

A DERIVATIVE-FREE LOCALIZED STOCHASTIC METHOD FOR VERY HIGH DIMENSIONAL SEMILINEAR PARABOLIC PDES*

SHUIXIN FANG[†], CHANGTAO SHENG[‡], BIHAO SU[§], AND TAO ZHOU[¶]

Abstract. We develop a mesh-free, derivative-free, matrix-free, and highly parallel localized stochastic method for high-dimensional semilinear parabolic PDEs. The efficiency of the proposed method is built upon four essential components: (i) a martingale formulation of the forward backward stochastic differential equation (FBSDE); (ii) a small scale stochastic particle method for local linear regression (LLR); (iii) a decoupling strategy with a matrix-free solver for the weighted least-squares system used to compute ∇u ; (iv) a Newton iteration for solving the univariate nonlinear system in u . Unlike traditional deterministic methods that rely on global information, this localized computational scheme not only provides explicit pointwise evaluations of u and ∇u but, more importantly, is naturally suited for parallelization across particles. In addition, the algorithm avoids the need for spatial meshes and global basis functions required by classical deterministic approaches, as well as the derivative-dependent and lengthy training procedures often encountered in machine learning. More importantly, we rigorously analyze the error bound of the proposed scheme, which is fully explicit in both the particle number M and the time step size Δt . Numerical results conducted for problem dimensions ranging from $d = 100$ to $d = 10000$ consistently verify the efficiency and accuracy of the proposed method. Remarkably, all computations are carried out efficiently on a standard personal computer, without requiring any specialized hardware. These results confirm that the proposed method is built upon a principled design that not only extends the practically solvable range of ultra-high-dimensional PDEs but also maintains rigorous error control and ease of implementation.

Key words. High-dimensional PDEs, FBSDEs, Local linear regression, Stochastic particle methods, Error analysis, Parallel computing

MSC codes. 65C30, 65M75, 60H30, 65N15, 68W10

1. Introduction. Partial differential equations (PDEs) in high dimensions constitute a fundamental modeling tool across diverse scientific and engineering disciplines, including quantitative finance, statistical physics, modern control, and learning systems. Typical examples comprise the Schrödinger equation in quantum many-body systems, the Black–Scholes equation in financial mathematics, and Hamilton–Jacobi–Bellman equations (HJB) in control and reinforcement learning [18]. Despite their central role, the numerical treatment of such PDEs faces the notorious *curse of dimensionality* (CoD), where the computational cost grows exponentially with the dimension. Classical deterministic discretization methods based on meshes or global bases, such as finite differences, finite elements, and spectral methods, quickly become infeasible once the dimension exceeds a moderate scale. Sparse grids markedly reduce degrees of freedom versus tensor-product meshes and remain effective up to about $d \approx 10$ for smooth, mildly anisotropic solutions [6, 38, 40]. As d and anisotropy increase, accuracy and conditioning degrade, and complexity grows exponentially in d , which restricts practical use. Thus, deterministic approaches remain fundamentally

*Submitted to the editors DATE.

[†]Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, P. R. China. (sxfang@amss.ac.cn).

[‡]School of Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, China. (ctsheng@sufe.edu.cn).

[§]School of Mathematics and Statistics, Hainan University, Haikou 570100, China. (bihaosu@hainanu.edu.cn).

[¶]Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, P. R. China. (tzhou@lsec.cc.ac.cn).

constrained by CoD when facing genuinely high-dimensional settings.

Deep learning has established itself as a powerful tool for solving PDEs, and in recent years it has demonstrated notable strength in representing high-dimensional functions and mitigating the CoD, thereby emerging as a leading approach for high-dimensional PDEs. Existing methods can be broadly divided into two categories: i). direct learning; and ii). stochastic differential equations (SDE) based learning. Representative direct learning include Physics-Informed Neural Networks (PINN) [33], the Deep Galerkin Method [39], the Deep Ritz method [12], and Weak Adversarial Networks [46]. In these methods, losses are computed at randomly sampled points, enabling efficient parallelization, while automatic differentiation for PDE derivatives remains challenging in very high dimensions, especially for $d \times d$ Hessians. To mitigate this issue, a stochastic-dimension gradient-descent variant of PINNs has been proposed [22] and shows strong potential for ultra-high-dimensional PDEs.

In contrast to direct learning, SDE-based learning recast the problem as a backward stochastic differential equation (BSDE), which makes them inherently *derivative-free*. In pioneering work, Han et al. [11, 18] introduced a deep BSDE framework that parameterizes the solution with neural networks and enforces the equations via residual minimization, solving PDEs in up to 100 dimensions. Related approaches include Deep Splitting and Deep Galerkin, and others (see, e.g., [2, 14, 23, 29, 36, 48] and the references therein). Recently, Cai et al. [7, 9] introduced SOC-MartNet, a martingale-inspired architecture to solve HJB equations without explicit controls, and extended it to ultra-high-dimensional quasilinear parabolic equations, where it demonstrated strong performance on large-scale benchmarks. They later proposed a deep random difference method to reduce variance and improve stability [8]. Despite these advances, several challenges remain: limited stability of the optimization procedure, pronounced sensitivity to hyperparameters, and a lack of rigorous a priori error estimates.

Similar to deep learning, stochastic methods constitute another class of numerical approaches that effectively mitigate the CoD and are widely applied across numerous scientific and engineering fields (see, e.g., [21, 27, 28, 37]). Unlike the black-box nature of deep learning, stochastic methods operate in a more transparent framework, which makes them suitable for error analysis. Their core is a probabilistic representation: the Feynman Kac formula for linear/nonlinear problems and FBSDEs for nonlinear problems, which eliminates explicit derivatives and replaces spatial meshes with Monte Carlo samples and conditional expectations (cf. [17, 26, 31]). Recent advances, such as walk-on-spheres, show clear advantages for anomalous diffusion and other nonlocal effects (see, e.g., [41, 42]), because jump processes accelerate stochastic simulation relative to Brownian motion. As a result, for nonlocal problems with $d \geq 3$, stochastic methods are often more efficient than deterministic approaches. Nonetheless, their strengths lie primarily in high-dimensional linear cases, whereas nonlinear problems remain a substantial challenge (cf. [45]).

Extensive efforts have been made to confront the difficulties introduced by nonlinearities in stochastic algorithms. For example, probabilistic representations based on labeled branching diffusions with Malliavin automatic-differentiation weights absorb nonlinearity into branching, handle the ∇u term, and yield a Monte Carlo-ready random-variable representation (cf. [19]). However, longer horizons or stiff dynamics cause rapid variance growth unless control variates and other variance-reduction techniques are used [13, 19]. Hence, for nonlinear PDEs, probabilistic Monte Carlo methods based on BSDEs are more commonly used. These methods pair path simulation with regression-based estimators of conditional expectations, thereby avoiding spatial meshes, and proceed with a backward scheme to approximate $E_k[\cdot]$ via various

regression methods (cf. [3, 4, 16, 47]). In this way they retain the dimension-agnostic sampling of Monte Carlo and the clean measurability structure induced by filtrations. Nevertheless, accuracy and efficiency remain constrained by the bias-variance trade-off, the expressiveness and conditioning of the approximation spaces, and the distribution of samples in high-dimensional neighborhoods.

In spite of these advances, key gaps remain in stochastic methods for high-dimensional semilinear PDEs: (i) the absence of a mesh-free, fully parallel solver capable of mitigating the CoD and providing dimension-independent comprehensive error analysis; (ii) the lack of efficient and robust strategies to reconstruct u and ∇u from particle ensembles. The aim of this paper is to develop a mesh-free, derivative-free, matrix-free, and highly parallel localized stochastic method for high-dimensional semilinear PDEs, and to provide a rigorous error analysis. The novel contributions of this article to the construction and analysis of stochastic method for semilinear parabolic equation in very high dimensions include the following several aspects.

- **Derivative-free and pointwise local solver:** By casting the semilinear equation (see (2.1)) as a corresponding FBSDE and using a martingale formulation, we rigorously link PDEs to stochastic processes. This connection underpins two key advantages of our stochastic algorithm over traditional deterministic methods. First, it entirely eliminates derivative computations, including gradients and Hessians, which are prohibitively expensive in ultra high dimensions, even for deep neural networks. Second, it transforms global discretization into a genuinely local solver, enabling scalable, pointwise computations that are both simple and naturally parallel.
- **Small-scale local particle method:** We employ Gaussian weights to enhance particle discriminability and select all particles in the ensemble, thereby eliminating the radius tuning required in conventional LLR. This contrasts sharply with k -nearest neighbors (kNN), which in high dimensions tends to induce inflated radii and distance concentration (cf. [1]). In our analysis, the particle number M enters only through an exponentially suppressed bad-event probability e^{-cM} (cf. (3.23)), so a moderate M suffices, and the numerical evidence in Section 4 confirms that $M \approx 100$ already attains accurate results.
- **Decoupled scheme for u and ∇u and a matrix-free solver:** Unlike existing work [16], which relies on Picard iterations to solve the coupled nonlinear system involving u and ∇u and often causes a dramatic increase in computational cost in high dimensions, we adopt a decoupling strategy. Specifically, we first approximate the gradient ∇u via LLR by solving a least-squares problem. The associated $(d+1) \times (d+1)$ linear system is solved in a matrix-free manner, so the storage requirement is $\mathcal{O}(d)$ and the per-time-step cost is only $\mathcal{O}(Md)$, where M denotes the number of particles. Once ∇u is obtained, the remaining univariate nonlinear equation in u can be solved straightforwardly. This design enables efficient handling of problems in very high dimensions.
- **Analyzable computational framework:** Built on an interpretable computational framework, our algorithm admits a rigorous error bound of $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta t e^{-cM})$ (cf. Theorem 3.1), where M denotes the number of particles and Δt the time-step size. This result demonstrates first-order temporal accuracy and requires only the selection of an appropriate number of particles, and these theoretical findings are fully corroborated by numerical experiments.

The rest of the paper is organized as follows. In Section 2, we introduce the standing assumptions and provide a detailed description of the complete stochastic algorithm. Section 3 presents the necessary preparations for the theoretical analysis

and then establishes rigorous convergence results. The numerical aspects are discussed in Section 4, where extensive high-dimensional numerical experiments are evaluated to demonstrate the accuracy, efficiency, and robustness of the proposed method. We conclude in Section 5 with final remarks and an outlook on future research directions.

2. Main algorithm. In this section, we first present the problem together with the associated FBSDEs, and then provide a detailed description of the proposed stochastic algorithm. The procedure begins with employing the martingale formulation for time discretization. Subsequently, a local stochastic particle method combined with a localized reconstruction strategy is introduced, and a Newton iteration is finally applied to resolve the resulting pointwise nonlinear systems.

2.1. Problem setting. Consider the following semilinear parabolic PDE defined on $[0, T] \times \mathbb{R}^d$:

$$(2.1) \quad \begin{cases} (\partial_t + \mathcal{L})u(t, \mathbf{x}) + f(t, \mathbf{x}, u(t, \mathbf{x}), \sigma^\top \nabla u(t, \mathbf{x})) = 0, & (t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d, \\ u(T, \mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^d, \end{cases}$$

where $u : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the unknown scalar function, and $\mathcal{L}$ denotes the infinitesimal generator of the underlying Itô (or Lévy-type) process,

$$\mathcal{L}u(t, \mathbf{x}) = \frac{1}{2} \text{Tr}(\sigma(t, \mathbf{x})\sigma(t, \mathbf{x})^\top \text{Hess}_{\mathbf{x}}u(t, \mathbf{x})) + \langle \mu(t, \mathbf{x}), \nabla u(t, \mathbf{x}) \rangle.$$

Here ∇u and $\text{Hess}_{\mathbf{x}}u$ denote the gradient and the Hessian of u with respect to $\mathbf{x}$, $\sigma : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ is the matrix-valued diffusion coefficient, $\mu : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the vector-valued drift coefficient, $f : [0, T] \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a nonlinear source term, and $g : \mathbb{R}^d \rightarrow \mathbb{R}$ prescribes the terminal condition. In particular, we are often interested in evaluating the solution at the initial time $t = 0$ and spatial location $\mathbf{x} = \xi$ for some $\xi \in \mathbb{R}^d$.

In the semilinear case, u admits an FBSDE characterization, whereas if the nonlinearity depends explicitly on $\nabla^2 u$, one may employ second-order BSDEs (cf. [10]) or adopt local surrogate models for the Hessian. In this work, we focus on a very high-dimensional setting $d \gg 1$ where the nonlinearity f involves only gradient terms. To this end, we introduce the stochastic processes

$$(2.2) \quad Y_t = u(t, X_t), \quad Z_t = \sigma^\top(t, X_t) \nabla u(t, X_t),$$

where the forward process $\{X_t\}_{t \geq 0}$ solves the following SDE

$$(2.3) \quad dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t, \quad X_0 = \mathbf{x},$$

and W_t is a d -dimensional Brownian motion. It then follows that (2.1) is equivalent to the coupled forward-backward system

$$(2.4) \quad \begin{cases} dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t, & X_0 = \mathbf{x}, \\ dY_t = -f(t, X_t, Y_t, Z_t)dt + Z_t^\top dW_t, & Y_T = g(X_T). \end{cases}$$

This FBSDE formulation provides the foundation for probabilistic algorithm, as it allows the original high-dimensional PDE to be reformulated as a system of stochastic equations that can be solved by various discretization techniques for FBSDEs, including more recent works based on deep neural networks (see e.g., [7–9, 18]).

ASSUMPTION 2.1 (Global Lipschitz and linear growth). *Let $\mu : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\sigma : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$, $f : [0, T] \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$, and $g : \mathbb{R}^d \rightarrow \mathbb{R}$. There exist constants $L, C > 0$ such that for all $t \in [0, T]$, $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, $\mathbf{y}, \mathbf{y}' \in \mathbb{R}$, and $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^d$, the following hold*

1. *Global Lipschitz:*

$$\begin{aligned} \|\mu(t, \mathbf{x}) - \mu(t, \mathbf{x}')\| + \|\sigma(t, \mathbf{x}) - \sigma(t, \mathbf{x}')\| &\leq L \|\mathbf{x} - \mathbf{x}'\|, \\ |f(t, \mathbf{x}, \mathbf{y}, \mathbf{z}) - f(t, \mathbf{x}', \mathbf{y}', \mathbf{z}')| &\leq L(\|\mathbf{x} - \mathbf{x}'\| + |\mathbf{y} - \mathbf{y}'| + \|\mathbf{z} - \mathbf{z}'\|), \\ |g(\mathbf{x}) - g(\mathbf{x}')| &\leq L \|\mathbf{x} - \mathbf{x}'\|; \end{aligned}$$

2. *Linear growth:*

$$\|\mu(t, \mathbf{x})\| + \|\sigma(t, \mathbf{x})\| + |f(t, \mathbf{x}, \mathbf{y}, \mathbf{z})| + |g(\mathbf{x})| \leq C(1 + \|\mathbf{x}\| + |\mathbf{y}| + \|\mathbf{z}\|).$$

Here $\|\cdot\|$ denotes the Euclidean norm in the relevant space.

To ensure the well-posedness of this FBSDE formulation, we recall below a classical result under standard Lipschitz and growth conditions (cf. [20]).

LEMMA 2.1. *Suppose Assumption 2.1 holds. We further assume that $\sigma\sigma^\top$ is uniformly nondegenerate, i.e., there exists $\lambda > 0$ such that*

$$\xi^\top (\sigma(t, \mathbf{x})\sigma(t, \mathbf{x})^\top) \xi \geq \lambda \|\xi\|^2, \quad \forall \xi \in \mathbb{R}^d, \quad (t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d,$$

then the FBSDE admits a unique adapted solution $(X, Y, Z) \in \mathcal{S}^2(\mathbb{R}^d) \times \mathcal{S}^2(\mathbb{R}) \times \mathcal{H}^2(\mathbb{R}^d)$, where $\mathcal{S}^2$ denotes the space of square-integrable continuous adapted processes, and $\mathcal{H}^2$ denotes the space of square-integrable predictable processes.

2.2. Time Discretization based on Martingale formulation. We construct a uniform time grid on the interval $[0, T]$ by dividing it into N subintervals of equal length $\Delta t = T/N$, and denote the discrete time nodes by $t_k = k\Delta t$ for $k = 0, 1, \dots, N$. Starting from the backward stochastic differential equation (2.4):

$$dY_t = -f(t, X_t, Y_t, Z_t)dt + Z_t^\top dW_t,$$

together with the representation $Z_t = \sigma^\top(t, X_t)\nabla u(t, X_t)$, we integrate both sides over the subinterval $[t_k, t_{k+1}]$ to obtain

$$Y_{k+1} - Y_k = - \int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s)ds + \int_{t_k}^{t_{k+1}} Z_s^\top dW_s,$$

here $Y_k = Y_{t_k} = u(t_k, X_{t_k})$. Because the dynamics evolve backward in time [49], this relation can be rearranged as

$$Y_k = Y_{k+1} + \int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s)ds - \int_{t_k}^{t_{k+1}} Z_s^\top dW_s.$$

Noting that Y_k is $\mathcal{F}_{t_k}$ -measurable, we introduce the conditional expectation with respect to the filtration $\mathcal{F}_{t_k}$, namely $\mathbb{E}_k[\cdot] := \mathbb{E}[\cdot | \mathcal{F}_{t_k}]$. Using the fact that the Itô integral has zero conditional expectation, i.e., $\mathbb{E}_k[\int_{t_k}^{t_{k+1}} Z_s^\top dW_s] = 0$, we obtain the following recursion:

$$(2.5) \quad Y_k = \mathbb{E}_k \left[Y_{k+1} + \int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s)ds \right].$$

To discretize the integrals over $[t_k, t_{k+1}]$, we apply a first-order Euler–Maruyama approximation by freezing the coefficients at t_k , which gives

$$\int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \approx f(t_k, X_k, Y_k, Z_k) \Delta t, \quad \int_{t_k}^{t_{k+1}} Z_s^\top dW_s \approx Z_k^\top \Delta W_k,$$

where $\Delta W_k := W_{t_{k+1}} - W_{t_k}$ denotes the Brownian increment and $Z_k = Z_{t_k}$. Substituting these approximations into the conditional expectation relation (2.5) and denoting the numerical solution by $\{\tilde{Y}_k\}_{k=0}^N$ yield the semi-discrete backward scheme

$$(2.6) \quad \tilde{Y}_k = \mathbb{E}_k \left[\tilde{Y}_{k+1} + f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k) \Delta t \right], \quad 0 \leq k \leq N.$$

With this foundation, we next focus on solving univariate nonlinear systems involving expectation operators using a local stochastic particle methods.

2.3. Stochastic particle method. This subsection develops a stochastic particle approximation of the conditional expectation in (2.6). For the m -th particle at time t_k , the conditional expectation $\mathbb{E}_k[\cdot]$ is taken with respect to the filtration $\mathcal{F}_{t_k}$ generated by the ensemble of particle positions $\mathcal{S} = \{X_k^1, \dots, X_k^M\}$. More precisely, since $\tilde{Y}_k$ and $\tilde{Z}_k$ are $\mathcal{F}_{t_k}$ -measurable, we approximate, for each particle X_k^m , the conditional expectation in (2.6) by the empirical average over all particles at time t_{k+1} . In practice, one may certainly select a small subset of the nearest particles from $\mathcal{S}$ to perform the regression instead of using all particles. However, since our algorithm uses only a small number of particles (typically $M \leq 100$) and the computation for each particle is fully parallelizable, we using all particles for the regression for notational simplicity.

To this end, we first simulate M independent particle trajectories $\{X_k^j\}_{j=1}^M$ by the Euler–Maruyama discretization of the forward SDE, and denote the numerical solution of j -th particle at time t_k by $\tilde{X}_k^j$:

$$(2.7) \quad \tilde{X}_{k+1}^j = \tilde{X}_k^j + \mu(t_k, \tilde{X}_k^j) \Delta t + \sigma(t_k, \tilde{X}_k^j) \Delta W_k^j, \quad j = 1, 2, \dots, M,$$

where $\Delta W_k^j \sim \mathcal{N}(0, \Delta t I)$ are independent Brownian increments.

Since both $\tilde{Y}_k$ and $\tilde{Z}_k$ are $\mathcal{F}_{t_k}$ -measurable, the solution of discrete scheme (2.6) can, for each particle $\tilde{X}_k^m$, be approximated as

$$(2.8) \quad \begin{aligned} \tilde{Y}_k^m &= \mathbb{E}_k \left[\tilde{Y}_{k+1} | \tilde{X}_k^m \right] + f(t_k, \tilde{X}_k^m, \tilde{Y}_k^m, \tilde{Z}_k^m) \Delta t \\ &\approx \frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j + f(t_k, \tilde{X}_k^m, \tilde{Y}_k^m, \tilde{Z}_k^m) \Delta t, \quad 1 \leq m \leq M, \end{aligned}$$

where the conditional expectation is estimated by a local averaging procedure over those stochastic particles $\{\tilde{X}_k^j\}_{j=1}^M$ whose positions fall within a neighborhood of $\tilde{X}_k^m$. By recursively applying this procedure (2.8) backward in time from $k = N - 1$ to $k = 0$, the approximation of the solution at $t = 0$ is given by the particle average

$$\tilde{Y}_0 = \frac{1}{M} \sum_{m=1}^M \tilde{Y}_0^m.$$

REMARK 2.1. *A salient feature of our method is its sample efficiency: accuracy is attainable with few particles, often with only 100. This accords with Theorem 3.1, where the error bound contains $\Delta t, e^{-cM}$. An appropriate choice of M ensures first-order accuracy. However, for challenging problems, more particles may be needed to maintain accuracy. In such cases, with a suitable ε_k , the method can be viewed as a random batch method (cf. [24]), where reconstruction at each point uses only a fixed, small set of nearest neighbors, keeping the overall computational cost $O(M)$.*

2.4. Computation of $\{Z_k^m\}_{m=1}^M$ via Local Linear Regression. The principal difficulty in efficiently solving (2.8) arises from its structure as a coupled $(d+1)$ -dimensional nonlinear system in the variables $\tilde{Y}_k^m$ and $\tilde{Z}_k^m$. The approach proposed in [16] relies on applying Picard iterations directly to this $(d+1)$ -dimensional system, in conjunction with indicator functions on hypercubes for function reconstruction. While effective in low dimensions, this strategy becomes computationally prohibitive as the dimension increases. To overcome this challenge, we adopt a decoupling strategy: the d -dimensional component $\tilde{Z}_k^m$ is first approximated, after which the resulting univariate nonlinear system in $\tilde{Y}_k^m$ is solved. Therefore, the objective of this subsection is to estimate $Z_k^m = \sigma^\top(t_k, X_k^m) \nabla u(t_k, X_k^m)$ by computing the spatial gradient $\nabla u(t_k, X_k^m)$. To this end, we approximate the function $u(t_{k+1}, \cdot)$ in a neighborhood of $\tilde{X}_k^m$ via a first-order Taylor expansion:

$$(2.9) \quad u(t_{k+1}, \cdot) \approx u(t_k, \tilde{X}_k^m) + \partial_t u(t_k, \tilde{X}_k^m) \Delta t + \nabla u(t_k, \tilde{X}_k^m)^\top (\cdot - \tilde{X}_k^m),$$

where $\cdot$ denotes the spatial variable and the time is fixed at t_{k+1} . To approximate the gradient, we employ a local linear regression using all particles $\{\tilde{X}_k^j\}_{j=1}^M$ within the ε_k -neighborhood of $\tilde{X}_k^m$. It is important to note that, due to the backward-in-time nature of the algorithm, the values $u(t_{k+1}, \tilde{X}_{k+1}^j)$ have already been computed in the previous step, whereas the values at t_k are yet to be updated.

We now present the detailed procedure for estimating the gradient $\nabla u(t_k, \tilde{X}_k^m)$ at time t_k . Since $\tilde{X}_{k+1}^j = \tilde{X}_k^j + \Delta X^j$ and ΔX^j is known, the value $u(t_{k+1}, \tilde{X}_{k+1}^j)$ can be regarded as a function of $\tilde{X}_k^j$. In the fitting process, we directly perform a linear regression in the $\tilde{X}_k$ -space using the pairs $\{(\tilde{X}_k^j, \tilde{Y}_{k+1}^j)\}_{j=1}^M$. To this end, we adopt a local linear approximation centered at the anchor point $\tilde{X}_k^m$. For notational convenience, we set

$$\alpha := u(t_k, \tilde{X}_k^m) + \partial_t u(t_k, \tilde{X}_k^m) \Delta t \in \mathbb{R},$$

and define

$$\alpha_{\mathbf{x}} := \nabla u(t_k, \tilde{X}_k^m) = (\partial_{x_1} u(t_k, \tilde{X}_k^m), \dots, \partial_{x_d} u(t_k, \tilde{X}_k^m))^\top \in \mathbb{R}^d.$$

Then, for each $\tilde{X}_k^m$, the unknown coefficients $\alpha := (\alpha; \alpha_{\mathbf{x}}) \in \mathbb{R}^{d+1}$ in (2.9) are obtained by minimizing the weighted least-squares functional:

$$(2.10) \quad J(\alpha) = \sum_{j=1}^M w_j \left(\tilde{Y}_{k+1}^j - \alpha - \alpha_{\mathbf{x}}^\top (\tilde{X}_k^j - \tilde{X}_k^m) \right)^2, \quad 1 \leq m \leq M,$$

where w_j denotes the weight assigned to each neighbor. When the particle distribution is non-uniform, weighted least squares can significantly reduce estimation variance.

We compute the coefficient vector $\alpha \in \mathbb{R}^{d+1}$ by minimizing the weighted sum of squared residuals, where the weights w_j are determined based on the proximity of each $\tilde{X}_k^j$ to the anchor point $\tilde{X}_k^m$. Specifically, we define

$$(2.11) \quad D_j := \tilde{X}_k^j - \tilde{X}_k^m \in \mathbb{R}^d, \quad w_j := \frac{K\left(\frac{\|D_j\|}{\varepsilon_k}\right)}{\sum_{i=1}^M K\left(\frac{\|D_i\|}{\varepsilon_k}\right)},$$

where K is a given kernel function (e.g., the Gaussian kernel), and $\varepsilon_k > 0$ represents the maximum distance between point $\tilde{X}_k^j$ and $\tilde{X}_k^m$. As a result, the weighted least squares objective (2.10) reads

$$(2.12) \quad J(\alpha) := \sum_{j=1}^M w_j \left(\tilde{Y}_{k+1}^j - \alpha - \alpha_{\mathbf{x}}^\top D_j \right)^2.$$

To minimize the objective functional (2.12) with respect to $\alpha \in \mathbb{R}^{d+1}$, we set its gradient to zero, leading to the normal equations:

$$(2.13) \quad \begin{aligned} \frac{\partial J}{\partial \alpha} &= -2 \sum_{j=1}^M w_j (\tilde{Y}_{k+1}^j - \alpha - \alpha_{\mathbf{x}}^\top D_j) = 0, \\ \frac{\partial J}{\partial \alpha_{\mathbf{x}}} &= -2 \sum_{j=1}^M w_j (\tilde{Y}_{k+1}^j - \alpha - \alpha_{\mathbf{x}}^\top D_j) \cdot D_j = 0. \end{aligned}$$

We now define the design matrix, response vector, and weight matrix as

$$(2.14) \quad \mathbf{D} = \begin{pmatrix} 1 & D_1^\top \\ 1 & D_2^\top \\ \vdots & \vdots \\ 1 & D_M^\top \end{pmatrix} \in \mathbb{R}^{M \times (d+1)}, \quad \mathbf{Y} = \begin{pmatrix} \tilde{Y}_{k+1}^1 \\ \tilde{Y}_{k+1}^2 \\ \vdots \\ \tilde{Y}_{k+1}^M \end{pmatrix} \in \mathbb{R}^M, \quad \mathbf{W} = \begin{pmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_M \end{pmatrix}.$$

With these definitions, the system (2.13) can be rewritten compactly as

$$(2.15) \quad (\mathbf{D}^\top \mathbf{W} \mathbf{D}) \alpha = \mathbf{D}^\top \mathbf{W} \mathbf{Y}.$$

Under the condition $\sum_j w_j D_j = 0$, the weighted least squares problem (2.15) admits a unique solution. In actual computation, we adopt a *matrix-free* strategy: iterative Krylov solvers such as LSQR or preconditioned conjugate gradient (PCG) are applied, where only matrix-vector products with $\mathbf{D}$ and $\mathbf{D}^\top$ are required. This approach reduces the cost to $\mathcal{O}(Md)$ per time step and avoids storing $\mathbf{D}$ or explicitly forming $\mathbf{D}^\top \mathbf{W} \mathbf{D}$. Specifically, for any given vector $\alpha = (\alpha, \alpha_{\mathbf{x}})^\top \in \mathbb{R}^{d+1}$, the matrix-vector products in the left side of (2.15) are computed in two steps as follows

Step 1 Forward product with weights ($\mathbf{W} \mathbf{D} \alpha$): for each $j = 1, \dots, M$,

$$\beta_j := (\mathbf{W} \mathbf{D} \alpha)_j = w_j (\alpha + D_j^\top \alpha_{\mathbf{x}}).$$

Step 2 Transpose product ($\mathbf{D}^\top \beta$): for $\beta \in \mathbb{R}^M$,

$$(\mathbf{D}^\top \beta)_0 = \sum_{j=1}^M \beta_j, \quad (\mathbf{D}^\top \beta)_{1:d} = \sum_{j=1}^M \beta_j D_j.$$

This matrix-free scheme for (2.15) achieves linear complexity in both M and d per time step and is thus particularly suitable for very high-dimensional problems. The vector α is of interest only through its last d components, which correspond to the spatial gradient $\nabla u(t_k, \tilde{X}_k^m)$. The term $\tilde{Z}_k^m$ is then computed as $\tilde{Z}_k^m = \sigma^\top(t_k, \tilde{X}_k^m) \nabla u(t_k, \tilde{X}_k^m)$, whereas the first component of α , denoted by α , is irrelevant to this computation and is therefore discarded.

REMARK 2.2. *In high-dimensional settings, it often occurs that the number of particles $M \ll d$, which renders the normal equations underdetermined or severely ill-conditioned. To address this issue in practical computations, we adopt ridge regression (also known as Tikhonov regularization). Specifically, instead of solving the weighted least-squares problem in its original form, we minimize the penalized functional*

$$J_\lambda(\alpha) = \sum_j w_j (\tilde{Y}_{k+1}^j - \alpha - \alpha^\top D_j)^2 + \lambda \|\alpha\|^2, \quad \lambda > 0,$$

which leads to the regularized solution

$$\alpha = (D^\top W D + \lambda I)^{-1} D^\top W Y.$$

The additional penalty term $\lambda \|\alpha\|^2$ guarantees the invertibility of the system matrix and improves numerical stability, while only introducing a mild bias. This regularization is particularly effective when M is small relative to d , as it balances variance reduction and stability in the estimation of $\tilde{Z}_k^m$.

REMARK 2.3. *Despite the concentration of Euclidean distances as the dimension increases, the LLR step in the proposed method remains efficient, owing to a kernel-based prioritization by relative distance. For the Gaussian kernel $K(u) = e^{-u^2}$,*

$$\frac{w_j}{w_i} = \exp\left(-\frac{\|D_j\|^2 - \|D_i\|^2}{\varepsilon_k^2}\right).$$

Even if the absolute distances $\|D_j\|$ concentrate, the relative gap $|\|D_j\| - \|D_i\||$ still provides discriminative weights that favor nearer neighbors. In addition, since the number of particles M is typically small, one can readily identify enough neighbors in each local region. Consequently, through these complementary mechanisms, the FB-SDE-LLR framework substantially improves the reliability of neighborhood selection and effectively overcomes the inherent limitations of classical LLR methods.

2.5. Computation of $\{\tilde{Y}_k^m\}_{m=1}^M$ via Newton iteration. With both $\tilde{X}_k^m$ and $\tilde{Z}_k^m$ specified, the nonlinear system (2.8) reduces to a one-dimensional equation in $\tilde{Y}_k^m$, which is subsequently solved in the backward update

$$\tilde{Y}_k^m = \frac{1}{M} \sum_j \tilde{Y}_{k+1}^j + f(t_k, \tilde{X}_k^m, \tilde{Y}_k^m, \tilde{Z}_k^m) \Delta t, \quad 1 \leq m \leq M.$$

To this end, we define the following nonlinear function of $\tilde{Y}_k^m$:

$$(2.16) \quad F(\tilde{Y}_k^m) = \tilde{Y}_k^m - \frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j + f(t_k, \tilde{X}_k^m, \tilde{Y}_k^m, \tilde{Z}_k^m) \Delta t,$$

such that the desired solution $\tilde{Y}_k^m$ corresponds to a root of F . To solve the nonlinear equation (2.16), one may employ various numerical solvers. In this work, we adopt

the Newton iteration method, which iteratively updates the solution via

$$(2.17) \quad \tilde{Y}_k^{m,(n+1)} = \tilde{Y}_k^{m,(n)} - \frac{F(\tilde{Y}_k^{m,(n)})}{F'(\tilde{Y}_k^{m,(n)})}, \quad n = 0, 1, 2, \dots, \quad 1 \leq m \leq M,$$

where F' denotes the derivative of F with respect to $\tilde{Y}_k^m$. For clarity, we summarize the complete algorithm as follows.

Algorithm 2.1 FBSDE Solver with Local Linear Regression method for (2.1).

Input: T : terminal time; d : spatial dimension; M : particle count; N : time step count; Δt : time step size; $\mathbf{x}$: target point

- 1: **for** $j = 1 : M$ (**in parallel**) **do**
- 2: Set the terminal condition $Y_N^j = g(X_N^j)$;
- 3: **end for**
- 4: **for** $k = 1 : N$ **do**
- 5: **for** $j = 1 : M$ (**Forward in parallel**) **do**
- 6: Simulate the trajectories of particles $\tilde{X}_k^j$ by (2.7);
- 7: **end for**
- 8: **end for**
- 9: **for** $k = N - 1 : 0$ **do**
- 10: **for** $m = 1 : M$ (**Backward in parallel**) **do**
- 11: Compute $\alpha = (\alpha, \alpha_{\mathbf{x}})^\top$ by a matrix-free solver applied to (2.15).
- 12: Compute $\nabla u \leftarrow \alpha_{\mathbf{x}}$ and $\tilde{Z}_k \leftarrow \sigma^\top \nabla u$;
- 13: Compute $\tilde{Y}_k^m$ by using the Newton method (2.17);
- 14: **end for**
- 15: **end for**
- 16: Calculate the estimated value of $\tilde{Y}_0 = \frac{1}{M} \sum_{m=1}^M \tilde{Y}_0^m$.

Output: The estimated value of the initial value $u(0, \mathbf{x})$;

3. Error estimates. In this section, we first analyze the various sources of error in the computation process and introduce several auxiliary lemmas that will be used in the final error analysis. These include the time discretization error of stochastic differential equations, stochastic matrix estimates associated with linear regression along random paths, and truncation errors from stochastic expansions. Finally, we present a rigorous error analysis tailored to the proposed algorithm.

We recall a classical result on the strong error of the Euler–Maruyama scheme (2.7) for $\tilde{X}_k$ (see [26, Theorem 10.2.2]), which determines its convergence order and forms the basis of our analysis.

LEMMA 3.1. (Strong convergence of forward SDE for $\tilde{X}_k$) *Let X_t be the solution of (2.3), where the coefficients μ and σ satisfy global Lipschitz continuity and linear growth conditions*

$$\begin{aligned} \|\mu(t, \mathbf{x}) - \mu(t, \mathbf{x}')\| &\leq L\|\mathbf{x} - \mathbf{x}'\|, \quad \|\sigma(t, \mathbf{x}) - \sigma(t, \mathbf{x}')\| \leq L\|\mathbf{x} - \mathbf{x}'\|, \\ \|\mu(t, \mathbf{x})\| &\leq K(1 + \|\mathbf{x}\|), \quad \|\sigma(t, \mathbf{x})\| \leq K(1 + \|\mathbf{x}\|). \end{aligned}$$

The continuous-time Euler–Maruyama approximation is then defined for $t \in [t_k, t_{k+1})$ by

$$\tilde{X}_t = X_{t_k} + \mu(t_k, X_{t_k})(t - t_k) + \sigma(t_k, X_{t_k})(W(t) - W(t_k)),$$

Clearly, for $t = t_k$ this reduces to the standard Euler-Maruyama scheme. Moreover, the scheme is known to achieve strong convergence of order $1/2$, in the sense that

$$(3.1) \quad \max_{0 \leq t \leq T} \mathbb{E} [\|X_t - \tilde{X}_t\|^2] \leq C\Delta t, \quad \mathbb{E} \left[\sup_{0 \leq t \leq T} \|X_t - \tilde{X}_t\|^2 \right] \leq C\Delta t,$$

where $C > 0$ is a constant depending only on L, K, T and the initial location $X_0 = \mathbf{x}$.

The following discrete Gronwall Lemma can be found in [5].

LEMMA 3.2. Assume that $\{k_j\}$ ($j \geq 0$) is a given non-negative sequence, and the sequence $\{\varepsilon_n\}$ satisfies $\varepsilon_0 \leq \rho_0$ and

$$(3.2) \quad \varepsilon_n \leq \rho_0 + \sum_{j=0}^{n-1} q_j + \sum_{j=0}^{n-1} k_j \varepsilon_j, \quad n \geq 1,$$

with $\rho_0 \geq 0$, $q_j \geq 0$ ($j \geq 0$). Then

$$(3.3) \quad \varepsilon_n \leq \left(\rho_0 + \sum_{j=0}^{n-1} q_j \right) \exp \left(\sum_{j=0}^{n-1} k_j \right), \quad n \geq 1.$$

We now analyze the time discretization error of the semi-discrete Euler-Maruyama scheme for the Martingale formulation of backward SDE associated with $\tilde{Y}_k$ in (2.6).

LEMMA 3.3. (**Discretization error for Euler scheme (2.6)**) If $f \in C^{1,2}$ and satisfies the Lipschitz condition (2.1), then the local truncation error of semi-discrete backward scheme (2.6) is bounded by

$$(3.4) \quad |\mathcal{E}_k| := \left| \mathbb{E}_k \left[\int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \right] - f(t_k, X_k, Y_k, Z_k) \Delta t \right| \leq C(\Delta t)^2,$$

where C is a positive constant independent of Δt .

Proof. For ease of notation, set $F(t, \mathbf{x}) := f(t, \mathbf{x}, u(t, \mathbf{x}), (\nabla_x u)(t, \mathbf{x}) \sigma(t, \mathbf{x}))$ so that the discretization error (3.4) satisfies

$$\begin{aligned} \mathcal{E}_k &= \mathbb{E}_k \left[\int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \right] - f(t_k, X_k, Y_k, Z_k) \Delta t \\ &= \int_{t_k}^{t_{k+1}} \left(\mathbb{E}_k [F(s, X_s)] - F(t_k, X_k) \right) ds. \end{aligned}$$

which implies

$$(3.5) \quad |\mathcal{E}_k| \leq \int_{t_k}^{t_{k+1}} |\mathbb{E}_k [F(s, X_s)] - F(t_k, X_k)| ds.$$

Applying Itô's formula to $F(t, \mathbf{x})$ yields

$$(3.6) \quad F(t, X_t) = F(t_k, X_k) + \int_{t_k}^t (\partial_t + \mathcal{L})F(s, X_s) ds + \int_{t_k}^t \nabla_x F(s, X_s) \sigma(s, X_s) dW_s,$$

where the generator $\mathcal{L}$ is the one defined in (2.1). Taking conditional expectation and differentiating in t yields

$$(3.7) \quad \frac{d}{dt} \mathbb{E}_k [F(t, X_t)] = \mathbb{E}_k [(\partial_t + \mathcal{L})F(t, X_t)], \quad t \in [t_k, t_{k+1}].$$

Therefore,

$$\sup_{t \in [t_k, t_{k+1}]} \left| \frac{d}{dt} \mathbb{E}_k[F(t, X_t)] \right| \leq M := \sup_{(t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d} |(\partial_t + \mathcal{L})F(t, \mathbf{x})|.$$

By the mean value theorem, for $t \in [t_k, t_{k+1}]$,

$$|\mathbb{E}_k[F(t, X_t)] - F(t_k, X_k)| \leq \sup_{s \in [t_k, t_{k+1}]} \left| \frac{d}{ds} \mathbb{E}_k[F(s, X_s)] \right| (t - t_k) \leq M(t - t_k).$$

Inserting this bound into (3.5) leads to

$$|\mathcal{E}_k| \leq \int_{t_k}^{t_{k+1}} M(t - t_k) dt = \frac{1}{2} M(\Delta t)^2 \leq C(\Delta t)^2,$$

which establishes the claimed estimate. $\square$

To ensure the stability of LLR estimator, it is crucial to establish nondegeneracy conditions for the weighted design matrix. The following two lemmas provide moment bounds and a spectral lower bound for the associated population covariance matrix.

LEMMA 3.4. *Let $D_j := X_k^j - \mathbf{x} \in \mathbb{R}^d$ and define radial weights $w_j := K(\|D_j\|/\varepsilon_k)$, where $K : [0, \infty) \rightarrow [0, \infty)$ is Lipschitz, compactly supported on $[0, 1]$, and there exist constants $0 < \rho \leq 1$ and $K_{\min} > 0$ such that $K(r) \geq K_{\min}$ for all $r \in [0, \rho]$; moreover $K(r) \leq K_{\max}$ for all $r \geq 0$. Assume the sampling density p on $\mathbb{B}_{\varepsilon_k}(\mathbf{x})$ is bounded and positive: $0 < p_0 \leq p(\xi) \leq p_1 < \infty, \forall \xi \in \mathbb{B}_{\varepsilon_k}(\mathbf{x})$. Define the population moments*

$$(3.8) \quad \xi_0 := \mathbb{E}[w_j], \quad \xi_1 := \mathbb{E}[w_j D_j], \quad \Sigma := \mathbb{E}[w_j D_j D_j^\top],$$

where the expectation is taken with respect to the conditional law of D_j , whose density is proportional to $p(\mathbf{x} + \xi) \mathbf{1}_{\{\|\xi\| \leq \varepsilon_k\}}$ restricted to $\mathbb{B}_{\varepsilon_k}(0)$. If, in addition, the sampling is angularly symmetric around $\mathbf{x}$ (i.e., conditional on $\|D_j\| = r$, the direction $D_j/\|D_j\|$ is uniformly distributed on the unit sphere), then the following bounds hold:

$$(3.9) \quad p_0 K_{\min} \text{vol}(\mathbb{B}_{\rho \varepsilon_k}) \leq \xi_0 \leq p_1 K_{\max} \text{vol}(\mathbb{B}_{\varepsilon_k}), \quad \xi_1 = 0,$$

and

$$(3.10) \quad \lambda_{\min}(\Sigma) \geq C_\Sigma \varepsilon_k^{d+2}, \quad C_\Sigma := \frac{\pi^{d/2}}{(d+2)\Gamma(d/2+1)} p_0 K_{\min} \rho^{d+2}.$$

Here $\lambda_{\min}(\Sigma)$ denotes the smallest eigenvalue of the symmetric positive semidefinite matrix Σ , and the volume of a d -dimensional ball of radius r is $\text{vol}(\mathbb{B}_r) = \omega_d r^d/d$, with $\omega_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}$.

Proof. By the definition of ξ_0 , setting $D := \xi - \mathbf{x}$ gives

$$\xi_0 = \int_{\|D\| \leq \varepsilon_k} K\left(\frac{\|D\|}{\varepsilon_k}\right) p(\mathbf{x} + D) dD.$$

For the lower bound in (3.9), we restrict to the region $\|D\| \leq \rho \varepsilon_k$, where $K \geq K_{\min}$ and $p \geq p_0$, which yields

$$\xi_0 \geq p_0 K_{\min} \int_{\|D\| \leq \rho \varepsilon_k} dD = p_0 K_{\min} \text{vol}(\mathbb{B}_{\rho \varepsilon_k}).$$

For the upper bound, using $p \leq p_1$ and $K \leq K_{\max}$ on $\|D\| \leq \varepsilon_k$ gives

$$\xi_0 \leq p_1 K_{\max} \int_{\|D\| \leq \varepsilon_k} dD = p_1 K_{\max} \text{vol}(\mathbb{B}_{\varepsilon_k}).$$

Similarly, by the definition of ξ_1 , we have

$$\xi_1 = \int_{\|D\| \leq \varepsilon_k} K\left(\frac{\|D\|}{\varepsilon_k}\right) p(\mathbf{x} + D) D dD.$$

Under the *angular symmetry* assumption (uniform directions conditional on radius), the angular integral of D over any sphere $\{D : \|D\| = r\}$ is zero, while the weight $K(\|D\|/\varepsilon_k)$ depends only on r . Hence the integral vanishes and (3.9) follows.

It remains to consider Σ , for which we have

$$\Sigma = \int_{\|D\| \leq \varepsilon_k} K\left(\frac{\|D\|}{\varepsilon_k}\right) p(x + D) D D^\top dD \succeq p_0 K_{\min} \int_{\|D\| \leq \rho \varepsilon_k} D D^\top dD,$$

where $\succeq$ denotes the Loewner order on symmetric matrices. Exploiting isotropy of the integral, we obtain

$$\begin{aligned} \int_{\|D\| \leq \rho \varepsilon_k} D D^\top dD &= \frac{1}{d} \left(\int_{\|D\| \leq \rho \varepsilon_k} \|D\|^2 dD \right) I_d = \frac{\omega_d}{d} \left(\int_0^{\rho \varepsilon_k} r^2 \cdot r^{d-1} dr \right) I_d \\ &= \frac{\omega_d}{d(d+2)} (\rho \varepsilon_k)^{d+2} I_d = \frac{\pi^{d/2}}{(d+2)\Gamma(d/2+1)} (\rho \varepsilon_k)^{d+2} I_d, \end{aligned}$$

where $\omega_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}$ and I_d represents $d \times d$ identity matrix. From the above two estimates we obtain $\Sigma \succeq C_\Sigma \varepsilon_k^{d+2} I_d$. Therefore, by the Rayleigh–Ritz characterization, the smallest eigenvalue of the symmetric matrix Σ satisfies $\lambda_{\min}(\Sigma) \geq C_\Sigma \varepsilon_k^{d+2}$, which proves the bound (3.10). $\square$

LEMMA 3.5. Define $\eta_j := w_j D_j D_j^\top \succeq 0$ and $\Gamma = \sum_{j=1}^M \eta_j$, so that $\mathbb{E}[\Gamma] = M\Sigma$. We introduce the event

$$(3.11) \quad \mathcal{A}_k := \left\{ \lambda_{\min}(\Gamma) \geq \frac{1}{2} C_{\mathcal{A}} M \varepsilon_k^{d+2} \right\}, \quad C_{\mathcal{A}} = \frac{\delta^2 C_\Sigma}{2K_{\max}}.$$

There exist constants $C_{\mathcal{A}} > 0$ such that

$$(3.12) \quad \mathbb{P}(\mathcal{A}_k^c) \leq d e^{-C_{\mathcal{A}} M \varepsilon_k^d}.$$

Moreover, on the event $\mathcal{A}_k$ one has

$$(3.13) \quad \|\Gamma^{-1}\| \leq \frac{2}{C_\Sigma M \varepsilon_k^{d+2}}, \quad \|S\| = \left\| \sum_j w_j D_j \right\| \leq K_{\max} M \varepsilon_k^{d+1}, \quad S_0 = \sum_j w_j \asymp M \varepsilon_k^d.$$

where the constant C_Σ is defined in (3.10).

Proof. We derive from (3.8) and (3.10) that

$$(3.14) \quad \lambda_{\min}(\mathbb{E}[\Gamma]) = \lambda_{\min}(M\Sigma) \geq C_\Sigma M \varepsilon_k^{d+2}.$$

Since $w_j \leq K_{\max}$ and $\|D_j\| \leq \varepsilon_k$, each summand $\eta_j = w_j D_j D_j^\top$ satisfies $\lambda_{\max}(\eta_j) = \|\eta_j\| \leq w_j \|D_j\|^2 \leq K_{\max} \varepsilon_k^2$. By the matrix Chernoff bound (cf. [44, Thm. 5.1]), for any $\delta \in (0, 1)$,

$$\mathbb{P}\left\{\lambda_{\min}(\Gamma) \leq (1 - \delta)\lambda_{\min}(\mathbb{E}[\Gamma])\right\} \leq d \exp\left(-\frac{\delta^2}{2} \cdot \frac{\lambda_{\min}(\mathbb{E}[\Gamma])}{K_{\max} \varepsilon_k^2}\right).$$

In particular, inserting (3.14) yields

$$\mathbb{P}\left\{\lambda_{\min}(\Gamma) \leq (1 - \delta)C_\Sigma M \varepsilon_k^{d+2}\right\} \leq d \exp\left(-\frac{\delta^2 C_\Sigma}{2K_{\max}} M \varepsilon_k^d\right) = d \exp(-C_{\mathcal{A}} M \varepsilon_k^d),$$

With $\delta = \frac{1}{2}$, we obtain $\mathbb{P}(\mathcal{A}_k^c) \leq d e^{-C_{\mathcal{A}} M \varepsilon_k^d}$. On $\mathcal{A}_k$, the inverse bound $\lambda_{\min}(\Gamma) \geq \frac{1}{2} C_\Sigma M \varepsilon_k^{d+2}$ holds. Moreover,

$$\|S\| = \left\| \sum_j w_j D_j \right\| \leq \sum_j w_j \|D_j\| \leq K_{\max} M \varepsilon_k^{d+1},$$

and the law of large numbers together with Lemma 3.4 yields $S_0 = \sum_j w_j \asymp M \varepsilon_k^d$. $\square$

We proceed to a rigorous analysis of the error in the Taylor expansion (2.9), where the first-order truncation plays a crucial role by directly linking the known solution $u(t_{k+1}, \cdot)$ (approximated by $\{\tilde{Y}_{k+1}^j\}_j$) with the gradient ∇u , thereby enabling the particle-based LLR construction and yielding the gradient approximation $\alpha_{\mathbf{x}}$.

LEMMA 3.6. (Taylor truncation error (2.9)) *Let $\tilde{X}_k^m = \mathbf{x}$ in (2.9), so that the next point can be written as $\mathbf{x} + D_j$. Let $u(t_{k+1}, \cdot) \in C^2(\mathbb{B}_{\varepsilon_k}(\mathbf{x}))$ with $\|\nabla^2 u(t_{k+1}, \cdot)\|_\infty \leq C_{\nabla^2}$. For each j , introduce the Taylor remainder*

$$(3.15) \quad r_j = u(t_{k+1}, \mathbf{x} + D_j) - u(t_{k+1}, \mathbf{x}) - \partial_t u(t_k, \mathbf{x}) \Delta t - \nabla u(t_{k+1}, \mathbf{x})^\top D_j,$$

Then the following estimate holds

$$(3.16) \quad |r_j| \leq \frac{1}{2} \|\nabla^2 u\|_\infty \|D_j\|^2 \leq \frac{1}{2} C_{\nabla^2} \varepsilon_k^2.$$

Moreover, we have

$$(3.17) \quad \sum_{j=1}^M w_j |r_j| \leq C M \varepsilon_k^{d+2}, \quad \left\| \sum_{j=1}^M w_j r_j D_j \right\| \leq C M \varepsilon_k^{d+3}.$$

Proof. We apply the second-order Taylor expansion of $u(t_{k+1}, \cdot)$ at $\mathbf{x}$ in the direction D_j , which yields

$$u(t_{k+1}, \mathbf{x} + D_j) = u(t_{k+1}, \mathbf{x}) + \partial_t u(t_k, \mathbf{x}) \Delta t + \nabla u(t_{k+1}, \mathbf{x})^\top D_j + r_j,$$

where the remainder takes the integral form

$$r_j = \int_0^1 (1-s) D_j^\top (\nabla^2 u)(t_{k+1}, \mathbf{x} + s D_j) D_j ds.$$

Since $\|D_j\| \leq \varepsilon_k$ and $\|\nabla^2 u\|_\infty \leq C_{\nabla^2}$, it follows that $|r_j| \leq \frac{1}{2} C_{\nabla^2} \varepsilon_k^2$. Consequently, from (3.13) we deduce

$$\sum_j w_j |r_j| \leq \frac{1}{2} C_{\nabla^2} \sum_j w_j \varepsilon_k^2 \leq C \cdot M \varepsilon_k^d \cdot \varepsilon_k^2 = C M \varepsilon_k^{d+2}.$$

Similarly, invoking (3.10), we obtain

$$\begin{aligned} \left\| \sum_j w_j r_j D_j \right\| &\leq \sum_j w_j |r_j| \|D_j\| \\ &\leq \left(\frac{1}{2} C_{\nabla^2} \varepsilon_k^2 \right) \cdot \sum_j w_j \|D_j\| \leq C \varepsilon_k^2 \cdot M \varepsilon_k^{d+1} = C M \varepsilon_k^{d+3}. \end{aligned}$$

This completes the proof. $\square$

A direct analysis of the error between the numerical solution $\alpha_{\mathbf{x}}$ and the gradient ∇u is rather difficult. To address this, we first introduce an auxiliary least-squares solution $\alpha_{\mathbf{x}}^*$ by incorporating the Taylor remainder term r_j , and then establish bounds for the associated Schur complement matrix of the auxiliary problem, thereby preparing the ground for the subsequent analysis of $\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^*$.

LEMMA 3.7. (Bounds for Schur complement matrix) *Let $D_j := X_k^j - \mathbf{x} \in \mathbb{R}^d$ denote local displacements around an anchor $\mathbf{x}$, and let kernel weights be $w_j = K(\|D_j\|/\varepsilon_k)$ with a bounded kernel K supported on $[0, 1]$. Define the weighted moments*

$$S_0 := \sum_{j=1}^M w_j, \quad S := \sum_{j=1}^M w_j D_j, \quad \Gamma := \sum_{j=1}^M w_j D_j D_j^\top.$$

Define the auxiliary noiseless responses

$$(3.18) \quad Y_{k+1}^j := u(t_{k+1}, \mathbf{x} + D_j) = \alpha^* + (\alpha_{\mathbf{x}}^*)^\top D_j + r_j,$$

together with the corresponding weighted least-squares minimizers $(\alpha^, \alpha_{\mathbf{x}}^*)$, where the Taylor remainder r_j is defined in (3.15). Let $(\alpha, \alpha_{\mathbf{x}})$ denote the weighted least-squares minimizers associated with $\{\tilde{Y}_{k+1}^j\}$ as in (2.12). Then their differences, defined as $\delta_{k+1}^j := \tilde{Y}_{k+1}^j - Y_{k+1}^j$, can be expressed as*

$$(3.19) \quad (\Gamma - S S_0^{-1} S^\top)(\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^*) = \sum_{j=1}^M w_j (\delta_{k+1}^j + r_j) D_j - S S_0^{-1} \sum_{j=1}^M w_j (\delta_{k+1}^j + r_j),$$

and

$$(3.20) \quad \alpha - \alpha^* = S_0^{-1} \left(\sum_{j=1}^M w_j (\delta_{k+1}^j + r_j) - S^\top (\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^*) \right).$$

Moreover, there exists a constant $c_{\text{sch}} > 0$ such that, on the event $\mathcal{A}_k$,

$$(3.21) \quad \lambda_{\min}(\Gamma - S S_0^{-1} S^\top) \geq c_{\text{sch}} M \varepsilon_k^{d+2}, \quad \|(\Gamma - S S_0^{-1} S^\top)^{-1}\| \leq \frac{1}{c_{\text{sch}} M \varepsilon_k^{d+2}},$$

and the complement satisfies $\mathbb{P}(\mathcal{A}_k^c) \leq d \exp(-C_{\mathcal{A}} M \varepsilon_k^d)$, where $\mathcal{A}_k$ and $C_{\mathcal{A}}$ are defined in (3.11).

Proof. For clarity, we first rewrite the linear system (2.15) obtained from the weighted least-squares minimizers (2.12), together with its counterpart corresponding to (3.18), into a Schur complement matrix representation

$$\begin{pmatrix} S_0 & S^\top \\ S & \Gamma \end{pmatrix} \begin{pmatrix} \alpha \\ \alpha_{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} \sum_j w_j \tilde{Y}_{k+1}^j \\ \sum_j w_j \tilde{Y}_{k+1}^j D_j \end{pmatrix}, \quad \begin{pmatrix} S_0 & S^\top \\ S & \Gamma \end{pmatrix} \begin{pmatrix} \alpha^* \\ \alpha_{\mathbf{x}}^* \end{pmatrix} = \begin{pmatrix} \sum_j w_j (Y_{k+1}^j - r_j) \\ \sum_j w_j (Y_{k+1}^j - r_j) D_j \end{pmatrix}.$$

Subtracting the two systems yields

$$\begin{pmatrix} S_0 & S^\top \\ S & \Gamma \end{pmatrix} \begin{pmatrix} \alpha - \alpha^\star \\ \alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^\star \end{pmatrix} = \begin{pmatrix} \sum_j w_j (\delta_{k+1}^j + r_j) \\ \sum_j w_j (\delta_{k+1}^j + r_j) D_j \end{pmatrix}.$$

By applying the standard Schur complement procedure, we readily obtain (3.19) and (3.20).

For the spectral bound, observe that for any $v \in \mathbb{R}^d$,

$$v^\top (SS_0^{-1}S^\top)v \leq \frac{\|S\|^2}{S_0} \|v\|^2,$$

which implies

$$\lambda_{\min}(\Gamma - SS_0^{-1}S^\top) \geq \lambda_{\min}(\Gamma) - \frac{\|S\|^2}{S_0}.$$

On the event $\mathcal{A}_k$, we find from (3.13) that

$$\frac{\|S\|^2}{S_0} \leq \frac{K_{\max}^2 M^2 \varepsilon_k^{2d+2}}{cM\varepsilon_k^d} = CM\varepsilon_k^{d+2}.$$

By choosing M sufficiently large (or absorbing constants into C), we may fix $c_{\text{sch}} := \frac{1}{4}C_{\mathcal{A}} > 0$ such that

$$(3.22) \quad \lambda_{\min}(\Gamma - SS_0^{-1}S^\top) \geq \left(\frac{1}{2}C_{\mathcal{A}} - C\right)M\varepsilon_k^{d+2} \geq c_{\text{sch}}M\varepsilon_k^{d+2}.$$

Finally, the corresponding inverse bound follows directly as the reciprocal of this minimal eigenvalue. $\square$

REMARK 3.1. *We note that reusing common randomness across particles may induce weak correlations in $\{\delta_{k+1}^j\}$. Such correlations only modify constants in the variance via an effective-sample-size factor and do not alter the rate in (3.23). For clarity, we adopt the standard i.i.d. assumption in Lemma 3.8; this assumption holds if we draw fresh auxiliary simulations for each particle at every time level.*

The following lemma provides error estimates for the weighted least-squares minimizer $\alpha_{\mathbf{x}}$ (see (2.12)) in comparison with the exact gradient ∇u , which play a central role in the final error analysis of $\tilde{Y}$.

LEMMA 3.8. (Error bound for the gradient estimator ∇u) *Let $\alpha_{\mathbf{x}}$ denote the finite-sample minimizer of (2.12) associated with $\tilde{Y}_{k+1}^j = Y_{k+1}^j + \delta_{k+1}^j$, where δ_{k+1}^j represents the error in $\tilde{Y}_{k+1}^j$. Assume that, conditional on $\mathcal{F}_{t_k}$, the error terms $\{\delta_{k+1}^j\}_j$ are independent and identically distributed. Then it holds that*

$$(3.23) \quad \mathbb{E}_k \left[\|\alpha_{\mathbf{x}} - \nabla u(t_k, \mathbf{x})\|^2 \right] \leq C\varepsilon_k^2 + C\varepsilon_k^{-2} \mathbb{E}_k[|\delta_{k+1}|^2] + Ce^{-C_{\mathcal{A}_k} M \varepsilon_k^d},$$

where C is a positive constant independent of ε_k and M , and $C_{\mathcal{A}_k}$ is defined in (3.11).

Proof. We introduce the **ideal** least-squares solution $\alpha_{\mathbf{x}}^\star$ (cf. (3.18)) corresponding to the noise-free case and decompose the error into bias and variance components:

$$\mathbb{E}_k \left[\|\alpha_{\mathbf{x}} - \nabla u(t_k, \mathbf{x})\|^2 \right] \leq 2\mathbb{E}_k \left[\|\alpha_{\mathbf{x}}^\star - \nabla u(t_k, \mathbf{x})\|^2 \right] + 2\mathbb{E}_k \left[\|\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^\star\|^2 \right].$$

In fact, in the limit $M \rightarrow \infty$, it follows from (3.18) that $\nabla u(t_k, \mathbf{x})$ coincides with the optimal solution of the weighted regression. Hence outside the event $\mathcal{A}_k$ (sufficient sampling within the ε_k -ball), the contribution is negligible. More intuitively, as long as $M\varepsilon_k^d$ is large enough, the probability of the event $\mathcal{A}_k^c$ with a lack of samples in the neighborhood will rapidly decay at the rate of $de^{-C_{\mathcal{A}_k} M\varepsilon_k^d}$. Therefore, when estimating the error, the contribution of this tail event can be safely ignored, and only an additional $de^{-C_{\mathcal{A}_k} M\varepsilon_k^d}$ term needs to be added to cover it. In the following we restrict to $\mathcal{A}_k$, ignoring the exponentially small complement.

We now turn to the estimation of the second term $|\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^*|$. By (3.19), we have

$$\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^* = (\Gamma - SS_0^{-1}S^\top)^{-1} \left(\sum_j w_j(\delta_{k+1}^j + r_j)D_j - SS_0^{-1} \sum_j w_j(\delta_{k+1}^j + r_j) \right).$$

Under the condition that $\{\delta_{k+1}^j\}_j$ are independent and identically distributed, then on $\mathcal{A}_k$, we have

$$\mathbb{E}_k [\delta_{k+1}^j] = \mathbb{E}_k [\delta_{k+1}], \quad \mathbb{E}_k [|\delta_{k+1}^j|^2] = \mathbb{E}_k [|\delta_{k+1}|^2],$$

which implies

$$\mathbb{E}_k \left[\left\| \sum_j w_j(\delta_{k+1}^j + r_j)D_j \right\|^2 \right] \leq \mathbb{E}_k [|\delta_{k+1}|^2] \left(\sum_j w_j D_j \right)^2 + \mathbb{E}_k \left[\left\| \sum_j w_j r_j D_j \right\|^2 \right].$$

Since $w_j \leq K_{\max}$ and $\|D_j\| \leq \varepsilon_k$, we obtain that

$$\sum_j w_j D_j \leq K_{\max} M \varepsilon_k^{d+1}.$$

From the above inequality and (3.17), it follows that

$$\mathbb{E}_k \left[\left\| \sum_j w_j \delta_{k+1}^j D_j \right\|^2 \right] \leq (K_{\max} M)^2 \varepsilon_k^{2d+2} \mathbb{E}_k [|\delta_{k+1}|^2] + M^2 \varepsilon_k^{2d+6}.$$

A similar bound holds for the SS_0^{-1} term,

$$\begin{aligned} \mathbb{E}_k \left[\left\| SS_0^{-1} \sum_j w_j(\delta_{k+1}^j + r_j) \right\|^2 \right] &\leq \mathbb{E}_k [(\delta_{k+1})^2] \left(\varepsilon_k \sum_j w_j \right)^2 + \mathbb{E}_k \left[\left\| \varepsilon_k \sum_j w_j r_j \right\|^2 \right] \\ &\leq (K_{\max} M)^2 \varepsilon_k^{2d+2} \mathbb{E}_k [|\delta_{k+1}|^2] + M^2 \varepsilon_k^{2d+6}. \end{aligned}$$

Using the spectral bound (3.22) for $(\Gamma - SS_0^{-1}S^\top)^{-1}$, we deduce on $\mathcal{A}_k$,

$$\mathbb{E}_k [\|\alpha_{\mathbf{x}} - \alpha_{\mathbf{x}}^*\|^2] \leq \frac{K_{\max}^2}{c_{sch}^2} \varepsilon_k^{-2} \mathbb{E}_k [|\delta_{k+1}|^2] + \frac{K_{\max}^2}{c_{sch}^2} \varepsilon_k^2.$$

On the complement event $\mathcal{A}_k^c$, we employ a crude envelope bound weighted by the exponentially small probability $\mathbb{P}(\mathcal{A}_k^c) \leq de^{-C_{\mathcal{A}_k} M\varepsilon_k^d}$. Taking expectations and combining the results on the events $\mathcal{A}_k$ and $\mathcal{A}_k^c$ then yields (3.23). $\square$

Denote $\text{Var}_k(\cdot) := \text{Var}(\cdot | \mathcal{F}_{t_k})$ as the conditional variance with respect to the filtration at time t_k . Then, we obtain the following conditional variance bound.

LEMMA 3.9. (**Conditional variance bound for Y_{k+1}**) Under Assumption 2.1, then we have

$$(3.24) \quad \text{Var}_k(Y_{k+1}) \leq C \Delta t,$$

where the positive constant C independent of Δt .

Proof. For notational simplicity, set $\tilde{f}(s) := f(s, X_s, Y_s, Z_s)$ in this proof. Recall the BSDE on $[t_k, t_{k+1}]$:

$$Y_{k+1} = Y_k - \int_{t_k}^{t_{k+1}} \tilde{f}(s) ds + \int_{t_k}^{t_{k+1}} Z_s dW_s.$$

Taking conditional expectation with respect to $\mathcal{F}_{t_k}$ on both sides and subtracting, and using $\mathbb{E}_k[\int_{t_k}^{t_{k+1}} Z_s dW_s] = 0$ together with conditional Fubini, we obtain that

$$\begin{aligned} \text{Var}_k(Y_{k+1}) &= \mathbb{E}_k \left[(Y_{k+1} - \mathbb{E}_k[Y_{k+1}])^2 \right] \\ &= \mathbb{E}_k \left[\left(\int_{t_k}^{t_{k+1}} Z_s dW_s - \int_{t_k}^{t_{k+1}} (\tilde{f}(s) - \mathbb{E}_k \tilde{f}(s)) ds \right)^2 \right] \\ &\leq 2 \mathbb{E}_k \left[\left(\int_{t_k}^{t_{k+1}} Z_s dW_s \right)^2 \right] + 2 \mathbb{E}_k \left[\left(\int_{t_k}^{t_{k+1}} (\tilde{f}(s) - \mathbb{E}_k \tilde{f}(s)) ds \right)^2 \right] \\ &= 2 \mathbb{E}_k \int_{t_k}^{t_{k+1}} \|Z_s\|^2 ds + 2 \mathbb{E}_k \left[\left(\int_{t_k}^{t_{k+1}} (\tilde{f}(s) - \mathbb{E}_k \tilde{f}(s)) ds \right)^2 \right] \\ &\leq 2 \mathbb{E}_k \int_{t_k}^{t_{k+1}} \|Z_s\|^2 ds + 2 \Delta t \int_{t_k}^{t_{k+1}} \mathbb{E}_k \left[|\tilde{f}(s) - \mathbb{E}_k \tilde{f}(s)|^2 \right] ds \\ &\leq 2 \mathbb{E}_k \int_{t_k}^{t_{k+1}} \|Z_s\|^2 ds + 2 \Delta t \int_{t_k}^{t_{k+1}} \mathbb{E}_k \left[|\tilde{f}(s)|^2 \right] ds. \end{aligned}$$

From the standard a priori estimate $\sup_{s \leq T} \mathbb{E} \|Z_s\|^2 \leq C$ it follows that the first term is $\leq C \Delta t$, which yields the desired result. $\square$

In fact, the family of numerical solution $\{\tilde{Y}_k^j\}_{j=1}^M$ is conditionally exchangeable given $\mathcal{F}_{t_k}$ rather than independent, since each $\tilde{Y}_k^j$ is formed via partial averaging of given data $\{\tilde{Y}_{k+1}^j\}_{j=1}^M$. The next lemma quantifies the resulting correlation.

LEMMA 3.10. Assume that the particles $\{\tilde{Y}_{k+1}^j\}_{j=1}^M$ are conditionally exchangeable given $\mathcal{F}_{t_k}$. Let

$$(3.25) \quad \bar{\rho}_k := \frac{2}{M(M-1)} \sum_{1 \leq j < \ell \leq M} \text{Corr}_k \left(\tilde{Y}_{k+1}^j, \tilde{Y}_{k+1}^\ell \right), \quad M_{\text{eff}}(k) := \frac{M}{1 + (M-1)\bar{\rho}_k}.$$

Define

$$\xi_{k+1} := \frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j - \mathbb{E}_k[\tilde{Y}_{k+1}],$$

then

$$(3.26) \quad \mathbb{E}_k[|\xi_{k+1}|^2] \leq \frac{C \Delta t}{M_{\text{eff}}(k)} + \frac{C}{M_{\text{eff}}(k)} \mathbb{E}_k[|\tilde{Y}_{k+1} - Y_{k+1}|^2].$$

Proof. Set $\text{Corr}_k(\cdot) := \text{Corr}(\cdot | \mathcal{F}_{t_k})$. Define the centered variables $U_j := \tilde{Y}_{k+1}^j - \mathbb{E}_k[\tilde{Y}_{k+1}^j]$ with $\mathbb{E}_k[U_j] = 0$, and define the conditional pairwise correlations $\rho_{j\ell,k} := \text{Corr}_k(\tilde{Y}_{k+1}^j, \tilde{Y}_{k+1}^\ell)$ for $j \neq \ell$. One can verify easily that

$$\bar{\rho}_k := \frac{2}{M(M-1)} \sum_{1 \leq j < \ell \leq M} \rho_{j\ell,k} \in \left[-\frac{1}{M-1}, 1\right].$$

By the definition of U_j we deduce that

$$\begin{aligned} \text{Var}_k \left(\frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j \right) &= \mathbb{E}_k \left[\left(\frac{1}{M} \sum_{j=1}^M U_j \right)^2 \right] \\ (3.27) \quad &= \frac{1}{M^2} \sum_{j=1}^M \mathbb{E}_k[U_j^2] + \frac{2}{M^2} \sum_{1 \leq j < \ell \leq M} \mathbb{E}_k[U_j U_\ell]. \end{aligned}$$

Conditional exchangeability implies $\mathbb{E}_k[U_j^2] = \text{Var}_k(\tilde{Y}_{k+1})$ for all j , and

$$\mathbb{E}_k[U_j U_\ell] = \text{Cov}_k(\tilde{Y}_{k+1}^j, \tilde{Y}_{k+1}^\ell) = \rho_{j\ell,k} \text{Var}_k(\tilde{Y}_{k+1}), \quad j \neq \ell.$$

Hence

$$\begin{aligned} \text{Var}_k \left(\frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j \right) &= \frac{1}{M^2} \left(M \text{Var}_k(\tilde{Y}_{k+1}) + 2 \text{Var}_k(\tilde{Y}_{k+1}) \sum_{1 \leq j < \ell \leq M} \rho_{j\ell,k} \right) \\ &= \frac{\text{Var}_k(\tilde{Y}_{k+1})}{M^2} \left(M + M(M-1)\bar{\rho}_k \right), \end{aligned}$$

because $\sum_{j < \ell} \rho_{j\ell,k} = \frac{M(M-1)}{2} \bar{\rho}_k$ by the definition of $\bar{\rho}_k$ and covariance decomposition for a correlated mean,

$$(3.28) \quad \text{Var}_k \left(\frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j \right) = \frac{1 + (M-1)\bar{\rho}_k}{M} \text{Var}_k(\tilde{Y}_{k+1}) = \frac{1}{M_{\text{eff}}(k)} \text{Var}_k(\tilde{Y}_{k+1}).$$

Write $\tilde{Y}_{k+1} = Y_{k+1} + (\tilde{Y}_{k+1} - Y_{k+1})$ and apply $(a+b)^2 \leq 2a^2 + 2b^2$ conditionally:

$$(3.29) \quad \text{Var}_k(\tilde{Y}_{k+1}) \leq 2 \text{Var}_k(Y_{k+1}) + 2 \mathbb{E}_k[|\tilde{Y}_{k+1} - Y_{k+1}|^2].$$

In view of (3.27)–(3.29), we obtain that

$$\begin{aligned} \mathbb{E}_k[|\xi_{k+1}|^2] &= \mathbb{E}_k \left[\left(\frac{1}{M} \sum_{j=1}^M U_j \right)^2 \right] = \frac{1}{M_{\text{eff}}(k)} \text{Var}_k(\tilde{Y}_{k+1}) \\ &\leq \frac{2}{M_{\text{eff}}(k)} \left(\text{Var}_k(Y_{k+1}) + \mathbb{E}_k[|\tilde{Y}_{k+1} - Y_{k+1}|^2] \right). \end{aligned}$$

Finally, the above equation and (3.24) directly imply the desired result. $\square$

We are now ready to present the error estimate of the final numerical solution $\tilde{Y}_0$.

THEOREM 3.1. (Global Error) Define the error $\delta_k := \tilde{Y}_k - Y_k$ for $0 \leq k \leq N$. Suppose Assumption 2.1 and the hypotheses of Lemma 3.3 are satisfied. Then, for all sufficiently large M , we have

$$(3.30) \quad \mathbb{E}[|\delta_0|^2] \leq C \Delta t + C \Delta t e^{-c_1 M},$$

where $C > 0$ depends only on T , the Lipschitz constant L , but is independent of Δt , while the constant c_1 depends on ε_k .

Proof. Along the forward particles the exact solution (2.5) satisfies

$$Y_k = \mathbb{E}_k \left[Y_{k+1} + \int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \right],$$

and its implemented time-discrete approximation (2.8) reads

$$\tilde{Y}_k = \frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j + \Delta t f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k).$$

Subtracting the above two relations and incorporating the remainder estimate (3.4), we obtain

$$(3.31) \quad \begin{aligned} \delta_k &= \frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j - \mathbb{E}_k[Y_{k+1}] + f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k) \Delta t - \mathbb{E}_k \left[\int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \right] \\ &= \left(\frac{1}{M} \sum_{j=1}^M \tilde{Y}_{k+1}^j - \mathbb{E}_k[\tilde{Y}_{k+1}] \right) + \left(\mathbb{E}_k[\tilde{Y}_{k+1}] - \mathbb{E}_k[Y_{k+1}] \right) \\ &\quad + \left(f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k) - f(t_k, X_k, Y_k, Z_k) \right) \Delta t \\ &\quad + \left(f(t_k, X_k, Y_k, Z_k) \Delta t - \mathbb{E}_k \left[\int_{t_k}^{t_{k+1}} f(s, X_s, Y_s, Z_s) ds \right] \right) \\ &= \xi_{k+1} + \mathbb{E}_k[\delta_{k+1}] + \left(f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k) - f(t_k, X_k, Y_k, Z_k) \right) \Delta t + \mathcal{E}_k. \end{aligned}$$

Then, the last term in the above equation can be bounded by the Lipschitz condition

$$\begin{aligned} &|f(t_k, X_k, \tilde{Y}_k, \tilde{Z}_k) - f(t_k, X_k, Y_k, Z_k)| \\ &= |f(t_k, X_k, \tilde{Y}_k, \sigma^\top \alpha_{\mathbf{x},k}) - f(t_k, X_k, Y_k, \sigma^\top \nabla u_k)| \leq L(|\delta_k| + \|\sigma\| \cdot \|\alpha_{\mathbf{x},k} - \nabla u_k\|), \end{aligned}$$

which together with (3.31) leads to

$$(1 - L\Delta t)|\delta_k| \leq \mathbb{E}_k[|\delta_{k+1}|] + L\Delta t(\|\sigma\| \cdot \|\alpha_{\mathbf{x},k} - \nabla u_k\|) + |\xi_{k+1}| + |\mathcal{E}_k|.$$

Hence, by conditional Jensen's inequality, $|\mathbb{E}_k[\delta_{k+1}]|^2 \leq \mathbb{E}_k[|\delta_{k+1}|^2]$, together with (3.4), it follows that, for any $\eta > 0$,

$$(3.32) \quad \begin{aligned} (1 - L\Delta t)^2 \mathbb{E}_k[|\delta_k|^2] &\leq (1 + \eta) \mathbb{E}_k[|\delta_{k+1}|^2] + C_\eta (L\Delta t \|\sigma\|)^2 \mathbb{E}_k[\|\alpha_{\mathbf{x},k} - \nabla u_k\|^2] \\ &\quad + \mathbb{E}_k[|\xi_{k+1}|^2] + C(\Delta t)^4, \end{aligned}$$

where C_η is a positive constant depends on η . A combination of (3.32), (3.23), and (3.26) leads to

$$(3.33) \quad \mathbb{E}[|\delta_k|^2] \leq \gamma_k \mathbb{E}[|\delta_{k+1}|^2] + \beta_k,$$

with

$$(3.34) \quad \begin{aligned} \gamma_k &= \frac{(1 + \eta) + C_\eta (L\Delta t \|\sigma\|)^2 \varepsilon_k^{-2} + \frac{C_\eta}{M_{\text{eff}}(k)}}{(1 - L\Delta t)^2}, \\ \beta_k &= \frac{C_\eta}{(1 - L\Delta t)^2} \left((L\Delta t \|\sigma\|)^2 (\varepsilon_k^2 + e^{-c_1 M \varepsilon_k^d}) + (\Delta t)^4 + \frac{\Delta t}{M_{\text{eff}}(k)} \right). \end{aligned}$$

By a Taylor expansion, for sufficiently small Δt we obtain $(1 - L\Delta t)^{-2} \leq 1 + C \Delta t$. When the radius is a constant $\varepsilon_k \in (0, 1]$, so that $\Delta t \varepsilon_k^{-2} = O(\Delta t)$, we set $\eta := L\Delta t$ and applying (3.34) gives

$$\gamma_k \leq (1 + C\Delta t) (1 + C \Delta t \varepsilon_k^{-2}) \leq (1 + C\Delta t)(1 + C\Delta t) \leq 1 + C\Delta t.$$

Otherwise, when the radius is small with $\varepsilon_k \asymp \sqrt{\Delta t}$, we have $\varepsilon_k^{-2} \asymp \Delta t^{-1}$, hence $(L\Delta t)^2 \varepsilon_k^{-2} = L^2 \Delta t = O(\Delta t)$, and to keep γ_k in the form $1 + C\Delta t$ we choose a constant $\eta \in (0, 1]$, whence

$$\gamma_k \leq 1 + C\Delta t.$$

On the other hand, it is known from $\gamma_k \leq 1 + C\Delta t$ that

$$\log \left(\prod_{j=k}^{N-1} \gamma_j \right) = \sum_{j=k}^{N-1} \log(\gamma_j) \leq \sum_{j=k}^{N-1} (\gamma_j - 1) \leq \sum_{j=k}^{N-1} C\Delta t = C(T - t_k),$$

which implies

$$\left(\prod_{j=k}^{N-1} \gamma_j \right) \leq \exp \left(\sum_{j=k}^{N-1} C\Delta t \right) \leq e^{C(T-t_k)}.$$

Using the fact that $\delta_N = 0$, the discrete Gronwall Lemma 3.2 and (3.33) yield

$$\mathbb{E}[|\delta_0|^2] \leq e^{CT} \sum_{k=0}^{N-1} \beta_k \leq C \sum_{k=0}^{N-1} \left((L\Delta t)^2 \varepsilon_k^2 + (L\Delta t \|\sigma\|)^2 e^{-c_1 M \varepsilon_k^d} + (\Delta t)^4 + \frac{\Delta t}{M_{\text{eff}}(k)} \right).$$

Since the last term $\Delta t \sum_{k=0}^{N-1} \frac{1}{M_{\text{eff}}(k)}$ is of order $O(\Delta t)$ for sufficiently large M , the desired bound follows. This ends the proof. $\square$

4. Numerical experiments. In this section, we present several representative numerical experiments in very high dimensions to verify the accuracy, efficiency, and stability of the proposed stochastic algorithm. We employ contrived analytic solutions to demonstrate the temporal convergence rates of the proposed methods. It is worth noting that the test cases cover a range of challenging scenarios, including strong non-linearity, gradient dependence, and problem dimensions up to 10000. All experiments were performed on a personal laptop MacBook Pro (model Z15H000THCH/A), Apple M1 Pro chip (10 cores: 8 performance + 2 efficiency), 32 GB unified memory, macOS system firmware version 10151.140.19.

4.1. Allen-Cahn equation. We first consider the Allen-Cahn equation in high dimensions, a classical reaction-diffusion model in physics that serves as a prototype for phase separation and order-disorder transitions.

$$(4.1) \quad \partial_t u(t, \mathbf{x}) + \Delta u(t, \mathbf{x}) + f(u) = 0, \quad (t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d.$$

In our experiments, we study two cases with different nonlinear terms.

Case 1. Double-well potential $f(u) = u - u^3$ and terminal condition $u(T, \mathbf{x}) = 1/(2 + 0.4\|\mathbf{x}\|^2)$, with $\mathbf{x} \in \mathbb{R}^d$.

Case 2. Logarithmic potential $f(u) = \frac{\theta}{2} \ln(\frac{1+u}{1-u}) - \theta_c u$ with $\theta < \theta_c$ are two positive constants. To facilitate numerical validation, we construct a manufactured solution

$$u(t, \mathbf{x}) = \cos\left(\prod_{j=1}^d x_j\right) e^{\cos t - \|\mathbf{x}\|^2}, \quad \mathbf{x} \in \mathbb{R}^d,$$

by adding an external source term on the right-hand side.

For **Case 1**, we adopt the parameter setting as in [18], with terminal time $T = 0.3$ and spatial dimension $d = 100$. The objective is to evaluate $u(0, \mathbf{x}_0)$ at the initial point $\mathbf{x}_0 = (0, \dots, 0)^\top \in \mathbb{R}^{100}$. The analytic reference value at $\mathbf{x}_0$, obtained by the branching diffusion method and reported in [18], is $u(0, \mathbf{x}_0) \approx 0.0528$. To this end, we apply Algorithm 2.1 to compute numerical solutions, and Figure 1 presents the corresponding absolute and relative errors plotted against Δt on a log-log scale. From Figure 1, we observe that both errors have slopes close to 1 in the log-log plots, which indicates first-order convergence in time. This result is consistent with our theoretical analysis (cf. Theorem 3.1), which establishes that the scheme achieves an $O(\Delta t)$ convergence rate once the bias from the local expansion is sufficiently controlled. We also observe that varying the number of particles M influences the accuracy of the numerical solution but does not alter the convergence rate, again in agreement with our theory in Theorem 3.1. Moreover, we conduct tests with different time steps N (where $\Delta t = T/N$) and particle numbers M , using local expansions together with a Newton solver at each step. When $N = 10^4$ (i.e., $\Delta t = 3 \times 10^{-5}$) and $M = 100$, the absolute error attains a value of about 1.2×10^{-5} . Finally, the scheme demonstrates excellent stability: the explicit-implicit treatment with Newton's method effectively handles the cubic nonlinearity without introducing spurious oscillations, in sharp contrast to naive finite-difference schemes.

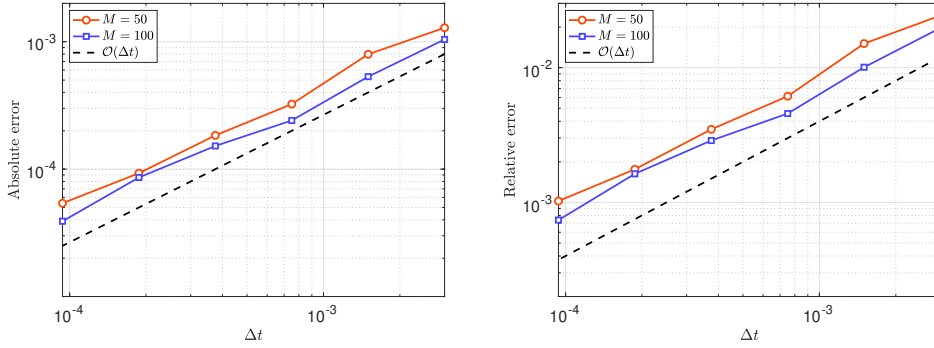


FIG. 1. Numerical error for (1) in Case 1 of the 100-dimensional Allen-Cahn equation at $\mathbf{x} = (0, \dots, 0)$ with $T = 0.3$. The reference value of the exact solution is $u(0, \mathbf{x}) \approx 0.0528$ as reported in [18]. Left: absolute errors; Right: relative errors.

Notably, the method is highly robust to dimensionality and compares favorably with prior methods. Whereas branching diffusion methods (see, e.g., [19]) typically scale as $\mathcal{O}(d^2)$, our scheme is linear in d because each regression is confined to a small neighborhood; even $d = 100$ causes no intrinsic slowdown. Deep BSDE solvers (see, e.g., [15, 23]) can handle the 100-dimensional Allen-Cahn equation but require heavy training, while our linear-regression-plus-Monte Carlo approach attains comparable

accuracy at much lower cost. All error components (time discretization, polynomial approximation bias, and Monte Carlo variance) follow the predicted rates; this confirms the stability and the robustness of the scheme.

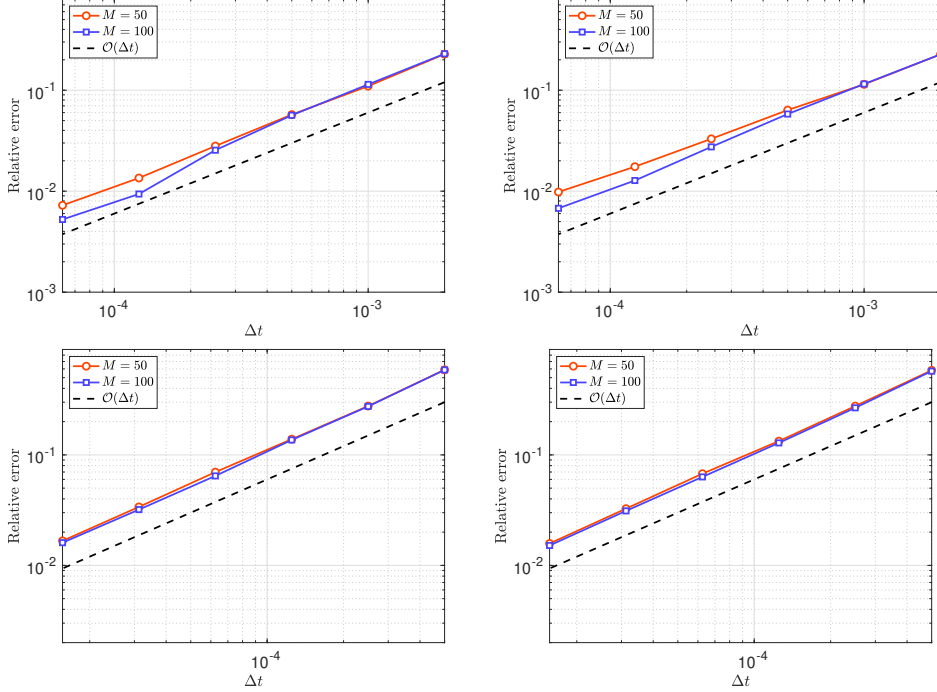


FIG. 2. Relative errors for the Allen-Cahn equation in Case 2 with $T = 1$. Top: $d = 100$; Bottom: $d = 1000$. Left: compute error at $\mathbf{x} = (0, \dots, 0)$; Right: compute error at $\mathbf{x} = (0.1, \dots, 0.1)$.

For **Case 2**, we ran the proposed algorithm with $T = 1$, evaluating the solution at the points $\mathbf{x} = (0, \dots, 0)$ and $\mathbf{x} = (0.1, \dots, 0.1)$, with tests conducted in dimensions $d = 100$ and $d = 1000$. For the case $d = 100$, Figure 2 (top) shows the relative errors in log-log scale as Δt decreases, and the results exhibit first-order convergence. In particular, when $\Delta t < 0.0000625$ (i.e., $N \geq 16000$), the error drops to about 10^{-3} , and further reducing Δt yields a linear decrease. This indicates that the time dependence of various nonlinear terms does not affect the temporal accuracy or convergence rate. Even when $f(t, \mathbf{x}, u)$ is non-smooth and does not satisfy the Lipschitz condition, Newton's method converges rapidly without additional regularization, thereby ensuring both efficiency and robustness. Table 1 shows that the wall-clock runtime grows essentially linearly with $N \cdot M$, consistent with the Monte Carlo complexity. For fixed M , doubling N (where $N = T/\Delta t$) approximately doubles the CPU time. Hence, the cost-accuracy tradeoff can be predicted in a straightforward manner.

We then increased the dimension to $d = 1000$ to assess the scalability of the algorithm. Figure 2 (bottom) shows that, even at this higher dimension, the relative error remains well below 1% once N is sufficiently large. This insensitivity to d highlights the dimension-robustness of the localized regression: each particle explores a random path in $\mathbb{R}^{1000}$, yet at every time step only a local polynomial fit is performed, thereby bypassing the CoD. In contrast, classical regression-based BSDE solvers rely on global basis functions, whose number grows combinatorially with d and quickly becomes ill-conditioned for $d > 200$. Our empirical results demonstrate that the LLR

TABLE 1
Runtime (s) for Allen-Cahn equation of Case 2 with $d = 100$ and $d = 1000$.

$d = 100$	$\Delta t = 0.002$	$\Delta t/2$	$\Delta t/2^2$	$\Delta t/2^3$	$\Delta t/2^4$
$M = 50$	1.24	2.28	4.57	8.83	17.27
$M = 100$	3.81	7.45	14.99	29.55	59.74
$d = 1000$	$\Delta t = 0.0005$	$\Delta t/2$	$\Delta t/2^2$	$\Delta t/2^3$	$\Delta t/2^4$
$M = 50$	36.52	72.32	145.91	288.43	583.69
$M = 100$	129.24	257.59	521.38	1039.55	2082.74

in our method remain well-conditioned and accurately capture the solution even in one thousand dimensions. Moreover, in this example, when both N and M are sufficiently large, the dominant error originates from the local regression bias (see Lemma 3.8) rather than from time stepping or Monte Carlo noise. Overall, Case 2 confirms that the proposed algorithm effectively handles complex nonlinear forcing and scales to very high dimensions with only linear growth in computational cost. Notably, compared with modern deep BSDE approaches, our method attains comparable accuracy with roughly 40% fewer total samples, underscoring the efficiency gained by employing analytic local approximations instead of black-box neural networks.

4.2. Burgers' equation. As a benchmark problem, we next consider the d -dimensional Burgers' equation, a canonical nonlinear model with applications in fluid mechanics, nonlinear acoustics, and traffic flow. It captures both wave-propagation and shock-formation phenomena and, in d spatial dimensions, takes the form

$$(4.2) \quad \frac{\partial u}{\partial t} + \left(u(t, \mathbf{x}) - \frac{2+d}{2} \right) \sum_{i=1}^d \frac{\partial u}{\partial x_i} + \frac{d^2}{2} \nu \Delta u(t, \mathbf{x}) = 0, \quad (t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d,$$

where ν is the Kinematic viscosity ($\nu > 0$ for viscous flow; $\nu = 0$ reduces to the inviscid form).

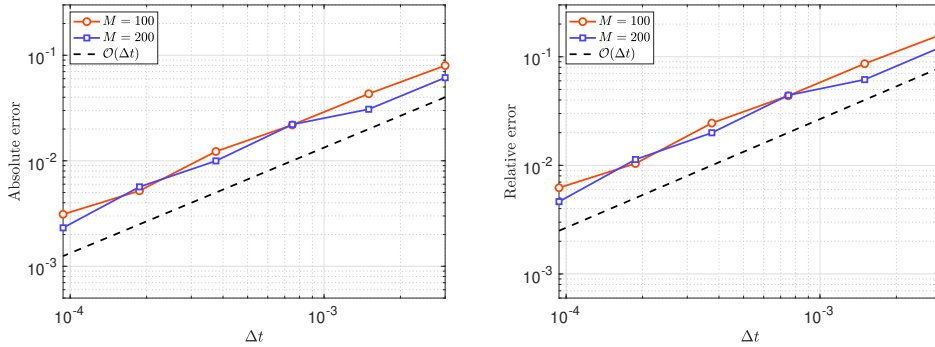


FIG. 3. Numerical error for 10000-dimensional Burger's equation (4.2) at point $\mathbf{x} = (0, 0, \dots, 0)$ with $T = 0.3$. Left: absolute errors; Right: relative errors.

In our simulations, we consider Burgers' equation in spatial dimensions up to $d = 10^4$ and adopt the terminal condition from [18]:

$$u(T, \mathbf{x}) = \frac{\exp(T + \sum_i x_i/d)}{1 + \exp(T + \sum_i x_i/d)},$$

so that at the spatial node $\mathbf{x}_0 = (0, \dots, 0) \in \mathbb{R}^{10000}$ one has $u(0, \mathbf{x}_0) = 0.5$. The results in Figure 3 indicate near first-order convergence in time, i.e., $\mathcal{O}(\Delta t)$. Meanwhile, the proposed scheme remains stable under convective nonlinearity. Unlike finite difference methods that typically require artificial viscosity, our probabilistic approach introduces neither spurious oscillations nor dissipation errors. Moreover, the local polynomial surrogate accurately resolves the solution's sharp gradient structure. Table 2 further reports the CPU runtime of our proposed method for different time step size. The results indicate that the wall-clock time grows essentially linearly with NM , fully consistent with the theoretical Monte Carlo complexity. Compared with deep-learning-based PDE solvers, our approach has the advantage of directly approximating the gradient term through LLR, which is crucial for accurately capturing shock fronts. Overall, these numerical results demonstrate that the proposed algorithm attains high accuracy even for ultra-high-dimensional, strongly nonlinear PDEs and that its computational cost increases only mildly with the dimension d .

TABLE 2
Runtime (s) for 10000d Burgers' equation

$d = 10000$	$\Delta t = 0.003$	$\Delta t/2$	$\Delta t/2^2$	$\Delta t/2^3$	$\Delta t/2^4$
$M = 100$	1040.15	2160.11	4757.37	10593.83	22174.49
$M = 200$	2189.28	4633.74	9674.18	21194.52	45724.85

4.3. Hamilton-Jacobi type equation. Finally, we validate the proposed algorithm on a d -dimensional Hamilton-Jacobi type equation with a gradient dependent sink $R(u, \nabla u) = \kappa u \|\nabla u\|^2$, which enforces self-suppression in regions of large gradient, and the governing equation reads

$$(4.3) \quad \frac{\partial u}{\partial t} + u(t, \mathbf{x}) + f(t, \mathbf{x}, u, \nabla u) = 0, \quad (t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d,$$

where $\kappa = 0.1$ is the reaction coefficient, and the forcing term is given by

$$f(t, \mathbf{x}, u, \nabla u) = \frac{4d}{(1+4t)^{(d+2)/2}} \frac{e^{-\|\mathbf{x}\|^2}}{1+4t} - R(u, \nabla u).$$

Then, the corresponding exact solution is given by

$$u(t, \mathbf{x}) = (1+4t)^{-d/2} \exp\left(-\frac{\|\mathbf{x}\|^2}{1+4t}\right),$$

which spreads rapidly in high dimensions with decaying at rate $\mathcal{O}(t^{-d/2})$ as $t \rightarrow \infty$.

We employ the algorithm to solve (4.3) numerically and evaluate the solution at $\mathbf{x} = (0, \dots, 0) \in \mathbb{R}^d$, with spatial dimensions $d = 500$ and $d = 2000$, and the maximum number of time steps $N = T/\Delta t = 3 \times 10^4$. Figure 4 shows the relative error versus Δt on a log-log scale and indicates a first-order convergence rate. Throughout the simulations, Newton iteration method for the scalar variable Y remains robust and requires only 2-3 iterations per time step, which proves far more efficient than a fully implicit solver for the coupled (Y, Z) system. From Table 3, we observe that the runtime in this example scales almost linearly with N . This near-linear scaling again beats the exponential growth of mesh methods. The use of LLR is central here: we found that using only about 10% of the global polynomial basis points (via LLR)

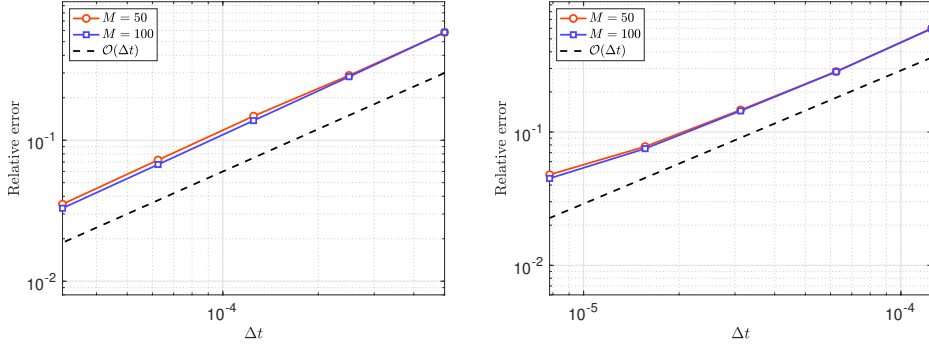


FIG. 4. Relative error of the problem (4.3) against different Δt at point $\mathbf{x} = (0, 0, \dots, 0)$ with $T = 0.5$. Left: $d = 500$; Right: $d = 2000$.

yields the same accuracy, whereas a full global polynomial fit in $d = 2000$ would be hopelessly overfitted or ill-conditioned. Consequently, numerical solution preserves the high-frequency modes of the stiff solution without blowup, while for very stiff, gradient-dominated reactions the proposed method still attains reliable accuracy with only linear work growth.

TABLE 3
Runtime (s) for 500d Hamilton-Jacobi type equation (4.3).

$d = 500$	$\Delta t = 0.0005$	$\Delta t/2$	$\Delta t/2^2$	$\Delta t/2^3$	$\Delta t/2^4$
$M = 50$	89.07	183.69	363.86	740.39	1512.62
$M = 100$	185.06	376.91	765.88	1517.73	3016.35

5. Conclusion. In this paper, we propose a localized and decoupled stochastic algorithm based on FBSDE-LLR that effectively mitigates the CoD for a broad class of semilinear parabolic equations. The key methodological innovation lies in incorporating LLR and a decoupling strategy into the Monte Carlo framework for FBSDEs, specifically through two components: (i) it fits particles within the state space and updates them dynamically, thus capturing fine-scale solution features without global basis functions or neural networks; (ii) it fully decouples the triplet (X, Y, Z) and computes them sequentially in the order $X \rightarrow Z \rightarrow Y$. As a result of these strategies, the algorithm uses only simple linear regression and random sampling, is easy to implement, admits provable convergence, and remains interpretable, and accordingly we present a rigorous error analysis corroborated by extensive numerical experiments. All numerical experiments were conducted on a personal laptop for three representative cases: the Allen-Cahn equation in 100 dimensions, the Burgers' equation in 10000 dimensions, and Hamilton-Jacobi type equation in 2000 dimensions. The results show that the stochastic algorithm is highly efficient and accurate, and its computational cost is essentially linear in both d and M .

At the algorithmic level, the combined strategy demonstrates competitive performance and practical advantages over existing approaches, such as the branching diffusion Monte Carlo method [19] that admits $\mathcal{O}(d^2)$ complexity, regression-based BSDE methods [16] that rely on global bases to solve coupled nonlinear systems, and deep-learning PDE solvers [15, 23, 25, 34] that require extensive training. By con-

trast, the proposed method couples FBSDE sampling with local expansions and a decoupling scheme, achieves comparable or superior accuracy at substantially lower computational cost, and yields a highly scalable, efficient solver for high dimensional nonlinear PDEs that is mesh-free, derivative-free, matrix-free, and highly parallel.

The methodologies and theoretical framework introduced in this work can be further extended to develop efficient stochastic algorithms for ultra-high-dimensional PDEs with strongly nonlinear systems. Potential applications include:

- solving fully nonlinear problems via second-order BSDE formulations [10, 32];
- multi-asset option pricing, high-dimensional stochastic control, and mean-field models [30, 35];
- large-scale filtering and state estimation in engineering systems [43].

We will investigate and report these applications in our future studies.

Acknowledgments. The research of the second author was partially supported by the NSF of China (under grant 12571389). The research of the third author was partially supported by the NSF of China (under grant 12501541). The last author was supported by the NSF of China (under grants 12288201 and 12461160275)

REFERENCES

- [1] C. C. AGGARWAL, A. HINNEBURG, & D. A. KEIM. *On the surprising behavior of distance metrics in high dimensional space*. In International conference on database theory (pp. 420-434). Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
- [2] C. BECK, S. BECKER, P. CHERIDITO, A. JENTZEN, AND A. NEUFELD. *Deep splitting method for parabolic PDEs*. SIAM J. Sci. Comput., 43(5): A3135–A3154, 2021.
- [3] C. BENDER AND R. DENK. *A forward scheme for backward SDEs*. Stochastic Proc. Appl., 117(12): 1793–1812, 2007.
- [4] B. BOUCHARD AND N. TOUZI. *Discrete-time approximation and Monte-Carlo simulation of BSDEs*. Stochastic Proc. Appl., 111(2): 175–206, 2004.
- [5] H. BRUNNER. *Collocation methods for Volterra Integral and Related Functional Equations*, Cambridge University Press, Cambridge, 2004.
- [6] H.-J. BUNGARTZ AND M. GRIEBEL. *Sparse grids*. Acta Numerica, 13: 147–269, 2004.
- [7] W. CAI, S. FANG, AND T. ZHOU. *SOC-MartNet: A Martingale Neural Network for the Hamilton–Jacobi–Bellman Equation Without Explicit in Stochastic Optimal Controls*. SIAM J. Sci. Comput., 47(4): C795–C819, 2025.
- [8] W. CAI, S. FANG, AND T. ZHOU. *Deep random difference method for high dimensional quasilinear parabolic partial differential equations*. arXiv:2506.20308, 2025.
- [9] W. CAI, S. FANG, W. ZHANG, AND T. ZHOU. *Martingale deep learning for very high dimensional quasi-linear partial differential equations and stochastic optimal controls*. arXiv:2408.14395, 2024.
- [10] P. CHERIDITO, H. M. SONER, N. TOUZI, AND N. VICTOIR. *Second-order backward stochastic differential equations and fully nonlinear parabolic PDEs*. Comm. Pure Appl. Math. 60(7): 1081–1110, 2007.
- [11] W. E, J. HAN, AND A. JENTZEN. *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*. Commun. Math. Stat. 5(4): 349–380, 2017.
- [12] W. E AND B. YU. *The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems*. Commun. Math. Stat. 6 (1): 1–12, 2018.
- [13] A. FAHIM, N. TOUZI, AND X. WARIN. *A probabilistic numerical method for fully nonlinear parabolic PDEs*. Ann. Appl. Probab., 21(4): 1322–1364, 2011.
- [14] R. FREY AND V. KÖCK. *Convergence analysis of the deep splitting scheme: the case of partial integro-differential equations and the associated forward backward SDEs with jumps*. SIAM J. Sci. Comput. 47 (1): A527–A552, 2025.
- [15] M. GERMAIN, H. PHAM & X. WARIN. *Approximation error analysis of some deep backward schemes for nonlinear PDEs*. SIAM J. Sci. Comput., 44(1), A28–A56, 2022.
- [16] E. GOBET, J.-P. LEMOR, AND X. WARIN. *A regression-based Monte Carlo method to solve BSDEs*. Ann. Appl. Probab., 15(3): 2172–2202, 2005.
- [17] J. MA AND J. YONG. *Forward–Backward Stochastic Differential Equations and Their Appli-*

- cations. Springer, 1999.
- [18] J. HAN, A. JENTZEN, AND W. E. *Solving high-dimensional partial differential equations using deep learning*. Proc. Natl. Acad. Sci. USA, 115(34): 8505–8510, 2018.
 - [19] P. HENRY-LABORDERE, N. OUDJANE, X. TAN, N. TOUZI, AND X. WARIN. *Branching diffusion representation of semilinear PDEs and Monte Carlo approximation*. Ann. Inst. Henri Poincaré Probab. Stat., 55(1): 184–210, 2019.
 - [20] D. J. HIGHAM AND P. E. KLOEDEN. *Numerical methods for nonlinear stochastic differential equations with jumps*, Numer. Math., 101(1): 101–119, 2005.
 - [21] E. HORTON, A. E. KYPRIANOU, AND D. VILLEMONAIS. *Stochastic methods for the neutron transport equation I: linear semigroup asymptotics*. Ann. Appl. Probab. 30(6): 2573–2612, 2020.
 - [22] Z. HU, K. SHUKLA, G. E. KARNIADAKIS, AND K. KAWAGUCHI. *Tackling the curse of dimensionality with physics-informed neural networks*. Neural Netw., 176: 106369, 2024.
 - [23] J. HURÉ, H. PHAM, AND X. WARIN. *Deep backward schemes for high-dimensional nonlinear PDEs*. Math. Comp., 89(324): 1547–1579, 2020.
 - [24] S. JIN, L. LI, & J. LIU., *Random batch methods (RBM) for interacting particle systems*. J. Comput. Phys., 2020, 400, 108877.
 - [25] L. KAPLLANI & L. TENG. *A backward differential deep learning-based algorithm for solving high-dimensional nonlinear backward stochastic differential equations*. IMA J. Numer. Anal., draf022, 2025.
 - [26] P. E. KLOEDEN AND E. PLATEN. *Numerical solution of stochastic differential equations*, volume 23 of Applications of Mathematics (New York). Springer-Verlag, Berlin, 1992.
 - [27] Z. LEI, S. SHAO, & Y. XIONG., *An efficient stochastic particle method for moderately high-dimensional nonlinear PDEs*. J. Comput. Phys., 2025, 113818.
 - [28] G. LORD, C. POWELL, AND T. SHARDLOW. *An introduction to computational stochastic PDEs*. Cambridge University Press, New York, 2014.
 - [29] L. LU, H. GUO, X. YANG, & Y. ZHU. *Temporal difference learning for high-dimensional PIDEs with jumps*. SIAM J. Sci. Comput. 46(4): C349–C368, 2024.
 - [30] D. ONKEN, L. NURBEKYAN, X. LI, S. FUNG, S. OSHER, L. RUTHOTTO. *A neural network approach for high-dimensional optimal control applied to multiagent path finding*. IEEE Trans. Control Syst. Technol., 31(1): 235–51. 2022.
 - [31] E. PARDOUX AND S. PENG. *Adapted solution of a backward stochastic differential equation*. Syst. Control Lett., 14(1): 55–61, 1990.
 - [32] D. POSSAMAÏ & C. ZHOU. *Second order backward stochastic differential equations with quadratic growth*. Stochastic Process. Appl., 123(10): 3770–3799, 2013.
 - [33] M. RAISSI, P. PERDIKARIS, AND G. E. KARNIADAKIS. *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*. J. Comput. Phys., 378: 686–707, 2019.
 - [34] M. RAISSI.(2024). *Forward-backward stochastic neural networks: deep learning of high-dimensional partial differential equations*. In Peter Carr Gedenkschrift: Research Advances in Mathematical Finance, 637–655, 2024.
 - [35] L. RUTHOTTO, S.J. OSHER, W. LI, L. NURBEKYAN AND S.W. FUNG. *A machine learning framework for solving high-dimensional mean field game and mean field control problems*. Proc. Natl. Acad. Sci. USA, 117(17): 9183–9193, 2020.
 - [36] Y. SAPORITO AND Z. ZHANG. *Path-dependent deep Galerkin method: a neural network approach to solve path-dependent partial differential equations*. SIAM J. Financial Math. 12(3): 912–940, 2021.
 - [37] S. SHAO AND Y. XIONG. *Branching random walk solutions to the Wigner equation*. SIAM J. Numer. Anal. 58 (5): 2589–2608, 2020.
 - [38] J. SHEN AND H. YU. *Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems*. SIAM J. Sci. Comput. 32(6): 3228–3250, 2010.
 - [39] J. SIRIGNANO AND K. SPILIOPOULOS. *DGM: A deep learning algorithm for solving PDEs*. J. Comput. Phys., 375: 1339–1364, 2018.
 - [40] S. A. SMOLYAK. *Quadrature and interpolation formulas for tensor products of certain classes of functions*. Soviet Math. Dokl., 4: 240–243, 1963.
 - [41] C. SHENG, B. SU, AND C. XU. *Efficient Monte Carlo method for integral fractional Laplacian in multiple dimensions*. SIAM J. Numer. Anal., 61(5): 2035–2061, 2023.
 - [42] C. SHENG, B. SU, AND C. XU. *An implicit-explicit Monte Carlo method for semi-linear PDEs driven by α -stable Lévy process and its error estimates*. Math. Comp., 95(357): 263–291, 2026.
 - [43] A. SPANTINI, R. BAPTISTA & Y. MARZOUK. *Coupling techniques for nonlinear ensemble filtering*. SIAM Rev., 64(4), 921–953, 2022.

- [44] J. A. TROPP. *User-friendly tail bounds for sums of random matrices*. Found. Comput. Math., 12(4): 389-434, (2012).
- [45] M. YANG, G. ZHANG, D. DEL-CASTILLO-NEGRETTE, AND Y. CAO. *A probabilistic scheme for semilinear nonlocal diffusion equations with volume constraints*. SIAM J. Numer. Anal. 61(6): 2718-2743, 2023.
- [46] Y. ZANG, G. BAO, X. YE, AND H. ZHOU. *Weak adversarial networks for high-dimensional partial differential equations*. J. Comput. Phys. 411(14): 109409, 2020.
- [47] J. ZHANG. *A numerical scheme for BSDEs*. Ann. Appl. Probab., 14(1): 459-488, 2004.
- [48] W. ZHANG AND W. CAI. *FBSDE based neural network algorithms for high-dimensional quasilinear parabolic PDEs*. J. Comput. Phys. 470: 111557, 14, 2022.
- [49] W. ZHAO, L. CHEN, & S. PENG. *A new kind of accurate numerical method for backward stochastic differential equations*. SIAM J. Sci. Comput., 28(4): 1563-1581, 2006.