algorithm in [30] takes a long time to converge. In the meantime, some option pricing problems arising from the Volterra SDEs also lead to path-dependent PDEs. Numerical algorithms are also designed to solve such problems [19], [27].

In this work, inspired by the work in [20] [21] and the theoretical findings in [22], we propose to introduce the path signature as the underlying features in the replacement of the original paths of the SDE. The benefit, as pointed out in both [20] and [14], is that financial timeseries data is usually of high frequencies, and path signature does not lose information when truncated at high enough order.

We noted that, in [20], it is reported that the RNN structure together with the path signatures can very well learn the solution of an SDE given only the simulated Brownian paths. We observe in [14] that the exact same idea was adopted to solve non-markovian FBSDEs. More specifically, SDE paths are first simulated on a fine meshgrid. Then, truncated path signatures/log-signatures are generated on segments of the fine meshgrid which are defined through a coarse meshgrid. Then, signatures on each segment will be used as features. Together with the RNN structure, the function Z_{t_n} in the BSDE will be approximated recursively at discrete time locations.

In this work, we report two new methods (forward and backward methods) which are also based on using the path signature similar to [14] and [13] as features. The contributions of this paper can be summarized as follows:

- Two neural network structures are proposed with proof of universal approximation/convergence given, and numerical examples provided. We use a framework the same as [32]. That is, instead of using RNNs for the function approximation purpose, we use individual neural networks to learn the function Z_{t_i} at each discrete time, where the argument of the function will be the signature of the entire path X_t up to time t_i. Since individual neural networks are used, the input will be the truncated signatures of paths starting from time 0.

- The method in [14] can only be used to solve FBSDEs related to semi-linear parabolic PDEs of path dependent type. We also propose a backward algorithm that can solve the reflected FSBDE (optimal stopping types), which are directly related to the pricing of American options. The algorithm is inspired by the work in [31].
- We solve the open computational problem stated in [14]. Both works [14] and [13] have the limitation of not being able to solve problems of high dimensions: the highest dimension problem they can solve is 20. We propose a method that uses the technique from [5]. More specifically, we pass the simulated paths through an embedding layer prior to the construction of the signature of the paths. The embedding layer plays the role of *dimension reduction and information extraction*. We remark that the data-stream structure is still preserved after the original data stream is processed.

The result of the paper is structured as follows. In Section 2, we give a brief introduction of the signature method and state our main algorithms for both the forward and backward methodologies. We comment that a different backward algorithm was also considered by [13]. Assumptions made in the paper are given in Section 3. Convergence analysis for both the Markovian and non-Markovian FBSDEs are provided in Section 4. The proof for the forward algorithm was inspired by the proof in [14], which is now much simplified because the recurrent neural network structure is no longer adopted. Instead, a collection of individual neural networks are used at discrete times for function approximation purpose. The proof for the backward algorithm is similar to that in [16] with minor changes, and we also provide it here for completeness. Lastly, in Section 5 we provide numerical examples and compare our results to those in the literature. The code for all the tests can be found on Github: https://github.com/Huisun317/path-dependent-FBSDE-/tree/main

2. Algorithm and notations.

2.1. A quick introduction to Signature methods.

In this section, we give a brief introduction to the path signature. We first provide definition of the signature of a path and then present a few examples, which is then followed by the log-signature. We comment that the dimension of the signature of a path increases exponentially with the dimension of the underlying state process. The log-signature on the other hand will effectively reduce the dimension without losing the information content of the path signature. We then introduce Chen's identity, which is needed for the efficiency of numerical computation of signatures. Lastly, we introduce the universal nonlinearity of proposition, which shows the approximation power of functions using the signature as features to the path dependent functions.

Definition 2.1. (Definition A.1.[5]) Let $a, b \in \mathbb{R}$ and $X = (X^1, ..., X^d) : [a, b] \to \mathbb{R}^d$ be a continuous piecewise smooth path. The signature of X is then defined as the collection of iterated integrals.

$$
\begin{aligned}
Sig(X)_{a,b} &= \Big(\int_{a<t_1<\ldots<t_k<b} \ldots \int dX_{t_1} \otimes \ldots \otimes dX_{t_k} \Big)_{k\geq 0} \\
&= \left(\Big(\int_{a<t_1<\ldots<t_k<b} \ldots \int dX^{i_1}_{t_1} \otimes \ldots \otimes dX^{i_k}_{t_k} \Big)_{1\leq i_1,\ldots,i_k\leq d} \right)_{k\geq 0}. \qquad (2)
\end{aligned}
$$

We note that the truncated signature is defined as

$$Sig^m(X)_{a,b} = \Big(\int_{a<t_1<...<t_k<b} ... \int dX_{t_1} \otimes ... \otimes dX_{t_k} \Big)_{0\leq k\leq m}. \tag{3}$$

The signature can be expressed in the formal power series; by using still the same notation (see [6], Definition 5),

$$Sig(X)_{a,b} = \sum_{k=0}^{\infty} \sum_{i_1,...,i_k\in\{1,...,d\}} Sig(X)_{a,b}^{i_1,...,i_k} e_{i_1}...e_{i_k} \tag{4}$$

where k stands for the level of the truncation, and $i_1, i_2, ..., i_k \in \{1, ..., d\}$ is the multi-index. We provide simple examples for the illustration of path signatures. For more examples and theoretical results, see [20] [21] [22] [6] and [5].

Example 2.1. We take $d = 1$. Then, we have

$$\begin{aligned} Sig^1(X)_{a,b} &= X_b - X_a, \\ Sig^2(X)_{a,b} &= \frac{(X_b - X_a)^2}{2!}, \\ Sig^3(X)_{a,b} &= \frac{(X_b - X_a)^3}{3!}, \\ &... \end{aligned}$$

Then, the signature of the path $X : [a, b] \to \mathbb{R}$ is given by

$$(1, Sig^1(X)_{a,b},\ Sig^2(X)_{a,b},\ Sig^3(X)_{a,b}, ...).$$

Example 2.2. We take $d = 2$, and for $X_t = (X_t^1, X_t^2)$,

$$\begin{aligned} Sig^1(X)_{a,b} =& (\int_a^b dX_{t_1}^1, \int_a^b dX_{t_2}^2), \\ Sig^2(X)_{a,b} =& (\int_a^b \int_a^{t_2} dX_{t_1}^1 dX_{t_2}^1, \int_a^b \int_a^{t_2} dX_{t_1}^1 dX_{t_2}^2, \\ & \int_a^b \int_a^{t_2} dX_{t_1}^2 dX_{t_2}^2, \int_a^b \int_a^{t_2} dX_{t_1}^2 dX_{t_2}^1), \\ & ... \end{aligned} \tag{5}$$

Then, the signature of path $X : [a, b] \to \mathbb{R}^d$ is given by

$$(1, Sig^1(X)_{a,b},\ Sig^2(X)_{a,b}, ...).$$

We comment that there is a transformation of the the path signature called the log signature, which corresponds to taking the formal logarithm of the signature in the algebra of formal power series. For a power series x where

$$x = \sum_{k=0}^{\infty} \sum_{i_1,...,i_k\in\{1,...,d\}} \lambda_{i_1,...,i_k} e_{i_1}...e_{i_k} \tag{6}$$

for $\lambda_0 > 0$, the logarithm is defined as

$$\log(x) = \log(\lambda_0) + \sum_{n\geq 1} \frac{(-1)^n}{n} (1 - \frac{x}{\lambda_0})^{\otimes n}. \tag{7}$$

Definition 2.2. (Log signature **Definition 6** [6]). For a path $X : [a, b] \to \mathbb{R}^d$, the log signature of X is defined as the formal power series $\log S(X)_{a,b}$.

Definition 2.3. (Concatenation **Definition 4** [6]) For two paths $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$, we define the concatenation as the path $X * Y : [a, c] \to \mathbb{R}^d$ for which

$$(X * Y)_t = \begin{cases} X_t, & t \in [a, b] \\ X_b + (Y_t - Y_b), & t \in [b, c] \end{cases} \tag{8}$$

Chen's identity can provide a simpler computational method for two paths that are concatenated. This theorem will be helpful if one needs to consider the signature of different paths that have the same origin and large amount of overlaps.

Theorem 2.4. *(Chen's Identity **Theorem 2** [6]). Let $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ be two paths. Then,*

$$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c}. \tag{9}$$

We also give the definition of the time augmented path. This is needed because, in the numerical implementation, we will always use this augmented process.

Definition 2.5. Given a path $X : [a, b] \to \mathbb{R}^d$, we define the corresponding time-augmented path by $\hat{X} = (t, X_t)$, which is a path in $\mathbb{R}^{d+1}$.

We provide the following proposition on universal non-linearity as it gives the theoretical grounding/motivation for us to use deep neural networks to approximate the functional F. Basically, the proposition says that the function of the path is approximately linear on the signature. In some sense, the signature can be treated as a 'universal nonlinearity' on paths.

Proposition 2.6. *(Universal nonlinearity **Proposition A.6** [5]) Let F be a real-valued continuous function on continuous piecewise smooth paths in $\mathbb{R}^d$, and let $\mathcal{K}$ be a compact set of such paths. Furthermore, assume that $X_0 = 0$ for all $X \in \mathcal{K}$. Let $\epsilon > 0$. Then there exists a linear functional L such that, for all $X \in \mathcal{K}$,*

$$|F(X) - L(Sig(\hat{X})| < \epsilon. \tag{10}$$

2.2. **Algorithms.** We use the following numerical scheme to approximate the solution of the forward SDE for (1):

$$X^{\tilde{N}}_{\tilde{t}_{i+1}} = X^{\tilde{N}}_{\tilde{t}_i} + b(\tilde{t}_i, X^{\tilde{N}}_{\tilde{t}_i})h + \sigma(\tilde{t}_i, X^{\tilde{N}}_{\tilde{t}_i})\Delta W^{\tilde{N}}_{\tilde{t}_i} \tag{11}$$

where we use tilde to denote that we have a fine mesh, and $h = T/\tilde{N}$. Based on the fine meshgrid, we also have a coarse mesh grid that has N total segments, where we define $\Delta t := \frac{T}{N} = hM$, and where M is the number of fine grids in each segment. In this case, it is then clear that $N = T/hM$. Note that $\Delta W^{\tilde{N}}_{\tilde{t}_i} = W^{\tilde{N}}_{\tilde{t}_{i+1}} - W^{\tilde{N}}_{\tilde{t}_i}$. Notice that one can define $\{X^N_{t_n}\}_{1 \le n \le N}$ by naturally taking snapshots of $\{X^{\tilde{N}}_{\tilde{t}_i}\}_{0 \le i \le \tilde{N}}$ at $i = 0, M, 2M, ..., MN$. $\{W^N_n\}_{0 \le n \le N-1}$ can be defined similarly. The motivation for defining these two solutions with total steps of size $\tilde{N}$ and N is that the financial data are usually of high frequency, hence more information can be obtained when using more granular data (larger time discretization number in our case).

To approximate Z_t, we use a sequence of neural networks defined on this courser meshgrid $\{0, \frac{T}{N}, , ... \frac{nT}{N}, ..., T\}$, where we denote the approximator as $Z^{N,sig}_{n-1}$. Each neural network will take the truncated signature of the path X_t of level m as input. We denote the truncated signature of the process X up until time t as $\pi_m(Sig(X_{0:t}))$, and that of the approximated path according to (11) as

$$\pi_m(Sig(X^{\tilde{N}}_{0:\tilde{t}_i})), \quad \tilde{t}_i = t_0, t_1, .., t_n, ..., t_{N-1}. \tag{12}$$

In our first approach, the sample-wise solution of the BSDE is then approximated by the following discretization:

$$Y_n^{N,sig} = Y_{n-1}^{N,sig} - f_{n-1}(X_{n-1}^N, Y_{n-1}^{N,sig}, Z_{n-1}^{N,sig})\Delta t + Z_{n-1}^{N,sig}\Delta W_n^N, \tag{13}$$

where we use the short hand notation $f_n(\cdot,\cdot,\cdot) := f(t_n,\cdot,\cdot,\cdot)$, and use n in place of t_n. We remark that here $X_n^N := X_{t_n}^N$ is essentially $X_{Mhn}^{\tilde{N}}$. In later analysis, we will also use $(X_{t_n}^N, Y_{t_n}^{N,sig}, Z_{t_n}^{N,sig})$ in place of $(X_n^N, Y_n^{N,sig}, Z_n^{N,sig})$ to make it clear that those are the approximations at time t_n. This should not cause any confusion since the upper index N denotes that those are quantities obtained through numerical scheme.

We did not specify the terminal condition/discretization because, in our first approach, the sample Y propagates forward: we initialize a batch of the Y_0 and propagate through (13) to match the terminal condition.

On the other hand, we note that such forward method cannot deal with the optimal stopping problems, which typically requires a backward scheme. As such, inspired by the algorithm in [31], we also propose a backward algorithm. But, in this case, the process Z_t takes the path of the process X_t.

$$Y_{n-1}^{N,sig} = Y_n^{N,sig} + f_{n-1}(X_{n-1}^N, Y_n^{N,sig}, Z_{n-1}^{N,sig})\Delta t - Z_{n-1}^{N,sig}\Delta W_n^N \tag{14}$$

where $Z_n^{N,Sig} := Z_n^{\theta_n}(\pi_m(Sig(\tilde{X}_{0:n\Delta t}^{j,\tilde{N}})))$, and $\{Z_n^{\theta_n}\}_{i=1,\dots N-1}$ is the feed-forward neural network estimator.

We point out that the optimal stopping problem in the financial setting has a PDE counterpart

$$\begin{cases} \min\{-\partial_t u - \mathcal{L}u - f(t,x,u,\sigma^T D_x u), u-g\}, \quad t\in[0,T), x\in\mathbb{R}^d, \\ u(T,x) = g(x), x\in\mathbb{R}^d. \end{cases} \tag{15}$$

where $\mathcal{L}$ is the Itô generator of a diffusion process X_t.

This variational inequality can be linked to the RBSDE (Reflected BSDE).

$$\begin{aligned} &X_t = x + \int_0^t b(s,X_s)ds + \int_0^t \sigma(s,X_s)dW_s, \\ &Y_t = g(X_T) + \int_t^T f(s,X_s,Y_s,Z_s)ds - \int_t^T Z_s dW_s + K_T - K_t, \\ &Y_t \geq g(X_t),\ 0\leq t\leq T. \end{aligned} \tag{16}$$

where K is an adapted non-decreasing process satisfying

$$\int_0^T [Y_t - g(X_t)]dK_t = 0.$$

Accordingly, the numerical scheme in general takes the form

$$\begin{cases} Y_T^{N,Sig} = g(X_{0:T}^{\tilde{N}}) \\ Y_{n-1}^{N,sig} = Y_n^{N,sig} + f_{n-1}(X_{n-1}^N, Y_n^{N,sig}, Z_{n-1}^{N,sig})\Delta t - Z_{n-1}^{N,sig}\Delta W_n^N \\ Y_{n-1}^{N,sig} = \max(g(X_{n-1}^N), Y_{n-1}^{N,sig}). \end{cases} \tag{17}$$

We state the algorithms for both the forward and backward Deep Signature algorithm in **Algorithms** 1 and 2.

Finally, we note that, due to the fact that the dimension of the truncated signatures increase exponentially with the dimension of the underlying paths, solving path-dependent FBSDEs of very high dimensions become impractical when the

state process is of very high dimension. The generated signature is of dimension $d^{m+1}/(d-1)$, where d is the dimension of the state process, and m is the level of the truncated depth of the signature. When the underlying process $\{X_t\}_{0\le t\le T}$ is of dimension 20 or more, even the alternative log-signatures will take high dimensions.

The approach that we take to overcome this difficulty is to pass the underlying paths say $\{X_n\}_{0\le n\le N-1}$ ($X_n \in \mathbb{R}^d$) through an embedding layer while still keeping the datastream structure. The trainable embedding layer will project the underlying process $\{X_n\}_{0\le n\le N-1}$ to a dimension $\tilde{d} < d$. The underlying squence after the transformation will then be used to generate signatures which will be passed through a sequence of individual neural networks or RNNs [14] to approximate Z_{t_n}. In the training process, error back-propagation of the signatures also needs to be done since the parameters of the embedding layers need to be trained. As such, we will use the Signatory library available in PyTorch since it facilitates such calculation [5]. This way, the dimension of the original data sequence is effectively reduced, which facilitates the training process.

Algorithm 1 Algorithm for forward deep FBSDE Signature method

Require: Initializing the following terms

- Y_0, margin ϵ, and $epoch = 0$, the total number of epoch iterations. A binary indicator *Embedding=0,1*.
- Feedforward neural network $\{Z_n^{\theta_n}\}_{i=0,...N-1}$. Embedding layer if *Embedding=1*.
- Time discretization h which determines the total number of temporal discretizations $\tilde{N}$ and the total number of segments N which determines coarser grid size Δt.

1: **while** $LOSS(Y_0) \ge \epsilon$ or $Iter > Epoch$ **do**

2:
- Randomly sample batch B of Brownian paths $(\tilde{W}_0, \tilde{W}_h, ..., \tilde{W}_{nMh}, ...\tilde{W}_{N\tilde{h}})$ and accordingly the state process $(\tilde{X}_0^{j,\tilde{N}}, ..., \tilde{X}_h^{j,\tilde{N}}, \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{N\tilde{h}}^{j,\tilde{N}})$
- Pass $(\tilde{X}_0^{j,\tilde{N}}, ..., \tilde{X}_h^{j,\tilde{N}}, \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{N\tilde{h}}^{j,\tilde{N}})$ through the embedding layer if *Embedding=1*. Still name the output $(\tilde{X}_0^{j,\tilde{N}}, ..., \tilde{X}_h^{j,\tilde{N}}, \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{N\tilde{h}}^{j,\tilde{N}})$.
- Create truncated signature (log-signature)

$$\Big(\pi_m(Sig(\tilde{X}_0^{j,\tilde{N}})), \pi_m(Sig(\tilde{X}_{0:\Delta t}^{j,\tilde{N}})), ..., \pi_m(Sig(\tilde{X}_{0:n\Delta t}^{j,\tilde{N}})), ..., \pi_m(Sig(\tilde{X}_{0:N\Delta t}^{j,\tilde{N}}))\Big)_{0\le j\le B}$$

- Compute $Z_n^{j,N,Sig} = Z_n^{\theta_n}(\pi_m(Sig(\tilde{X}_{0:n\Delta t}^{j,\tilde{N}})))$
- For each path $\{\tilde{X}_{0:\tilde{N}}^{j,\tilde{N}}\}_{1\le j\le B}$, find $Y_n^{j,N,Sig}$ iteratively using the Euler scheme (13).

3: Compute the loss by matching the terminal conditions

$$Loss(Y_0) = \frac{1}{B}\sum_{j=1}^{B}(Y_T^{j,N,Sig} - g(X_T^{j,\tilde{N},Sig}))^2$$

4: Update the parameters through gradient descent. $Iter = Iter + 1$

5: **end while**

6: **return** Y_0

Algorithm 2 Algorithm for Backward deep FBSDE Signature method

Require: Initializing the following terms

- A margin ϵ and $epoch = 0$, the total number of epoch iterations. A binary indicator *Embedding=0,1.*
- Feedforward neural network $\{Z_n^{\theta_n}.\}_{i=0,...N-1}$. Embedding layer if *Embedding=1*
- Time discretization h which determines the total number of temporal discretizations $\tilde{N}$ and the total number of segments N which determines coarser grid size Δt.

1: **while** $LOSS(Y_0, Z_0) \geq \epsilon$ or $Iter > Epoch$ **do**

2:
- Randomly sample batch B of Brownian paths $(\tilde{W}_0, \tilde{W}_h, ..., \tilde{W}_{nMh}, ...\tilde{W}_{\tilde{N}\tilde{h}})$ and accordingly the state process $(\tilde{X}_0^{j,\tilde{N}}, \tilde{X}_h^{j,\tilde{N}}, ..., \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{\tilde{N}\tilde{h}}^{j,\tilde{N}})$
- Pass $(\tilde{X}_0^{j,\tilde{N}}, ..., \tilde{X}_h^{j,\tilde{N}}, \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{\tilde{N}\tilde{h}}^{j,\tilde{N}})$ through the embedding layer if *Embedding=1.* Still name the output $(\tilde{X}_0^{j,\tilde{N}}, ..., \tilde{X}_h^{j,\tilde{N}}, \tilde{X}_{nMh}^{j,\tilde{N}}, ...\tilde{X}_{\tilde{N}\tilde{h}}^{j,\tilde{N}})$.
- Create truncated signature (log-signature)

$$\left(\pi_m(Sig(\tilde{X}_0^{j,\tilde{N}})), \pi_m(Sig(\tilde{X}_{0:\Delta t}^{j,\tilde{N}})), ..., \pi_m(Sig(\tilde{X}_{0:n\Delta t}^{j,\tilde{N}})), ..., \pi_m(Sig(\tilde{X}_{0:N\Delta t}^{j,\tilde{N}}))\right)_{0\leq j\leq B}$$

- Compute $Z_n^{j,N,Sig} = Z_n^{\theta_n}(\pi_m(Sig(\tilde{X}_{0:n\Delta t}^{j,\tilde{N}})))$
- For each path $\{\tilde{X}_{0:\tilde{N}}^{j,\tilde{N}}\}_{1\leq j\leq B}$, find $Y_n^{j,N,Sig}$ iteratively using the Euler scheme (14).

3: Compute the loss by finding the variance the batch $\{Y_0^{j,N,Sig}\}_{1\leq j\leq B}$

$$Loss(Y_0^{N,Sig}) = \mathrm{Var}(Y_0^{N,Sig})$$

4: Update the parameters through gradient descent.

5: **end while**

6: **return** Y_0

3. Proof of convergence.

3.1. Assumptions.

Assumption 3.1. *Let b, σ, f, g be deterministic functions such that:*

1. *$b(\cdot, 0), \sigma(\cdot, 0), f(\cdot, 0, 0, 0)$, and $g(0)$ are uniformly bounded.*
2. *b, σ, f, g are uniformly Lipschitz continuous in (x, y, z) with Lipschitz constant L.*
3. *b, σ, f are uniformly Hölder-$\frac{1}{2}$ continuous in t with Hölder constant L.*
4. *f has slow and at most linear growth in y and z:*

$$|f(t, x, y_1, z_1) - f(t, x, y_2, z_2)|^2 \leq K_y|y_1 - y_2|^2 + K_z|z_1 - z_2|^2$$

with K_y and K_z sufficiently small.

We comment that those assumptions are standard for the existence of the BSDE except for *4*, which is needed for the proof for the backward method.

We want to identify the solution of the BSDE with the solution of a semi-linear PDE according to the non-linear Feynman-Kac formula. Hence, we make the following assumption.

Assumption 3.2. *Assume that the following PDE has a classical solution* $u \in C^{1,2}([0,T)\times\mathbb{R}^d;\mathbb{R})$*:*

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}u(t,x) + f(t,x,u,\sigma^T\partial_x u) = 0, \ \ (t,x)\in[0,T)\times\mathbb{R}^d \\ u(T,x) = g(x), \ \ x\in\mathbb{R}^d. \end{cases} \tag{18}$$

Then, under Assumption 3.1 and Assumption 3.2 , one can write (Theorem 5.1.4 [34])

$$Y_t = u(t,X_t), \quad Z_t = \sigma^T(X_t)D_x u(t,X_t).$$

In all the analysis that follows, we will take $d=1$ for simplicity. We introduce the standing assumptions below.

3.2. **Markovian case.** Some simple estimates are given below, For simplicity, we use the following short hand notations, and C denotes a generic constant which may differ from line to line.

i. $\Delta f_n^N = f(t,X_t,Y_t,Z_t) - f_n(t_n,X_n^N,Y_n^{N,Sig},Z_n^{N,Sig})$; $\Delta f_{t_n} = f(t,X_t,Y_t,Z_t) - f_n(t_n,X_{t_n},Y_{t_n},Z_{t_n})$.
ii. $\Delta X_n^N = X_t - X_n^N$; $\Delta X_{t_n} = X_t - X_{t_n}$.
iii. $\Delta Y_n^N = Y_t - Y_n^{N,Sig}$; $\Delta Y_{t_n} = Y_t - Y_{t_n}$.
iv. $\Delta Z_n^N = Z_t - Z_n^{N,Sig}$; $\Delta Z_{t_n} = Z_t - Z_{t_n}$.
v. $\Delta g_n^N = g(X_t) - g(X_n^N)$; $\Delta g_{t_n} = g(X_t) - g(X_{t_n})$.

The following standard result from [34] is needed for the proof later.

Theorem 3.3. *Let Assumption 3.1 hold and assume* Δt *is small. Then,*

$$\max_{0\le n\le N}\mathbb{E}[\sup_{t_n\le t\le t_{n+1}}|Y_t - Y_{t_n}|^2] + \sum_{n=0}^{N-1}\mathbb{E}[\int_{t_n}^{t_{n+1}}|Z_t - Z_{t_n}|^2 dt] \le C(1+|x|^2)\Delta. \tag{19}$$

Lemma 3.4. *By Assumption 3.1 and Assumption 3.2, the following inequality holds true:*

i. $\mathbb{E}[|\Delta X_n|^2] \le C\Delta t$; $\mathbb{E}[|\Delta X_n^N|^2] \le C\Delta t$.
ii. $\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta f_n^N|^2 dt] \le C\Delta t^2 + C\mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2\Delta t] + C\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta Z_n^N|^2 dt]$.

Proof. The result for i) is standard, which is from Theorem 5.3.1 in [34]. ii) follows from the following sequence of inequalities and Theorem 3.3: for $t_n \le t \le t_{n+1}$ for any t,

$$\begin{aligned}
\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta f_n^N|^2 dt] &\le 2\mathbb{E}[\int_{t_n}^{t_{n+1}}|f(t,X_t,Y_t,Z_t) - f(t_n,X_t,Y_t,Z_t) + f(t_n,X_t,Y_t,Z_t) \\
&\quad - f(t_n,X_{t_n},Y_{t_n},Z_t)|^2 + |f(t_n,X_{t_n},Y_{t_n},Z_t) \\
&\quad - f(t_n,X_n^N,Y_n^{N,Sig},Z_n^{N,Sig})|^2 dt] \\
&\le C\Delta t^2 + C\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta Y_{t_n}|^2 + |Y_{t_n} - Y_n^{N,Sig}|^2 dt] \\
&\quad + C\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta Z_n^N|^2 dt] \\
&\le C\Delta t^2 + C\mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2\Delta t] + C\mathbb{E}[\int_{t_n}^{t_{n+1}}|\Delta Z_n^N|^2 dt].
\end{aligned}$$

□

3.2.1. *Forward algorithm.* We note that the proof follows similar lines as in [14], which is now simplified because the recurrent neural network structure is no longer adopted. Instead, individual neural networks are used at discrete times for the function approximation purpose.

Lemma 3.5. *Under Assumption 3.1 and Assumption 3.2, one has*

$$Z_t = \sigma^T(X_t) D_x u(t, X_t)$$

Further, assume that the function $F(t,x) := \sigma^T(x) D_x u(t,x)$ is Lipschitz in x, then there exist $Z_n^{N,Sig}(\cdot)$ for different $0 \le n \le N-1$ such that the following is true:

$$\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2 dt] \le C\Delta t. \tag{20}$$

Proof. We have the following sequence of inequalities:

$$\begin{aligned}
\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_{t_n}^{N,Sig}|^2 dt] &\le 2 \sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_{t_n}|^2 + |Z_{t_n} - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t + 4 \sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |F(t_n, X_{t_n}) - F(t_n, X_{t_n}^N)|^2 \\
&\quad + |F(t_n, X_{t_n}^N) - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t + 4 \sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |F(t_n, X_{t_n}^N) - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t. \qquad (21)
\end{aligned}$$

The last term can be made arbitrarily small by the universal non-linearity of the path signature. □

Next, we state and prove the main theorem of the algorithm.

Theorem 3.6. *Let the assumptions made in Lemma 3.5 hold. Then, there exist $Z_n^{N,Sig}(\cdot)$ for different $0 \le n \le N-1$ such that the following is true:*

$$\max_{0 \le n \le N} \mathbb{E}[\sup_{t_n \le t \le t_{n+1}} |Y_t - Y_n^{N,Sig}|^2] \le C\Delta t. \tag{22}$$

Proof. Taking the difference between the set of equations

$$\begin{cases} Y_{t_{n+1}} = Y_{t_n} - \int_{t_n}^{t_{n+1}} f(t, X_t, Y_t, Z_t) dt + \int_{t_n}^{t_{n+1}} Z_t dW_t \\ Y_{n+1}^{N,Sig} = Y_n^{N,Sig} - \int_{t_n}^{t_{n+1}} f(t_n, X_n^N, Y_n^{N,Sig}, Z_n^{N,Sig}) dt + \int_{t_n}^{t_{n+1}} Z_n^{N,Sig} dW_t \end{cases} \tag{23}$$

we obtain

$$\Delta \bar{Y}_{n+1}^{N,Sig} = \Delta \bar{Y}_n^{N,Sig} - \int_{t_n}^{t_{n+1}} (\Delta f_n) dt + \int_{t_n}^{t_{n+1}} (Z_t - Z_n^{N,Sig}) dW_t, \tag{24}$$

where $\Delta \bar{Y}_n^{N,Sig} := Y_{t_n} - Y_n^{N,Sig}$. By squaring both sides and using Young's inequality with epsilon, we have

$$\begin{aligned}
\mathbb{E}|\Delta \bar{Y}_{n+1}^{N,Sig}|^2 &\le (1 + \frac{\Delta t}{\epsilon}) \mathbb{E}[\mathbb{E}[|\Delta \bar{Y}_n^{N,Sig} + \int_{t_n}^{t_{n+1}} (Z_t - Z_n^{N,Sig}) dW_t|^2 | \mathcal{F}_{t_n}]] \\
&\quad + (1 + \frac{\epsilon}{\Delta t}) \mathbb{E}[\Big(- \int_{t_n}^{t_{n+1}} \Delta f_n dt \Big)^2]
\end{aligned}$$

$$
\begin{aligned}
&\le (1+\frac{\Delta t}{\epsilon})\mathbb{E}[\mathbb{E}[|\Delta \bar{Y}_n^{N,Sig}|^2 + |\int_{t_n}^{t_{n+1}} (Z_t - Z_n^{N,Sig})dW_t|^2|\mathcal{F}_{t_n}]] \\
&\quad + 2(1+\frac{\epsilon}{\Delta t})\Big(\mathbb{E}[\int_{t_n}^{t_{n+1}} |\Delta f_n|^2 dt]\Delta t\Big) \\
&\le (1+C\Delta t)\mathbb{E}[|\Delta \bar{Y}_n^{N,Sig}|^2] + C(\Delta t^2 + \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2 dt] \\
&\quad + \mathbb{E}[|\Delta \bar{Y}_n^{N,Sig}|^2]\Delta t) + C\mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2]dt \qquad (25) \\
&\le (1+C\Delta t)\mathbb{E}[|\Delta \bar{Y}_n^{N,Sig}|^2] + C(\Delta t^2 + \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2]dt),
\end{aligned}
$$

where we have picked $1/\epsilon$ to be a constant C that does not depend on the size Δt. And, in the third inequality, we used Lemma 3.4. Then, by the discrete Gronwall inequality, by Lemma 3.5 and Theorem 3.3 we have

$$
\sup_{0\le n\le N-1} \mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2] \le C\Delta t. \qquad (26)
$$

Then, together with Theorem 3.3, the claim is proved. □

3.2.2. *Backward algorithm.* Since the convergence takes the variance of $Y_0^{N,Sig}$ as the loss function, we first prove an a priori estimate for the loss.

Now we state the main thereom regarding the convergence of the backward scheme where we provide a lower bound for $\mathrm{Var}(Y_0^{N,Sig})$.

Theorem 3.7. *For any $\epsilon > 0$, we have*

$$
\begin{aligned}
Var(Y_0^{N,Sig}) > (1-\epsilon - \frac{2TK_z}{\epsilon})\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t \\
- \frac{4T^2K_y}{\epsilon}\mathbb{E}[\max_{0\le n\le N-1} |Y_{t_n} - Y_n^{N,Sig}|^2]. \qquad (27)
\end{aligned}
$$

Proof. Noting that the true solution Y_0 is deterministic, and that $\mathrm{Var}(Y_0^{N,Sig}) = \mathrm{Var}(Y_0 - Y_0^{N,Sig})$, we have the following inequalities:

$$
\begin{aligned}
\mathrm{Var}(Y_0^{N,Sig}) &= \mathbb{E}\Big[\big(\Delta g_n^N + \int_0^T \Delta f_n^N ds - \int_0^T \Delta Z_n^N dW_t - \mathbb{E}[\Delta g_n^N + \int_0^T \Delta f_n^N ds]\big)^2\Big] \\
&\ge \mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\
&\quad - 2\mathbb{E}\Big[\int_t^T \Delta Z_n^N dW_t(\Delta g_n^N + \int_0^T \Delta f_n^N dt - \mathbb{E}[\Delta g_n^N + \int_0^T \Delta f_n^N dt])\Big] \\
&\ge (1-\epsilon)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\
&\quad - \frac{1}{\epsilon}\mathbb{E}[(\Delta g_n^N + \int_0^T \Delta f_n^N dt - \mathbb{E}[\Delta g_n^N + \int_0^T \Delta f_n^N dt])^2] \qquad (28) \\
&\ge (1-\epsilon)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - \frac{1}{\epsilon}\mathbb{E}[(\Delta g_n^N + \int_0^T \Delta f_n^N dt)^2] \\
&\ge (1-\epsilon)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - \frac{2}{\epsilon}(\mathbb{E}[|\Delta g_n^N|^2] + T\mathbb{E}[\int_0^T |\Delta f_n^N|^2 dt])
\end{aligned}
$$

$$
\begin{aligned}
&\geq (1-\epsilon)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t \\
&\quad - \frac{2}{\epsilon}(C\Delta t + T\mathbb{E}[\int_0^T K_y|\Delta Y_n^N|^2 + K_z|\Delta Z_n^N|^2 dt]) \\
&\geq (1-\epsilon-\frac{2TK_z}{\epsilon})\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t \\
&\quad - \frac{2TK_y}{\epsilon}\mathbb{E}[\int_0^T |Y_t - Y_{t_n} + Y_{t_n} - Y_n^{N,Sig}|^2 dt] \\
&\geq (1-\epsilon-\frac{2TK_z}{\epsilon})\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t \\
&\quad - \frac{4T^2K_y}{\epsilon}\mathbb{E}[\max_{0\leq n\leq N-1} |Y_{t_n} - Y_n^{N,Sig}|^2], \qquad (29)
\end{aligned}
$$

where in the inequalities we used the fact that $Var(X) \leq \mathbb{E}[X^2]$ and the second inequality that $2ab \geq -\epsilon a^2 - \frac{1}{\epsilon^2}b^2$. In the second to last inequality, we used Theorem 3.3. □

We state the following main theorem about the convergence of the backward algorithm.

Theorem 3.8. *Under Assumption 3.1 and Assumption 3.2, the following inequality holds:*

$$
\sup_{t\in[0,T]} \mathbb{E}[|Y_t - Y_n^{N,Sig}|^2] + \int_0^T \mathbb{E}[|Z_t - Z_n^{N,Sig}|^2 dt] \leq C(\Delta t + Var(Y_0^{N,Sig})). \qquad (30)
$$

Proof. Taking the difference between the exact solution and the numerical solution, we have the following inequality:

$$
\begin{aligned}
\mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|] &\leq 3\Big(\mathbb{E}[|\Delta g^N|^2] + T\mathbb{E}[\int_0^T |\Delta f_n^N|^2 dt + \int_0^T |\Delta Z_n^N|^2 dt]\Big) \\
&\leq 3C\Delta t + C\Delta t + 3T\mathbb{E}[\int_0^T K_y|\Delta Y_n^N|^2 + K_z|\Delta Z_n^N|^2 dt] \\
&\quad + 3\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\
&\leq C\Delta t + 3(1+TK_y)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\
&\quad + 3TK_y\mathbb{E}[\int_0^T |Y_t - Y_{t_n} + Y_{t_n} - Y_n^{N,Sig}|^2 dt] \\
&\leq C\Delta t + 3(1+TK_y)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\
&\quad + 6T^2K_y \max_{0\leq n\leq N-1} \mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2]. \qquad (31)
\end{aligned}
$$

This shows that

$$
\begin{aligned}
\max_{0\leq n\leq N-1} \mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2] &\leq \frac{C\Delta t + 3(1+TK_y)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt]}{1-6T^2K_y} \\
&\leq C\Delta t + \frac{3(1+TK_y)}{1-6T^2K_y}\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt]. \qquad (32)
\end{aligned}
$$

Using Theorem 4.4 ((29)), we have

$$\begin{aligned}\mathrm{Var}(Y_0^{N,Sig}) &\geq (1-\epsilon-\frac{2TK_z}{\epsilon})\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t \\ &\quad - \frac{4T^2K_y}{\epsilon}\big(C\Delta t + \frac{3(1+TK_y)}{1-6T^2K_y}\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt]\big) \\ &\geq \Big((1-\epsilon-\frac{2TK_z}{\epsilon}) + \frac{4T^2K_y}{\epsilon}\frac{3(1+TK_y)}{1-6T^2K_y}\Big)\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] - C\Delta t. \end{aligned} \tag{33}$$

Then, take $\epsilon = \sqrt{2TK_z + 4T^2K_y\frac{3(1+TK_y)}{1-6T^2K_y}}$ by the assumption on K_z and K_y, since $2\epsilon < 1$, This implies that

$$\mathbb{E}[\int_0^T |\Delta Z_n|^2 dt] \leq C(\Delta t + \mathrm{Var}(Y_0^{N,Sig})). \tag{34}$$

Together with (32) and Theorem 3.3, we have

$$\sup_{t\in[0,T]} \mathbb{E}[|Y_t - Y_{t_n}^{N,Sig}|^2] + \int_0^T \mathbb{E}[|Z_t - Z_{t_n}^{N,Sig}|^2 dt] \leq C(\Delta t + \mathrm{Var}(Y_0^{N,Sig})). \tag{35}$$

□

Letting $F(t,x) := \sigma^T(x)D_x u(t,x)$, we have the following theorem.

Theorem 3.9. *Let the assumptions in Lemma 3.5 hold. Then, the following inequality holds:*

$$Var(Y_0^{N,Sig}) \leq C\Big(\Delta t + \sum_{0\leq n\leq N-1} \mathbb{E}[|F(t_n, X_n^N) - Z_{t_n}^{N,Sig}|^2]\Delta t\Big). \tag{36}$$

Proof. Again, note that the true solution Y_0 is deterministic and that $\mathrm{Var}(Y_0^{N,Sig}) = \mathrm{Var}(Y_0 - Y_0^{N,Sig})$. We then have the following result:

$$\begin{aligned}\mathrm{Var}(Y_0^{N,Sig}) &= \mathbb{E}\Big[\big(\Delta g_n^N - \int_0^T \Delta f_n^N ds + \int_0^T \Delta Z_n^N dW_t - \mathbb{E}[\Delta g_n^N - \int_0^T \Delta f_n^N ds]\big)^2\Big] \\ &\leq 2\mathbb{E}\Big[(\Delta g_n^N - \int_0^T \Delta f_n^N ds - \mathbb{E}[\Delta g_n^N - \int_0^T \Delta f_n^N ds])^2 + \int_0^T |\Delta Z_n|^2 dt\Big] \\ &\leq 2\mathbb{E}\Big[(\Delta g_n^N - \int_0^T \Delta f_n^N ds)^2 + \int_0^T |\Delta Z_n|^2 dt\Big] \\ &\leq 4\mathbb{E}\Big[|\Delta g_n^N|^2 + T\int_0^T |\Delta f_n^N|^2 ds] + 2E[\int_0^T |\Delta Z_n^N|^2 dt\Big] \\ &\leq C\Delta t + C\mathbb{E}[\int_0^T |\Delta Y_n^N|^2 + |\Delta Z_n^N|^2 dt] + 2E[\int_0^T |\Delta Z_n^N|^2 dt\Big] \\ &\leq C\Delta t + C\max_{0\leq n\leq N-1} \mathbb{E}[|Y_{t_n} - Y_n^{N,Sig}|^2] + C\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt] \\ &\leq C\Delta t + C\mathbb{E}[\int_0^T |\Delta Z_n^N|^2 dt], \end{aligned}$$

where we used (32) to bound $\max_{0\le n\le N-1}\mathbb{E}[|Y_{t_n}-Y_n^{N,Sig}|^2]$. Using Theorem 3.3, one has that

$$\begin{aligned}E[\int_0^T|\Delta Z_n^N|^2dt\Big]&\le 2E[\int_0^T|Z_t-Z_{t_n}|^2+|Z_{t_n}-Z_n^{N,Sig}|^2dt\Big]\\&\le 4\mathbb{E}[\int_0^T|Z_t-Z_{t_n}|^2+|F(t_n,X_n^N)-Z_{t_n}(X_{t_n})|^2\\&\quad+|F(t_n,X_n^N)-Z_n^{N,Sig}|^2dt\Big]\\&\le C\Delta t+\sum_{0\le n\le N-1}\mathbb{E}[|F(t_n,X_n^N)-Z_n^{N,Sig}|^2]\Delta t.\end{aligned}\tag{37}$$

Note that $F(t_n,X_n^N)$ can be approximated by $Z_n^{N,Sig}$ arbitrarily close given the same underlying path $\{X_n^N\}_{1\le n\le N-1}$ due to the universal non-linearity of the signature path. As such, inequality (36) holds. □

By putting together Theorem 3.9 and Theorem 3.8, we have the following result, which then shows convergence by the universality property of the path signature.

Theorem 3.10. *Let the assumptions in Lemma 3.5 hold, denoting* $F(t,x):=\sigma^T(x)D_xu(t,x)$. *Then, the following inequality holds:*

$$\begin{aligned}\sup_{t\in[0,T]}\mathbb{E}[|Y_t-Y_n^{N,Sig}|^2]+\int_0^T\mathbb{E}[|Z_t-Z_n^{N,Sig}|^2dt]\le C(\Delta t+\\\sum_{0\le n\le N-1}\mathbb{E}[|F(t_n,X_n^N)-Z_n^{N,Sig}|^2]\Delta t).\end{aligned}\tag{38}$$

3.3. **Non-Markovian generalization.** We have provided the proof for the Markovian case. We now consider the case for non-Markovian FBSDE, i.e., both the driver and the terminal function g depend on the entire path of the process $X_{[0,t]}$. For simplicity, we still consider dimension $d=1$. We introduce the following notation for the path analysis, which are mainly from Dupire [7].

For each $\omega\in\Omega$, $X:\Gamma\to\mathbb{R}$ is the canonical process $X_t(\omega):=\omega_t$. Let $|\cdot|$ denote the norm. For each $0\le t\le t'\le T$, define $\Lambda:=\bigcup_{0\le t\le T}\Lambda_t$ where $\Lambda_t:[0,t]\to\mathbb{R}$ is the path. We write

$$\begin{aligned}||\omega_t||&:=\sup_{r\in[0,t]}|\omega(r)|,\\d_\infty(\omega_t,\omega'_{t'})&:=\max(\sup_{r\in[0,t)}\{|\omega(r)-\omega'(r)|\},\sup_{r\in[t,t']}\{|\omega(t)-\omega'(r)|\})+|t-t'|.\end{aligned}$$

In Dupire's formulation of derivatives, one often regards $u(\omega_t)$ as a function of t,ω,x

$$u(\omega_t^x):=u(t,\omega(s)_{0\le s<t},\omega(t)+x).\tag{39}$$

The definition of spatial derivative is given below,

Definition 3.11. Let $u:\Lambda\to\mathbb{R}$ and $\omega_t\in\Lambda$. If there exists $p\in\mathbb{R}^d$ such that

$$u(\omega_t^x)=u(\omega_t)+p\cdot x+o(|x|),\quad x\in\mathbb{R},\tag{40}$$

then we say that u is vertically differentiable at ω_t, and we denote the gradient of $D_xu(\omega_t)=p$. u is said to be vertically differentiable in Λ if $D_xu(\omega_t)$ exists for each $\omega_t\in\Lambda$. We also define the Hessian $D_{xx}u(\omega_t)$. In a similar fashion, the Hessian is an $\mathbf{S}(d)$-valued function defined on Λ, where $\mathbf{S}(d)$ is the space of $d\times d$ symmetric matrices.

For the horizontal derivative, and we first define $\omega_t \in \Lambda$, we denote

$$\omega_{t,s}(r) = \omega(r)\mathbf{1}_{[0,t)}(r) + \omega(t)\mathbf{1}_{[t,s]}(r), \quad r \in [0,s]. \tag{41}$$

Definition 3.12. For a given $\omega_t \in \Lambda$, if we have

$$u(\omega_{t,s}) = u(\omega_t) + a(s-t) + o(|s-t|), \quad s \geq t, \tag{42}$$

then we say that $u(\omega_t)$ is horizontally differentiable in t at ω_t, and denote $D_t u(\omega_t) = a$. u is said to be horizontally differentiable in Λ if $D_t u(\omega_t) = a$ exists for all $\omega_t \in \Lambda$.

Under the assumption to be made in Assumption 3.13, define $u(t, X_t) := Y_t$, then we have the nonlinear Feynman-Kac formula from Proposition 3.8 in [25] that

$$Z_t = D_x u(t, X_t)\sigma(t, x_t) := F(t, X_t). \tag{43}$$

where we denoted X_t as the path $X_{\cdot \wedge t}$ and x_t as the spatial value of the path evaluated at time t,

We make the following strong assumption to obtain a convergence result similar to the Markovian case.

Assumption 3.13. *In addition to Assumption 3.1, assume the driver of the BSDE is also Hölder-$\frac{1}{2}$ continuous in t and it is uniformly Lipschitz continuous in (X, Y, Z) with Lipschitz constant L in the sense above. Also, assume that b, σ, f, g are smooth functions with bounded derivatives. Also assume F is Lipschitz continuous in X with Lipschitz constant L. Namely,*

$$|F(t, X_t) - F(t', X'_{t'})| \leq L d_\infty(X_t, X'_{t'}). \tag{44}$$

One immediate consequence of this assumption is that Theorem 3.3 holds also in the non-Markovian setting.

Theorem 3.14. *Let Assumptions 3.1, 3.13 hold and assume Δt is small. Then,*

$$\max_{0 \leq n \leq N} \mathbb{E}[\sup_{t_n \leq t \leq t_{n+1}} |Y_t - Y_{t_n}|^2] + \sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_{t_n}|^2 dt] \leq C(1 + |x|^2)\Delta t. \tag{45}$$

Proof.

$$\begin{aligned}
\int_{t_n}^{t_{n+1}} \mathbb{E}[|Z_t - Z_{t_n}|^2]dt &= \int_{t_n}^{t_{n+1}} \mathbb{E}[|F(t, X_t) - F(t_n, X_{t_n})|^2]dt \\
&\leq \int_{t_n}^{t_{n+1}} C((\Delta t)^2 + \mathbb{E}[\sup_{t_n \leq t \leq t_{n+1}} |x_t - x_{t_n}|^2])dt \qquad (46) \\
&\leq \int_{t_n}^{t_{n+1}} C\Delta t dt. \qquad (47)
\end{aligned}$$

This will imply that

$$\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_{t_n}|^2 dt] \leq C\Delta t.$$

To see that

$$\max_{0 \leq n \leq N} \mathbb{E}[\sup_{t_n \leq t \leq t_{n+1}} |Y_t - Y_{t_n}|^2] \leq C\Delta t, \tag{48}$$

we have

$$Y_t - Y_{t_n} = -\int_{t_n}^{t} f(s, X_s, Y_s, Z_s)ds + \int_{t_n}^{t} Z_s dW_s. \tag{49}$$

By squaring both sides, taking the sup and then taking the expectation, one obtains

$$\begin{aligned}
\mathbb{E}[\sup_{t_n\le t\le t_{n+1}} |Y_t - Y_{t_n}|^2] &\le C\Delta t\mathbb{E}[\int_{t_n}^{t_{n+1}} |f(s, X_s, Y_s, Z_s) - f(s, 0, 0, 0)|^2 ds] \\
&\quad + C(\Delta t)^2 + C\mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_s|^2 ds] \\
&\le C(\Delta t)^2\mathbb{E}[\sup_{0\le t\le T} |X_t|^2 + \sup_{0\le t\le T} |Y_t|^2] \\
&\quad + C(\Delta t)^2 + C\mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_s|^2 ds] \\
&\le C\Delta t + C\Delta t(\mathbb{E}[\sup_{0\le t\le T} |Y_t|^2] + \mathbb{E}[\sup_{0\le t\le T} |Z_t|^2]).
\end{aligned} \tag{50}$$

We used that $\mathbb{E}[||X_t||^2] \le \mathbb{E}[\sup_{0\le t\le T} |X_t|^2] \le C$ for some constant C. Also, by using Lemma 3.3 and Corollary 3.9 both from [25], we have the boundness result of (Y_t, Z_t):

$$\begin{aligned}
\mathbb{E}[\sup_{0\le t\le T} |Y_t|^2] &\le C, \\
\mathbb{E}[\sup_{0\le t\le T} |Z_t|^2] &\le C.
\end{aligned} \tag{51}$$

Then, one has the desired result. By combining (48) and (47), we obtained the desired result. □

One immediate result is a non-Markovian version of Lemma 3.5.

Lemma 3.15. *Under Assumptions 3.1, 3.13, there exist $Z_n^{N,Sig}(\cdot)$ for different $0 \le n \le N-1$ such that the following is true:*

$$\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2 dt] \le C\Delta t. \tag{52}$$

Proof. Note that we have the following sequence of inequalities:

$$\begin{aligned}
\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2 dt] &\le 2\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_{t_n}|^2 + |Z_{t_n} - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t + 4\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |F(t_n, X_{t_n}) - F(t_n, X_{t_n}^N)|^2 \\
&\quad + |F(t_n, X_{t_n}^N) - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t + 4\sum_{n=0}^{N-1} \mathbb{E}[\int_{t_n}^{t_{n+1}} |F(t_n, X_{t_n}^N) - Z_n^{N,Sig}|^2 dt] \\
&\le C\Delta t.
\end{aligned} \tag{53}$$

In the second inequality, we used Theorem 3.14, and in the third inequality, we used the Lipschitz assumption on Z_t and the fact that $\max_{0\le n<N} \mathbb{E}[\sup_t |X_t - X_{t_n}^N|^2] \le \Delta t$.

The last term can be made arbitrarily small by the universal non-linearity of the path signature. □

Theorem 3.16. *Let Assumptions 3.1, 3.13 hold. Then, there exist* $Z_n^{N,Sig}(\cdot)$ *for different* $0 \leq n \leq N-1$ *such that the following is true:*

$$\max_{0\leq n\leq N} \mathbb{E}[\sup_{t_n\leq t\leq t_{n+1}} |Y_t - Y_n^{N,Sig}|^2] \leq C\Delta t. \tag{54}$$

Proof. The proof follows the same line as Theorem 3.6, except that in the line of inequality (25), one has the following estimates for Δf_n:

$$\begin{aligned}
\mathbb{E}[\int_{t_n}^{t_{n+1}} |\Delta f_n|^2] &\leq 2\mathbb{E}[\int_{t_n}^{t_{n+1}} |f(t,X_t,Y_t,Z_t) - f(t_n,X_t,Y_t,Z_t)|^2 + \\
&\quad |f(t_n,X_t,Y_t,Z_t) - f(t_n,X_n^N,Y_n^{N,Sig},Z_n^{N,Sig})|^2 dt] \\
&\leq C(\Delta t)^2 + C\mathbb{E}[\int_{t_n}^{t_{n+1}} d_\infty^2(X_t,X_n^N) + |Y_t - Y_n^{N,Sig}|^2 \\
&\quad + |Z_t - Z_n^{N,Sig}|^2 dt] \\
&\leq C(\Delta t)^2 + \mathbb{E}[\int_{t_n}^{t_{n+1}} \sup_{t_n\leq t\leq t_{n+1}} |Y_t - Y_{t_n}|^2 + |Y_{t_n} - Y_n^{N,Sig}|^2 \\
&\quad + |Z_t - Z_n^{N,Sig}|^2 dt] \\
&\leq C(\Delta t)^2 + \mathbb{E}[|\Delta \bar{Y}_n^{N,Sig}|^2]\Delta t + \mathbb{E}[\int_{t_n}^{t_{n+1}} |Z_t - Z_n^{N,Sig}|^2 dt].
\end{aligned} \tag{55}$$

□

Remark 3.17. We note that Assumption 3.13 is a strong assumption on Z_t, and the version of Theorem 3.3 we obtained is also subject to this assumption. A stronger version removing the Lipschtiz requirement is not necessary since it is still needed for proving the non-Markovian version of Theorem 3.6 and Theorem 3.10.

As such, by using a similar argument, one can show that the non-Markovian version of Theorem 3.10 also holds.

4. Numerical examples.

In this section, we provide numerical examples and compare the results obtained using our algorithms to those from [14], [13], and [18]. For all the numerical examples, we generate new batch samples (data) for each iteration. This is different from examples from [14], where a fixed amount of data was pre-generated, say 10^5 trajectories. The main motivation for us to use new samples for each iteration is 1) we want to avoid over fitting, and 2) the data generation process is independent from the training process and it is easy to implement. We comment that we will generate the path signature for $\{X_{\tilde{t}_j}\}_{0\leq\tilde{t}_j\leq t_n}^B$ for each $n = 0, 1, ...N-1$, and $l \in B$ where B is the batch. One can use Chen's identity for the path generation to avoid regeneration of the signatures for the overlapping parts in the paths.

4.1. Lookback options.

Consider the one-dimensional Blackscholes setting where the stock price follows the following dynamics:

$$dX_t = rX_t dt + \sigma X_t dW_t. \tag{56}$$

The terminal payoff is defined to be

$$g(X_{\cdot\wedge T}) = X_T - \inf_{0\le t\le T} X_t. \tag{57}$$

The option price is defined to be

$$Y_t = \exp(-r(T-t))\mathbb{E}[g(X_{\cdot\wedge T})|\mathcal{F}_t] \tag{58}$$

Then, $e^{-rt}Y_t = \mathbb{E}[e^{-rT}g(X_{\cdot\wedge T})|\mathcal{F}_t]$ is a martingale, meaning that

$$d(e^{-rt}Y_t) = Z_t' dW_t \tag{59}$$

for some square integrable process Z_t', and then one immediately has

$$dY_t = rY_t dt + Z_t dW_t, \tag{60}$$

where $Z_t := e^{rt}Z_t'$. The solution Y_t has the analytic formula

$$Y_t = X_t\Phi(p_1) - m_t e^{-r(T-t)}\Phi(p_2) - X_t\frac{\sigma^2}{2r}\big(\Phi(-p_1) - e^{-r(T-t)}(\frac{m_t}{yt})^{2r/\sigma^2}\Phi(-p_3)\big), \tag{61}$$

where

$$m_t := \inf_{0\le u\le t} X_t, p_1 = \frac{\log(X_t/m_t) + (r+\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}, p_2 = p_1 - \sigma\sqrt{T-t}$$

and $p_3 = p_1 - \frac{2r}{\sigma}(\sqrt{T-t})$.

For the model specification, similar to [14], we use $\tilde{N} = 2000$ and $N = 20$. We use the truncated signature method with level $m = 3$. We take $x_0 = 1, \sigma = 1, r = 0.01$, and $T = 1$. Since in [14] it is demonstrated that the method therein is state of the art, we present comparison between results obtained using methods in [14] and our forward/backward methods in Table 1.

For the forward algorithm (method 1), we take the batch size to be 100. And, we use the Adam optimizer with learning rate 10^{-3}. Since $N = 20$, we use 20 individual neural network, to approximate Z_{t_n} for each $0 \le n \le N-1$. We use fully connected feedforward neural networks with 2 hidden layers each of 64 neurons. It can be observed from Figure 1 that our forward algorithm produces good results compared to the exact solution: the blue curve is very close to the red dashline.

For the backward algorithm (method), we note that, due to the designed model methodology, it is desirable to use a larger batch size: we are minimizing the variance of the mini-batches and use the mean of the batch samples as the sample estimate for Y_0. We observe that, for this method, convergence happens fast but with values typically fluctuating about a fixed level. This is due to the fact that:

- We are not training the model on a fixed amount of trajectories of $\{X_{\tilde{t}_j}\}_{0\le j\le\tilde{N}}^M$ where M is a fixed amount (say 10000). Instead, to overcome over-fitting, we generate different batch samples for each iteration. Hence, one trajectory of sample path may lead to high variance.
- We minimize the variance of the sample and use the mean of the sample as estimation for Y_0, so the estimated mean may fluctuate according to sample size.

For this numerical example, we use $B = 1000$ as the batch sample size. We perform 50 runs of the algorithm and obtain mean of a 0.578. The confidence interval is $[0.572, 0.584]$. As such, even though the estimated Y_0 for each iteration shows some fluctuation in the sample run, the mean is stable. We show the result of one of the sample runs in Figure 2.

As such, we observe that both methods are comparable to that in [14].

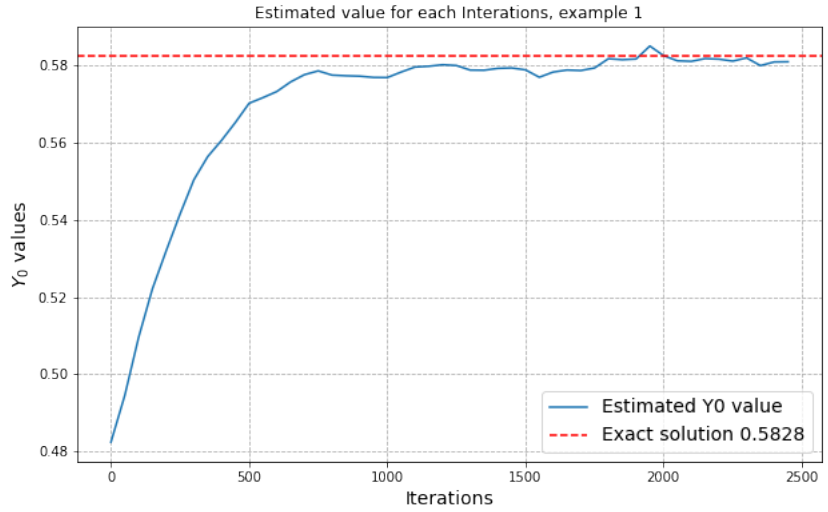

FIGURE 1. Predicted Y_0 value versus the number of interations trained.

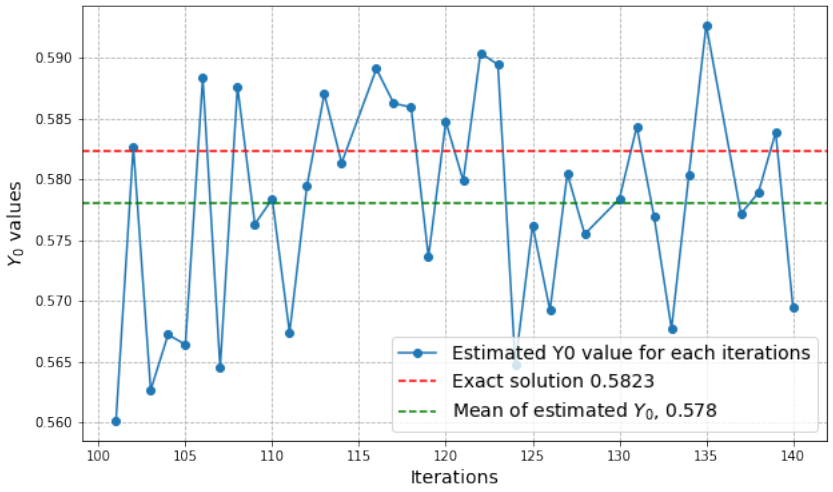

FIGURE 2. Predicted Y_0 value versus the number of interations trained.

4.2. **A higher dimension example.** The last example has dimension $d = 1$, and for this particular example, we take $d = 20$. We consider the high dimension

TABLE 1. Example 1 results comparison

	Exact	Method in [14]	Forward method 1	Backward method 2
Result Y_0	0.5828	0.579	0.581	0.578
Error	–	0.6 %	0.3 %	0.8%

TABLE 2. Example 2 result comparison, $d = 20$

$d = 20$	Exact	Method in [14]	Forward method 1	Backward method 2
Result Y_0	6.66	6.6	6.58	6.71
Error	–	1 %	1.2 %	0.75%
$d = 100$				
Result Y_0	33.33	–	33.15	33.50
Error	–	NA	0.538 %	0.534%

problem

$$\begin{cases} \int_t^T dX_s = \int_t^T dW_s \\ Y_t = g(X_{\cdot\wedge T}) + \int_t^T f(s, X_{\cdot\wedge s}, Y_s, Z_s)ds - \int_t^T Z_s dW_s. \end{cases} \tag{62}$$

For simplicity, we take $f = 0$ and $g(X_{\cdot\wedge T}) = (\int_0^T \sum_i^d X_s^i ds)^2$. We comment that, because the dimension of the signature grows exponentially as a function of the dimension of the underlying state process, we will use the log-signature of paths instead of signatures since it essentially contains the same amount of information but requires less dimensions. We take $d = 20, T = 1, N = 5, \tilde{N} = 100$. Again, it is observed that the results of our forward method 1 and backward method 2 are comparable to the exact solution and the method proposed in [14]. We comment that the convergence of the forward method can be slow in this case as it could be sensitive to the initial guess of Y_0. But, determination of a rough range of Y_0 is easy: Try running the algorithm with guess Y_0^1, and if the estimated result is monotonically increasing during the training process, then stop the algorithm and make a larger guess Y_0^2. Keep increasing the initial guess if it is still increasing, otherwise make a smaller guess. Using this methodology, we start training our algorithm with $Y_0 = 6.0$ and it stabilizes at 6.58. Method 2 converges rapidly, but there is some fluctuation among each iteration within one run. Again, we use the mean estimator and we observe the result is 6.71. The test result for $d = 20$ is summarized in Table 2.

We also present the result when the dimension is very high, $d = 100$, in which case the method in [14] is no longer practical. In this case, we construct a (trainable) embedding layer whose output data stream has state dimension 5. Since the dimension is reduced, we can take larger $\tilde{N}$ for the purpose of accuracy, in which case it is selected to be 10.

We present the PDE related to (62) and state its exact solution for completeness. For $(t, \omega) \in \big([0, T] \times C([0, T], \mathbb{R}^d)\big)$

$$\begin{cases} \partial_t u + \frac{1}{2} tr(\partial_{\omega\omega} u) + f(t, \omega, u\partial_\omega u) = 0 \\ u(T, \omega) = g(\omega), \quad g(\omega) = (\int_0^T \sum_i^d \omega_s^i ds)^2. \end{cases} \tag{63}$$

The exact solution is

$$u(t,\omega) = (\int_0^t \sum_i^d \omega_s^i ds)^2 + (\sum_i^d \omega_t^i)^2 (T-t)^2 + 2(T-t)(\sum_i^d \omega_t^i) \int_0^t \sum_i^d \omega_s^i ds + \frac{d}{3}(T-t)^3.$$

4.3. **An example of non-linear type: Amerasian Option.** As the last example, we provide an example used in [13]. We note that, in this case, our forward Algorithm 1 is not applicable anymore. We will consider only the backward algorithm according to (17).

Amerasian options are considered under the Black-Scholes model that involves d stocks $X_1, ...X_d$ which follow the following SDEs:

$$dX_t^i = rX_t^i dt + \sigma_t^i dW_t^i, \ X_0^i = x_0^i, i = 1, ..., d \tag{64}$$

where the W^i are assumed to be independent. The payoff of the basket Amerasian call option at strike price of K is defined as

$$g(X_{\cdot \wedge T}) = \Big(\sum_{i=1}^d \frac{w_i}{T} \int_0^T X_t^i dt - K \Big)^+, \tag{65}$$

where w_i are defined to be the weights. We consider the price of the Bermudan option, which is a type of American option where early excercise can only happen at prescribed dates. The model variables are taken to be $X_0^i = 100, r = 0.05, \sigma_i = 0.15, \omega = \frac{1}{d}, T = 1$, and $K = 100$. We take $\tilde{N} = 1000$ and $N = 20$. We show the benchmark results against 1) the European price, 2) the method from [13], and 3) the method from [18]. On the analytic side, for benchmark purposes, we comment that:

i. The American option price should be higher than the European price due to its flexibility of being able to exercise early.

ii. By Jensen's inequality, using the current parameters,

$$\mathbb{E}[e^{-rT}\Big(\sum_{i=1}^d \frac{1}{dT} \int_0^T X_t^i dt - K \Big)^+] \geq e^{-rT}\Big(\mathbb{E}[\frac{1}{dT} \sum_{i=1}^d \int_0^T X_t^i dt - K] \Big)^+ = 2.42.$$

iii. Again, we can show that the price of the Ameransian option price is *decreasing*. By using the Jensen inequality, we have

$$e^{-rT}\Big(\frac{1}{2dT} \sum_{i=1}^{2d} \int_0^T X_t^i dt - K \Big)^+ \leq \frac{1}{2} e^{-rT}\Big(\frac{1}{dT} \sum_{i=1}^d \int_0^T X_t^i dt - K \Big)^+ + \frac{1}{2} e^{-rT}\Big(\frac{1}{dT} \sum_{i=d+1}^{2d} \int_0^T X_t^i dt - K \Big)^+.$$

Then, taking the limit in d, we obtain that the price converges to 2.42 when d goes to infinity.

We show our results in Table 3. We run the algorithm for dimensions $d = 1, 5, 10, 20$, and 100 for 50 runs. The mean of the 50 estimates is taken as the estimator, and the confidence intervals are computed based on those samples.

We also note that our results agree with the theoretical findings. That is, compared to the benchmarks, the calculated prices are all above the European prices. We comment that each run takes much less time than reported in [13]; for example,

TABLE 3. Example 3 result comparison $d = 1$

d=1	European	Method in [13]	Method in [18]	Backward
Result Y_0	4.732	4.963	5.113	5.03
Confidence Interval	–	[4.896,5.03]	[5.009,5.217]	[4.97, 5.10]
d=5				
Result Y_0	3.078	3.190	3.335	3.11
Confidence Interval	–	[3.115, 3.266]	[3.207, 3.462]	[3.08,3.155]
d=10				
Result Y_0	2.701	2.914	3.142	2.76
Confidence Interval	–	[2.844,2.983]	[2.975,3.309]	[2.734, 2.813]
d=20				
Result Y_0	2.51	3.093	3.095	2.61
Confidence Interval	–	[3.017,3.168]	[2.883,3.3308]	[2.587, 2.627]

when $d = 5$, the runtime (one run) of our algorithm takes only 369.47 s to achieve the estimate, while in [13], 1927.55 s is reported. In the high dimension case where $d = 100$, the price is found to be 2.516 with confidence interval $[2.506, 2.526]$. We note that, in this case, the method in [13] fails due to the excessively large size of both the signature and path signatures in this case.

In the meantime, observing the trend in the price predicted, we note that our predicted prices agree with the trend better than [13]: when $d = 20$, their predicted price even increased compared to when $d = 10$, which does not align with theoretical result. We comment that, when using the embedding layers, one run for $d = 20$ reduces to 165.0 s, and for $d = 100$ one run takes about 320.0 s.

Acknowledgments. This project is partially supported by U.S. National Science Foundation through project DMS-2142672 and the support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Grant DE-SC0022297.

REFERENCES

[1] R. Archibald, F. Bao, Y. Cao and H. Sun, Numerical Analysis for Convergence of a Sample-Wise Backpropagation Method for Training Stochastic Neural Networks, *SIAM Journal on Numerical Analysis*, **62** (2024), 593-621.

[2] A. Bachouch, C. Huré, N. Langrené and H. Pham, Deep neural networks algorithms for stochastic control problems on finite horizon, part 2: Numerical applications, *Methodol. Comput. Appl. Probab.*, **24** (2022), 143-178.

[3] S. Becker, P. Cheridito and A. Jentzen, Deep optimal stopping, *Journal of Machine Learning Research,* **20** (2019), 1-25.

[4] S. Becker, P. Cheridito, A. Jentzen and T. Welti, Solving high-dimensional optimal stopping problems using deep learning, *European Journal of Applied Mathematics*, **32** (2021), 470-514.

[5] P. Bonnier, P. Kidger, I. P. Arribas, C. Salvi and T. Lyons, Deep Signature Transforms, arXiv:1905.08494

[6] I. Chevyrev and A. Kormilitzin, A primer on the signature method in machine learning, arXiv:1603.03788.

[7] B. Dupire, Functional Itô calculus, *Quantitative Finance*, **19** (2019), 721-729.

[8] I. Ekren, Viscosity solutions of obstacle problems for fully nonlinear path-dependent PDEs, *Stochastic Processes and their Applications*, **127** (2017), 3966-3996.

[9] I. Ekren, C. Keller, N. Touzi and J. Zhang, On viscosity solutions of path-dependent PDEs, *Ann. Probab.,* **42** (2014), 204-236.

[10] I. Ekren, N. Touzi and J. Zhang, Optimal stopping under nonlinear expectation, *Stochastic Processes and their Applications*, **124** (2014), 3277-3311.
[11] I. Ekren, N. Touzi and J. Zhang, Viscosity solutions of fully nonlinear parabolic path dependent PDEs: Part I, *Ann. Probab.*, **44** (2016), 1212-1253.
[12] I. Ekren, N. Touzi and J. Zhang, Viscosity solutions of fully nonlinear parabolic path dependent PDEs: Part II, *Ann. Probab.*, **44** (2016), 2507-2553.
[13] Q. Feng, E. Bayraktar and Z. Zhang, Deep signature algorithm for multidimensional path-dependent options, *SIAM Journal on Financial Mathematics*, **15** (2024).
[14] Q. Feng, M. Luo and Z. Zhang, Deep Signature FBSDE algorithm, *Numerical Algebra, Control and Optimization*, **13** (2023), 500-522.
[15] J.-P. Fouque and Z. Zhang, Deep learning methods for mean field control problems with delay, *Frontiers in Applied Mathematics and Statistics*, **6** (2020), 00011.
[16] C. Gao, S. Gao, R. Hu and Z. Zhu, Convergence of the backward deep bsde method with applications to optimal stopping problems, *SIAM J. Financial Math.*, **14** (2023), 1290-1303.
[17] C. Huré, H. Pham, A. Bachouch and N. Langrené, Deep neural networks algorithms for stochastic control problems on finite horizon, part I: Convergence analysis, *SIAM J. Numer. Anal.*, **59** (2021), 525-557.
[18] C. Huré, H. Pham and X. Warin, Deep Backward schemes for high-dimensional nonlinear PDEs, *Mathematics of Computation*, **89** (2020), 1547-1579.
[19] A. J. Jacquier and M. Oumgari, Deep ppdes for rough local stochastic volatility, Available at SSRN 3400035, 2019.
[20] S. Liao, T. Lyons, W. Yang and H. Ni, Learning stochastic differential equations using rnn with log signature features, preprint, 2019. arXiv:1908.08286.
[21] D. Levin, T. Lyons and H. Ni, Learning from the past, predicting the statistics for the future, learning an evolving system, preprint, 2013. arXiv:1309.0260.
[22] T. Lyons, Differential equations driven by rough signals (i): An extension of an inequality of lc young, *Mathematical Research Letters,* **1** (1994), 451-464.
[23] M. Musiela and M. Rutkowski, *Martingale Methods in Financial Modeling,* Springer, 2005.
[24] S. Peng, Backward stochastic differential equation, nonlinear expectation and their applications, *Proceedings of the International Congress of Mathematicians*, Hyderabad, India, 2010.
[25] S. Peng and F. Wang, Path-dependent PDE and nonlinear Feynman-Kac formula, *Sci. China Math.*, **59** (2016), 19-36.
[26] T. Pham and J. Zhang, Two person zero-sum game in weak formulation and path dependent Bellman-Isaacs equation, *SIAM J. Control Optim.*, **52** (2014), 2090-2121.
[27] J. Ruan, Numerical methods for high-dimensional path-dependent pdes driven by stochastic volterra integral equations, *University of Southern California Dissertations and Theses*, Volume12/etd-RuanJie-8638, 2020.
[28] Z. Ren, N. Touzi and J. Zhang, An overview on viscosity solutions of path-dependent PDEs, *Stochastic Analysis and Applications 2014 - In Honor of Terry Lyons, Springer Proceedings in Mathematics and Statistics*, **100** (2014), 397-453.
[29] M. Sabate-Vidales, D. Siska and L. Szpruch, Solving path dependent pdes with lstm networks and path signatures, preprint, 2020, arXiv:2011.10630.
[30] Y. F. Saporito and Z. Zhang, PDGM: A neural network approach to solve path- dependent partial differential equations, preprint, 2020, arXiv:2003.02035.
[31] H. Wang, H. Chen, A. Sudjianto, R. Liu and Q. Shen, Deep Learning Based BSDE solver for LIBOR market model with application to bermudan swaption pricing and hedging, arXiv:1807.06622
[32] E. Weinan, J. Han and A. Jentzen, Deep learning-based numerical methods for high dimensional parabolic partial differential equations and backward stochastic differential equations, *communications in Mathematics and Statistics*, **5** (2017), 349-380.
[33] Y. Yu, B. Hientzsch and N. Ganesan, Backward deep bsde methods and applications to nonlinear problems, Preprint, 2020, arXiv:2006.07635.
[34] J. Zhang, Backward stochastic differential equations, from linear to fully nonlinear theory, *Probability Theory and Stochastic Modeling*, Springer, 86.

Received January 2024; revised July 2024; early access September 2024.