

NONLINEAR FILTERING BASED ON DENSITY APPROXIMATION AND DEEP BSDE PREDICTION

KASPER BÅGMARK, ADAM ANDERSSON, AND STIG LARSSON

ABSTRACT. A novel approximate Bayesian filter based on backward stochastic differential equations is introduced. It uses a nonlinear Feynman–Kac representation of the filtering problem and the approximation of an unnormalized filtering density using the well-known deep BSDE method and neural networks. The method is trained offline, which means that it can be applied online with new observations. A hybrid *a priori-a posteriori* error bound is proved under a parabolic Hörmander condition. The theoretical convergence rate is confirmed in two numerical examples.

1. INTRODUCTION

We consider a hybrid system, which couples a state equation in continuous time with measurements in discrete time:

$$(1) \quad S_t = S_0 + \int_0^t \mu(S_s) ds + \int_0^t \sigma(S_s) dB_s, \quad t \in [0, T],$$
$$(2) \quad O_k = h(S_{t_k}) + V_k, \quad k = 1, \dots, K.$$

Here, the process S evolves according to a Stochastic Differential Equation (SDE) with time-independent coefficients μ and σ , driven by Brownian motion B , and with initial distribution π_0 . Observations are made at discrete times t_1, \dots, t_K , with a measurement function h and additive independent noise V_1, \dots, V_K . The filtering problem seeks to compute for every $k \in \{1, \dots, K\}$ the conditional distribution of the latent state S_{t_k} given observations $O_{1:k}$. In other words, the goal is to estimate the conditional density $p(S_{t_k} | O_{1:k})$, which encodes all information available about the system at time t_k . This problem is central to Bayesian inference for dynamical systems and is commonly referred to as Bayesian filtering.

Classical, exact or approximate, solutions to the filtering problem—such as the Kalman filter and its extensions—are effective in low dimensions or when the true solution is close to Gaussian. However, in the nonlinear, non-Gaussian regime, such methods are either biased (e.g., the extended Kalman filter) or suffer from the curse of dimensionality (e.g., particle filters).

This paper introduces a novel approximation scheme for the Bayesian filtering problem with the aim of mitigating the curse of dimensionality. We formulate the filtering problem recursively as a sequence of prediction steps governed by the Fokker–Planck Partial Differential Equation (PDE), intertwined with update steps via Bayes’ formula. By leveraging the connection between parabolic PDEs and Backward Stochastic Differential Equations (BSDE), we obtain a probabilistic representation of the prediction step. The resulting BSDEs are solved numerically by using the deep BSDE method [16], while the update step is implemented in closed form at each observation time t_k . This leads to a recursive approximation scheme.

Our main contributions are as follows:

1. We propose a density-based recursive filter, which is based on sequentially employing the deep BSDE method with observations as input.
2. We prove a hybrid *a priori-a posteriori* error estimates for the method, under a parabolic Hörmander condition, extending the convergence results of [25] to a setting involving a sequence of uncoupled forward-backward systems.

2020 *Mathematics Subject Classification.* 60G25, 60G35, 62F15, 62G07, 62M20, 65C30, 65M75, 68T07.

Key words and phrases. Filtering problem, Fokker–Planck equation, backward stochastic differential equations, numerical analysis, convergence order, Hörmander’s condition, deep learning.

3. We demonstrate the theoretical convergence order in numerical experiments.

The numerical experiments focus on convergence behaviour and are therefore restricted to one-dimensional examples. A complementary numerical study assessing scalability and comparison with classical filtering methods was conducted in [4]. There, the method was shown to scale successfully, including a nonlinear example in dimension 100. The code for the implementation is publicly available¹.

1.1. Bayesian filtering. Bayesian filtering has broad applicability across science and engineering. It plays a foundational role in target tracking [5, 8, 14, 21], robot localization [43], and neural decoding [31]. In finance, it helps to estimate latent variables like volatility [28], and in geosciences, it enables data assimilation through ensemble filters [19]. Computer vision tasks like object tracking can be tackled via particle filters [26]. For multi-sensor navigation and fusion of inertial data, Bayesian filters are a staple [35], and in epidemiology, they support real-time estimation of latent disease spread [27].

Bayesian approaches. When the system dynamics and the observations (1)–(2) are linear and Gaussian, the conditional distribution remains Gaussian and is computed recursively via the Kalman filter [29]. In the nonlinear case, there are several approximation techniques. The extended Kalman filter linearizes the dynamics through Taylor expansion, while the unscented Kalman filter employs sigma-point sampling. These filters approximate the posterior as Gaussian and often perform well in moderate dimensions when the nonlinearity is weak. Beyond Gaussian approximations, the ensemble Kalman filter uses particle ensembles to approximate moments of the posterior, though it still relies on linear-Gaussian observation models. Particle filters (also known as sequential Monte Carlo methods) represent the gold standard in nonlinear filtering, by propagating samples (particles) and updating them via importance weights. Sampling-based methods based on both particle filters and other techniques have met great interest and success in the last few years [10, 11, 12, 20, 36, 39, 44]. However, particle filters for state estimation often scale poorly with state dimension due to sample degeneracy and exponential growth in variance, which are hallmarks of the curse of dimensionality [41, 42].

PDE approaches. Classical filtering methods consist of a prediction step and an update step, where the prediction can be obtained by solving the Fokker–Planck equation. This is explored in [9, 15] among others. However, solving high-dimensional PDEs remains a fundamental challenge in computational mathematics. Classical grid-based methods quickly become infeasible due to the curse of dimensionality. In recent years, several deep learning-based approaches have emerged to tackle this limitation by leveraging probabilistic formulations and neural networks. Among these, physics-informed neural networks [38], the deep Ritz method [17], neural operators [33], and deep Picard iterations [24] offer alternatives for PDE approximation that are less mesh-dependent. However, their performance often degrades in high dimensions, especially when global approximations over the entire domain are required [32, 34]. The challenge intensifies in probabilistic formulations, where the PDE solution corresponds to a probability density. For example, in high dimensions, a Gaussian density becomes sharply concentrated, with values diminishing exponentially in d , resulting in severe numerical underflow and instability.

Two approaches that explicitly target high-dimensional problems, such as the ones we study, are the deep splitting method [7] and the deep BSDE method [16]. Both exploit probabilistic representations of PDEs through Feynman–Kac formulas and use neural networks to approximate solutions. These methods have demonstrated effectiveness in high-dimensional settings, and recent extensions of the deep BSDE method have further improved its robustness [1, 30]. Deep splitting solves a sequence of optimization problems along a time grid, but suffers from error accumulation and challenging training dynamics. In [2, 6, 13], deep splitting methods were applied to the Zakai equation, whose solution is a filtering density in the case of continuous in time observations. Later, in [3], the method was extended to the case of discrete observations, identical to the problem formulation of this paper. In the current paper, we introduce an approximate filter that uses the deep BSDE method for the prediction step.

1.2. Outline of the paper. Section 2 introduces the problem setting, notation, and the derivation of the new filtering method. In Section 3, we present the convergence theorem along with its proof.

¹<https://github.com/bagmark/deep-density-filtering>

Section 4 contains our numerical experiments with empirical convergence results and a discussion of the practical approximations required for the implementation as neural networks. In Section 5 we summarize the results and give a brief outlook on possible extensions. Implementation details are provided in the Appendix.

2. THE BSDE FORMULATION OF THE FILTERING PROBLEM

2.1. Notation. We use a fixed time horizon $T > 0$ and integers $d, d', K \geq 1$ throughout this paper. We denote by $\langle x, y \rangle$ and $\|z\|$ the standard Euclidean inner product and norm when $x, y, z \in \mathbb{R}^d$ and the Frobenius norm when $z \in \mathbb{R}^{d \times d}$. The function space $C^{k,n}([0, T] \times \mathbb{R}^d; \mathbb{R})$ consists of real-valued functions defined on $[0, T] \times \mathbb{R}^d$ that are k times continuously differentiable in time and n times continuously differentiable in space, without requiring the existence of mixed derivatives with respect to t and x . The subspace $C_b^{k,n}([0, T] \times \mathbb{R}^d; \mathbb{R})$ contains all, not necessarily bounded, functions with bounded derivatives. The notation is adapted in obvious ways, when k or n equals ∞ , or when the functions are independent of time. Given a measure space (A, \mathcal{B}, ν) and a Banach space U , for $p \in [1, \infty]$, we use the standard notation for the Bochner space $L^p(A; U)$. For vector fields $V, W \in C^1(\mathbb{R}^d; \mathbb{R}^d)$, we define their Lie bracket as

$$[V, W](x) = DW(x)V(x) - DV(x)W(x), \quad x \in \mathbb{R}^d,$$

where $DV(x)$ denotes the Jacobian matrix of V at x . We say that a collection of vector fields $V_0, \dots, V_n \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^d)$ satisfies the parabolic Hörmander condition if, for every $x \in \mathbb{R}^d$,

$$(3) \quad \text{span} \left\{ V_{j_0}(x), [V_{j_1}(x), V_{j_2}(x)], \left[[V_{j_1}(x), V_{j_2}(x)], V_{j_3}(x) \right], \dots; \right. \\ \left. j_0 \in \{1, \dots, n\}, j_i \in \{0, \dots, n\} \text{ for all } i = 1, \dots, n \right\} = \mathbb{R}^d.$$

Given a function $O: \mathbb{N} \rightarrow \mathbb{R}^{d'}$, we define the $d' \times k$ matrix $O_{1:k} = [O_1, \dots, O_k]$, if $k \geq 1$, and $O_{1:0} = \emptyset$, otherwise. To simplify the notation, we introduce the observation spaces $\mathbb{O}_k := \mathbb{R}^{d' \times k}$ and $\mathbb{O} := \bigcup_{k=1}^K \mathbb{O}_k$.

2.2. Relating the Fokker–Planck equation to BSDEs. Let the quadruple $(\Omega, \mathcal{A}, (\mathcal{F}_t)_{0 \leq t \leq T}, \mathbb{P})$ be a complete filtered probability space and let the filtration $\mathcal{F} := (\mathcal{F}_t)_{0 \leq t \leq T}$ be with respect to two independent d -dimensional Brownian motions B and W . We assume that the initial state S_0 of (1) is distributed according to some probability density π_0 . The observation equation (2) implicitly leads to the likelihood function $L(o, x) := p(O_k = o \mid S_{t_k} = x)$.

In the analysis and derivations of this paper, we require the following two conditions.

- (i) **Smooth boundedness.** The coefficients and functions in (1) and (2) are smooth, with all derivatives of order at least one bounded. More precisely, it holds

$$\mu \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^d), \quad \sigma \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^{d \times d}), \quad \pi_0 \in C_b^\infty(\mathbb{R}^d; \mathbb{R}), \quad L \in C_b^{0,\infty}(\mathbb{R}^{d'} \times \mathbb{R}^d; \mathbb{R}).$$

Moreover, L is uniformly bounded, i.e., there exists a constant $C > 0$ such that

$$(4) \quad \sup_{o \in \mathbb{R}^{d'}} \|L(o)\|_{L^\infty(\mathbb{R}^d; \mathbb{R})} \leq C.$$

- (ii) **Hypoellipticity.** The coefficients μ and σ satisfy the Hörmander condition. More precisely, by writing $\sigma = [\sigma_1, \dots, \sigma_d]$ for the column-wise decomposition and defining the vector fields

$$V_i := \sigma_i, \quad i = 1, \dots, d, \quad V_0 := \mu + \frac{1}{2} \sum_{j=1}^d DV_j V_j,$$

we assume that the family $\{V_0, \dots, V_d\}$ satisfies the parabolic Hörmander condition (3) for all $x \in \mathbb{R}^d$.

The state equation (1) has a unique solution $S = (S_t, t \in [0, T])$ under condition ((i)) [37]. We fix the discrete time points, $0 = t_0 < t_1 < \dots < t_K = T$, uniformly, i.e., $t_k - t_{k-1} = \frac{T}{K}$ for all $k = 1, \dots, K$. The continuous-discrete filtering problem that we consider consists of computing the conditional probability density of S at the time points t_k , $k = 1, \dots, K$, given observations

$o_{1:k} \in \mathbb{O}_k$. In order to introduce the equation for the filtering density, we begin by recalling the generator A associated to S , and its adjoint operator A^* , which are defined for $\varphi \in C^2(\mathbb{R}^d; \mathbb{R})$, by

$$A\varphi = \frac{1}{2} \sum_{i,j=1}^d a_{ij} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} + \sum_{i=1}^d \mu_i \frac{\partial \varphi}{\partial x_i}, \quad A^*\varphi = \frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2}{\partial x_i \partial x_j} (a_{ij} \varphi) - \sum_{i=1}^d \frac{\partial}{\partial x_i} (\mu_i \varphi),$$

where $a = \sigma\sigma^\top \in \mathbb{R}^{d \times d}$. We now state the filtering equations, whose solution is the desired conditional density over time $[0, T]$, yielding the filtering densities at the time points $(t_k)_{k=1}^K$. These equations define functions $p_k \in C([t_k, t_{k+1}] \times \mathbb{R}^d, \mathbb{R})$ sequentially for $k = 0, \dots, K-1$, and are parameterized by \mathbb{O} . We begin, for $x \in \mathbb{R}^d$, by initializing

$$(5) \quad p_0(0, x) = \pi_0(x)$$

and continue by solving, for $k = 0, \dots, K-1$ and $o_{1:k} \in \mathbb{O}_k$,

$$(6) \quad p_k(t, x, o_{1:k}) = p_k(t_k, x, o_{1:k}) + \int_{t_k}^t A^* p_k(s, x, o_{1:k}) ds, \quad t \in (t_k, t_{k+1}],$$

$$(7) \quad p_{k+1}(t_{k+1}, x, o_{1:k+1}) = p_k(t_{k+1}, x, o_{1:k}) L(o_{k+1}, x).$$

The first equation, (6), is the prediction equation, which is a deterministic PDE known as the Fokker–Planck equation. This is followed by the update equation, (7), which is immediately tractable up to a normalizing constant $C = C(o_{1:k+1}) = \left(\int_{\mathbb{R}^d} p_k(t_{k+1}, z, o_{1:k}) L(o_{k+1}, z) dz \right)^{-1}$ that is omitted here. Therefore, by (5)–(7), we obtain the (unnormalized) filtering density. In practical settings, it is sometimes necessary or useful to normalize the density, but it is not necessary for the theoretical results of this paper and is therefore omitted until Section 4.

Next, we quote a regularity result for p from [3, Proposition 2.1]. Throughout the remainder of this section, we assume that conditions ((i)) and ((ii)) hold.

Proposition 2.1. *There exists a unique $p_k \in L^\infty(\mathbb{O}_k; C([t_k, t_{k+1}] \times \mathbb{R}^d, \mathbb{R}))$, $k = 0, \dots, K-1$, satisfying (5)–(7). Moreover, $p_k(o_{1:k}) \in C_b^{1,\infty}([t_k, t_{k+1}] \times \mathbb{R}^d, \mathbb{R})$ for all $k = 0, \dots, K-1$ and $o_{1:k} \in \mathbb{O}_k$.*

In particular, we note that the filtering densities $p_k(t_k, \cdot, o_{1:k})$ belong to $C_b^\infty(\mathbb{R}^d; \mathbb{R})$ for all $k = 1, \dots, K$ and $o_{1:k} \in \mathbb{O}_k$.

Our next result is a nonlinear Feynman–Kac formula for the filtering problem, with a BSDE reformulation of the prediction equation (6). In order to formulate it, we introduce an auxiliary drift $b \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^d)$ and process X , satisfying

$$(8) \quad X_t = X_0 + \int_0^t b(X_s) ds + \int_0^t \sigma(X_s) dW_s, \quad t \in [0, T], \quad \mathbb{P}\text{-a.s.}$$

Here, W is a d -dimensional Brownian motion independent of B and $X_0 \sim \pi_0$. We denote by π_t the unconditional distribution of X at time $t \in [0, T]$. The corresponding generator A_b , is defined, for $\varphi \in C^2(\mathbb{R}^d; \mathbb{R})$, by

$$A_b \varphi = \frac{1}{2} \sum_{i,j=1}^d a_{ij} \frac{\partial^2 \varphi}{\partial x_i \partial x_j} + \sum_{i=1}^d b_i \frac{\partial \varphi}{\partial x_i}.$$

Defining the function $f_b: \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$ by

$$\begin{aligned} f_b(x, u, v) &= \sum_{i=1}^d \left(\sum_{j=1}^d \frac{\partial a_{ij}(x)}{\partial x_j} \right) v_i + \frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2 a_{ij}(x)}{\partial x_i \partial x_j} u - \sum_{i=1}^d \frac{\partial \mu_i(x)}{\partial x_i} u \\ &\quad - \sum_{i=1}^d \mu_i(x) v_i - \sum_{i=1}^d b_i(x) v_i, \end{aligned}$$

we see, for $\varphi \in C^2(\mathbb{R}^d; \mathbb{R})$ and $x \in \mathbb{R}^d$, that

$$(9) \quad (A^* \varphi)(x) = (A_b \varphi)(x) + f_b(x, \varphi(x), \nabla \varphi(x)).$$

We note that the introduction of an auxiliary drift b extends the original deep BSDE scheme from [16], where $b = \mu$ was the only choice. Our extension shares similarities with the approach from [1]. In Section 3 we utilize this freedom by choosing b as

$$(10) \quad b_i = -\mu_i + \sum_{j=1}^d \frac{\partial a_{ij}}{\partial x_j}, \quad i = 1, \dots, d.$$

For this choice of b , the v -dependent terms in f_b cancel. We thus obtain a reduced function $f: \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ defined by

$$(11) \quad f(x, u) = \frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2 a_{ij}(x)}{\partial x_i \partial x_j} u - \sum_{i=1}^d \frac{\partial \mu_i(x)}{\partial x_i} u.$$

The derivation of the nonlinear Feynman–Kac formula is similar to the classical one in [18, Proposition 4.3].

Theorem 2.1 (Nonlinear Feynman–Kac formula for the filtering problem). *Let p_k , $k = 0, \dots, K$, be the solution to (5)–(7). For $k \in \{0, \dots, K-1\}$, and $o_{1:k} \in \mathbb{O}_k$, let (X^k, Y^k, Z^k) be the unique process that satisfies, for $t \in [t_k, t_{k+1}]$, \mathbb{P} -a.s.*

$$(12) \quad \begin{aligned} X_t^k &= X_{t_k}^k + \int_{t_k}^t b(X_s^k) ds + \int_{t_k}^t \sigma(X_s^k) dW_s, \\ Y_t^k &= g_k(X_{t_{k+1}}^k, o_{1:k}) + \int_t^{t_{k+1}} f_b(X_s^k, Y_s^k, Z_s^k) ds - \int_t^{t_{k+1}} (Z_s^k)^\top \sigma(X_s^k) dW_s, \end{aligned}$$

where $X_{t_k}^k \sim \pi_{t_k}$ and the terminal condition g_k is defined by

$$(13) \quad g_k(x, o_{1:k}) = \begin{cases} p_{k-1}(t_k, x, o_{1:k-1}) L(o_k, x), & k \geq 1, \\ \pi_0(x), & k = 0. \end{cases}$$

Then we have, for $t \in [t_k, t_{k+1}]$, \mathbb{P} -a.s.

$$(14) \quad (p_k(t, X_{t_{k+1}+t_k-t}^k, o_{1:k}), \nabla p_k(t, X_{t_{k+1}+t_k-t}^k, o_{1:k})) = (Y_{t_{k+1}+t_k-t}^k, Z_{t_{k+1}+t_k-t}^k).$$

Proof. We fix $k \in \{0, \dots, K\}$, $o_{1:k} \in \mathbb{O}_k$, and omit the o -notation whenever suitable. We introduce the reparameterized solution $q_k(t) = p_k(t_{k+1} + t_k - t)$, for $t \in [t_k, t_{k+1}]$. By (9) and (6)–(7) it follows that q_k , for $t \in [t_k, t_{k+1}]$ and $x \in \mathbb{R}^d$, satisfies

$$\begin{aligned} \frac{\partial}{\partial t} q_k(t, x) + A_b q_k(t, x) &= -f_b(x, q_k(t, x), \nabla q_k(t, x)), \\ q_k(t_{k+1}, x, o_{1:k}) &= g_k(x, o_{1:k}). \end{aligned}$$

By Proposition 2.1, the solution p_k belongs to $C_b^{1,2}([t_k, t_{k+1}] \times \mathbb{R}^d; \mathbb{R})$ and thus so does q_k . We use Itô's formula for $q_k(t, X_t)$, $t \in [t_k, t_{k+1}]$, and $s \in [t_k, t]$, to obtain \mathbb{P} -a.s.

$$\begin{aligned} q_k(t, X_t) &= q_k(s, X_s) + \int_s^t \left(\frac{\partial}{\partial u} q_k(u, X_u) + A_b q_k(u, X_u) \right) du + \int_s^t \nabla q_k(u, X_u)^\top \sigma(X_u) dW_u \\ &= q_k(s, X_s) - \int_s^t f_b(X_u, q_k(u, X_u), \nabla q_k(u, X_u)) du + \int_s^t \nabla q_k(u, X_u)^\top \sigma(X_u) dW_u. \end{aligned}$$

Defining $Y_t = q_k(t, X_t)$ and $Z_t = \nabla q_k(t, X_t)$ we get

$$Y_t = Y_s - \int_s^t f_b(X_u, Y_u, Z_u) du + \int_s^t Z_u^\top \sigma(X_u) dW_u.$$

In particular, with $t = t_{k+1}$ we have

$$Y_{t_{k+1}} = Y_s - \int_s^{t_{k+1}} f_b(X_u, Y_u, Z_u) du + \int_s^{t_{k+1}} Z_u^\top \sigma(X_u) dW_u, \quad s \in [t_k, t_{k+1}].$$

Since Y satisfies the terminal condition $Y_{t_{k+1}} = g_k(X_{t_{k+1}}, o_{1:k})$, we can write the equations for (X, Y, Z) as an uncoupled Forward Backward Stochastic Differential Equation (FBSDE) in the interval $[t_k, t_{k+1}]$,

$$\begin{aligned} X_t &= X_{t_k} + \int_{t_k}^t b(X_s) ds + \int_{t_k}^t \sigma(X_s) dW_s, \\ Y_t &= g_k(X_{t_{k+1}}, o_{1:k}) + \int_t^{t_{k+1}} f_b(X_s, Y_s, Z_s) ds - \int_t^{t_{k+1}} Z_s^\top \sigma(X_s) dW_s. \end{aligned}$$

Under conditions ((i)) and ((ii)), this FBSDE has a unique solution, which we denote (X^k, Y^k, Z^k) and (14) is satisfied by construction, see [18, Proposition 4.3]. \square

We note that, while $X = (X^k)_{k=0}^{K-1}$ is defined in the entire interval $[0, T]$ as a continuous solution, the processes $(Y^k, Z^k)_{k=0}^{K-1}$ are different processes defined in separate intervals. In the following, we denote by $(X^{k,x}, Y^{k,x}, Z^{k,x})$ the process (X^k, Y^k, Z^k) conditioned on $X_{t_k}^k = x$, thus introducing a dependence on x also in (Y^k, Z^k) . By the stochastic representation of p in Theorem 2.1 the following deterministic representation is easily obtained.

Corollary 2.1. *Assume the setting of Theorem 2.1. For $k = 0, \dots, K-1$, $o_{1:k} \in \mathbb{O}_k$, it holds*

$$p_k(t_{k+1}, x, o_{1:k}) = \mathbb{E} \left[g_k(X_{t_{k+1}}^{k,x}, o_{1:k}) + \int_{t_k}^{t_{k+1}} f_b(X_s^{k,x}, Y_s^{k,x}, Z_s^{k,x}) ds \right], \quad x \in \mathbb{R}^d.$$

Proof. We begin by letting $k \in \{0, \dots, K-1\}$, $o_{1:k} \in \mathbb{O}_k$, and $t = t_{k+1}$. By taking the conditional expectation with respect to $X_{t_k}^k = x$ in the first component of (14), we obtain

$$\begin{aligned} p_k(t_{k+1}, x, o_{1:k}) &= \mathbb{E}[Y_{t_k}^k | X_{t_k}^k = x] \\ &= \mathbb{E} \left[g_k(X_{t_{k+1}}^k, o_{1:k}) + \int_{t_k}^{t_{k+1}} f_b(X_s^k, Y_s^k, Z_s^k) ds \middle| X_{t_k}^k = x \right] \\ &= \mathbb{E} \left[g_k(X_{t_{k+1}}^{k,x}, o_{1:k}) + \int_{t_k}^{t_{k+1}} f_b(X_s^{k,x}, Y_s^{k,x}, Z_s^{k,x}) ds \right]. \end{aligned}$$

This completes the proof. \square

2.3. Optimization formulation. The BSDE formulation (12) introduces a new possibility for approximating the prediction step probabilistically. To this end, we reformulate (12) as an equivalent optimization problem. This means that we extend the minimization problem of the original deep BSDE formulation in [16] to also include observations.

Proposition 2.2. *Assume the setting of Theorem 2.1 and let $O: \Omega \rightarrow \mathbb{O}$ be random observations, independent of X , and whose distribution is determined by (2). The solution p_k to (5)–(7), is then given by $p_k(t) = u_k^*(t_{k+1} + t_k - t)$, for $k = 0, \dots, K-1$, $t \in [t_k, t_{k+1}]$, where u_k^* is the solution to the minimization problem*

$$\begin{aligned} &\min_{u \in L^\infty(\mathbb{O}_k; C([t_k, t_{k+1}] \times \mathbb{R}^d; \mathbb{R}))} \mathbb{E} \left[\left| u(t_{k+1}, X_{t_{k+1}}^k, O_{1:k}) - g_k(X_{t_{k+1}}^k, O_{1:k}) \right|^2 \right] \\ &X_t^k = X_{t_k}^k + \int_{t_k}^t b(X_s^k) ds + \int_{t_k}^t \sigma(X_s^k) dW_s, \quad t \in [t_k, t_{k+1}], \\ &u(t, X_t^k, O_{1:k}) = u(t_k, X_{t_k}^k, O_{1:k}) - \int_{t_k}^t f_b(X_s^k, u(s, X_s^k, O_{1:k}), \nabla u(s, X_s^k, O_{1:k})) ds \\ &\quad + \int_{t_k}^t \nabla u(s, X_s^k, O_{1:k})^\top \sigma(X_s^k) dW_s, \quad t \in [t_k, t_{k+1}]. \end{aligned}$$

Proof. Follows directly by substitution and Theorem 2.1. \square

Next, we formulate an equivalent optimization problem by splitting u and ∇u into two separate functions w and v , where $w = u(t_k)$ and $v = \nabla u$ over $[t_k, t_{k+1}]$. As we shall see, this makes the discretization schemes introduced below feasible for a more practical implementation that avoids multiple layers of automatic differentiation. The new optimization problem reads as follows.

$$(15) \quad \begin{aligned} & \min_{\substack{w \in L^\infty(\mathbb{O}_k; C(\mathbb{R}^d; \mathbb{R})) \\ v \in L^\infty(\mathbb{O}_k; C([t_k, t_{k+1}] \times \mathbb{R}^d; \mathbb{R}^d))}} \mathbb{E} \left[\left| Y_{t_{k+1}}^k - g_k(X_{t_{k+1}}^k, O_{1:k}) \right|^2 \right] \\ & X_t^k = X_{t_k}^k + \int_{t_k}^t b(X_s^k) ds + \int_{t_k}^t \sigma(X_s^k) dW_s, \quad t \in [t_k, t_{k+1}], \\ & Y_t^k = w(X_{t_k}^k, O_{1:k}) - \int_{t_k}^t f_b(X_s^k, Y_s^k, v(s, X_s^k, O_{1:k})) ds \\ & \quad + \int_{t_k}^t v(s, X_s^k, O_{1:k})^\top \sigma(X_s^k) dW_s, \quad t \in [t_k, t_{k+1}]. \end{aligned}$$

It is easy to see that, for $k = \{0, \dots, K-1\}$, $(u_k^*(t_k), \nabla u_k^*)$, where u_k^* is the solution from Proposition 2.2, solves (15).

At this stage, we have obtained an optimization problem over both time, space, and observation space. Next, we introduce the Euler–Maruyama approximations \mathcal{X} and \mathcal{Y} of X and Y , respectively. Define a finer time partition $\{t_{k,n}\}_{n=0, k=0}^{N, K-1}$ of the interval $[0, T]$ with

$$t_k = t_{k,0} < \dots < t_{k,n} < \dots < t_{k,N} = t_{k+1}, \quad k = 0, \dots, K-1, \quad n = 0, \dots, N-1.$$

The remaining discretization step of the original deep BSDE method [16] is to fix the time grid for some $N > 0$ and approximate the time-continuous function v with $\bar{v} = (\bar{v}_n)_{n=0}^{N-1}$, $\bar{v}_i: \mathbb{R}^d \times \mathbb{O}_k \rightarrow \mathbb{R}^d$, defined at the discrete time points $(t_{k,n})_{n=0}^{N-1}$, $k = 0, \dots, K-1$, such that $\bar{v}_n \approx v(t_{k,n})$. We obtain the following discrete problem

$$(16) \quad \begin{aligned} & \min_{\substack{w \in L^\infty(\mathbb{O}_k; C(\mathbb{R}^d; \mathbb{R})) \\ (\bar{v}_n)_{n=0}^{N-1} \in \prod_{n=0}^{N-1} L^\infty(\mathbb{O}_k; C(\mathbb{R}^d; \mathbb{R}^d))}} \mathbb{E} \left[\left| \mathcal{Y}_N^k - \bar{g}_k(\mathcal{X}_N^k, O_{1:k}) \right|^2 \right] \\ & \mathcal{X}_{n+1}^k = \mathcal{X}_n^k + b(\mathcal{X}_n^k)(t_{k,n+1} - t_{k,n}) + \sigma(\mathcal{X}_n^k)(W_{t_{k,n+1}} - W_{t_{k,n}}), \quad n = 0, \dots, N-1, \\ & \mathcal{Z}_n^k = \bar{v}_n(\mathcal{X}_n^k, O_{1:k}), \quad n = 0, \dots, N-1, \\ & \mathcal{Y}_{n+1}^k = w(\mathcal{X}_0^k, O_{1:k}) - \sum_{\ell=0}^n \left(f_b(\mathcal{X}_\ell^k, \mathcal{Y}_\ell^k, \mathcal{Z}_\ell^k)(t_{k,\ell+1} - t_{k,\ell}) \right. \\ & \quad \left. - (\mathcal{Z}_\ell^k)^\top \sigma(\mathcal{X}_\ell^k)(W_{t_{k,\ell+1}} - W_{t_{k,\ell}}) \right), \quad n = 0, \dots, N-1, \end{aligned}$$

where \bar{g}_k is defined, for $x \in \mathbb{R}^d$ and $o_{1:k} \in \mathbb{O}_k$, from the previous optimum w_{k-1}^*

$$(17) \quad \bar{g}_k(x, o_{1:k}) = \begin{cases} w_{k-1}^*(x, o_{1:k-1})L(o_k, x), & k \geq 1, \\ \pi_0(x), & k = 0. \end{cases}$$

We define our filter approximation $\hat{p} = (\hat{p}_k)_{k=1}^K$, of the exact filter $(p_k(t_k))_{k=1}^K$, by

$$(18) \quad \hat{p}_k(x, o_{1:k}) = w_{k-1}^*(x, o_{1:k-1})L(o_k, x), \quad o_{1:k} \in \mathbb{O}_k, \quad x \in \mathbb{R}^d,$$

where $(w_k^*)_{k=0}^{K-1}$ are the solutions obtained from the discrete formulation (16). See Figure 1 for a flowchart of the methodology to obtain \hat{p} .

3. ERROR ANALYSIS

In this section, we quantify how discretization and optimization errors propagate through the recursive filtering scheme by studying a strong error. To simplify the notation, we fix the time grid to satisfy, for all $k = 0, \dots, K-1$, $n = 0, \dots, N-1$, $t_{k,n+1} - t_{k,n} = \frac{T}{KN} =: \tau$. We begin Section 3.1 by reciting some results that we need. In Section 3.2 we state the main result and give the corresponding proof. Throughout the section, we assume (i) and (ii).

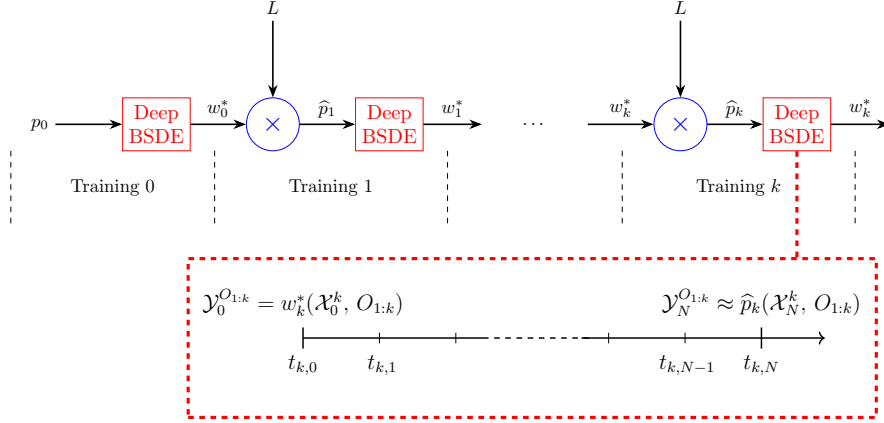


Figure 1. The figure illustrates the flowchart of the chain of prediction and update steps throughout the method. Each Deep BSDE box represents a BSDE which is solved through the optimization problem (16).

3.1. Relevant inequalities. Let (X, Y, Z) be the solution to (12) and $(\tilde{X}, \tilde{Y}, \tilde{Z}) = (\tilde{X}^k, \tilde{Y}^k, \tilde{Z}^k)_{k=0}^{K-1}$ be the solution, for $k \in \{0, \dots, K-1\}$, $o_{1:k} \in \mathcal{O}_k$, $t \in [t_k, t_{k+1}]$, to

$$(19) \quad \begin{aligned} \tilde{X}_t^k &= \tilde{X}_{t_k}^k + \int_{t_k}^t b(\tilde{X}_s^k) ds + \int_{t_k}^t \sigma(\tilde{X}_s^k) dW_s, \\ \tilde{Y}_t^k &= \bar{g}_k(\tilde{X}_{t_{k+1}}^k, o_{1:k}) + \int_t^{t_{k+1}} f_b(\tilde{X}_s^k, \tilde{Y}_s^k, \tilde{Z}_s^k) ds - \int_t^{t_{k+1}} (\tilde{Z}_s^k)^\top \sigma(\tilde{X}_s^k) dW_s. \end{aligned}$$

This system of equations is the same as (12) except for the terminal conditions of the \tilde{Y}^k processes and we note that $X = \tilde{X}$. By the *a priori* estimate [18, Proposition 2.1], there exists a constant $C_1 > 0$ so that

$$(20) \quad \sup_{t \in [t_k, t_{k+1}]} \mathbb{E} \left[|Y_t^k - \tilde{Y}_t^k|^2 \right]^{\frac{1}{2}} \leq C_1 \mathbb{E} \left[|g_k(X_{t_{k+1}}^k, o_{1:k}) - \bar{g}_k(X_{t_{k+1}}^k, o_{1:k})|^2 \right]^{\frac{1}{2}}.$$

We remark that in [18] the Z^k -component (and identically \tilde{Z}^k) is defined such that the diffusion coefficient $\sigma(X^k)$ is absorbed into Z^k . Consequently, the corresponding BSDE in [18] is written in a form that differs from (19) by this convention. When adapting the *a priori* estimates from [18] to our setting, the process denoted by Z^\top in [18] should therefore be identified with $(Z^k)^\top \sigma(X^k)$ in our notation.

In [25] the authors study the convergence of the deep BSDE method for more general coupled FBSDEs. The most restrictive condition in [25] is a weak coupling condition. This condition is trivially satisfied in our uncoupled framework and more generally it is easy to verify that conditions (i) and (ii) are sufficient for [25, Theorem 1]. The result states that, up to a constant, the approximation error is bounded by the uniform time step τ and the error at the terminal condition, thus a hybrid *a priori* and *a posteriori* error bound. To state the result, applied to our setting, we introduce some notation. For $k \in \{0, \dots, K-1\}$ we define the piecewise constant processes

$$(\mathcal{X}_t^k, \mathcal{Y}_t^k, \mathcal{Z}_t^k) = (\mathcal{X}_n^k, \mathcal{Y}_n^k, \mathcal{Z}_n^k), \quad t \in [t_{k,n}, t_{k,n+1}), \quad n \in \{0, \dots, N-1\},$$

where $(\mathcal{X}^k, \mathcal{Y}^k, \mathcal{Z}^k)$ are the solutions to (16). According to [25, Theorem 1], there exists a constant C independent of τ such that

$$(21) \quad \sup_{t \in [t_k, t_{k+1}]} \left(\mathbb{E} \|\tilde{X}_t^k - \mathcal{X}_t^k\|^2 + \mathbb{E} |\tilde{Y}_t^k - \mathcal{Y}_t^k|^2 \right) \leq C \left(\tau + \mathbb{E} |\bar{g}_k(\mathcal{X}_N^k, o_{1:k}) - \mathcal{Y}_N^k|^2 \right).$$

From (21) there exists $C_2 > 0$ such that for all $k \in \{0, \dots, K-1\}$, $o_{1:k} \in \mathcal{O}_k$, we have

$$(22) \quad \sup_{t \in [t_k, t_{k+1}]} \mathbb{E} \left[|\tilde{Y}_t^k - \mathcal{Y}_t^k|^2 \right]^{\frac{1}{2}} \leq C_2 \left(\tau^{\frac{1}{2}} + \mathbb{E} \left[|\bar{g}_k(\mathcal{X}_N^k, o_{1:k}) - \mathcal{Y}_N^k|^2 \right]^{\frac{1}{2}} \right).$$

We note that [25] adopts the same convention for the process Z as [18], so the same adaptation applies.

3.2. Convergence. We are now ready to state our main result, building on the *a priori* estimate (20) and error bound (22). Similar to (21) our theorem is a hybrid *a priori* and *a posteriori* bound. The *a posteriori* term is directly related to the objective function in (16).

Theorem 3.1. *Let $p_k, k = 1, \dots, K$, be the solution to (5)–(7), $\widehat{p}_k, k = 1, \dots, K$, be given by (18), \mathcal{X}^k and \mathcal{Y}^k be defined as in (16). If b satisfies (10), then there exists a constant $C > 0$, depending on f, T, K, C_1, C_2 and L , such that*

$$(23) \quad \sup_{k \in \{1, \dots, K\}} \|p_k(t_k) - \widehat{p}_k\|_{L^\infty(\mathbb{O}_k; L^\infty(\mathbb{R}^d; \mathbb{R}))} \leq C \left(\tau^{\frac{1}{2}} + \sum_{k=0}^{K-1} \sup_{o_{1:k} \in \mathbb{O}_k} \sup_{x \in \mathbb{R}^d} \mathbb{E} \left[|\bar{g}_k(\mathcal{X}_N^{k,x}, o_{1:k}) - \mathcal{Y}_N^{k,x}|^2 \right]^{\frac{1}{2}} \right).$$

Remark 3.1. *An analog of Theorem 3.1 can be obtained for a more general drift coefficient $b \in C_b^\alpha(\mathbb{R}^d; \mathbb{R}^d)$ under the stronger "ellipticity" condition that $a = \sigma \sigma^\top$ is invertible. This can be proven by following an analogue to the proof presented below, but requires ellipticity to use a corresponding a priori estimates from [18, Proposition 2.1] to the Z process.*

Proof of Theorem 3.1. We begin by fixing $k \in \{0, \dots, K-1\}$ and take $o_{1:k+1} \in \mathbb{O}_{k+1}$. Throughout the proof, $C > 0$ denotes a generic constant whose value may change from line to line. The constant C may depend on f, T, K, C_1, C_2 , and the uniform bound on the likelihood L , but is independent of τ, x , and the observations $o_{1:K} \in \mathbb{O}_K$. We recall that the exact filter solution is obtained by multiplying the prediction density with the likelihood

$$p_{k+1}(t_{k+1}, x, o_{1:k+1}) = p_k(t_{k+1}, x, o_{1:k})L(o_{k+1}, x).$$

Analogously, we recall the approximation, \widehat{p} , from (18), defined by

$$\widehat{p}_{k+1}(x, o_{1:k+1}) = w_k(x, o_{1:k})L(o_{k+1}, x).$$

Recalling the uniform bounding constant on L , we get

$$(24) \quad \begin{aligned} & \sup_{x \in \mathbb{R}^d} |p_{k+1}(t_{k+1}, x, o_{1:k+1}) - \widehat{p}_{k+1}(x, o_{1:k+1})| \\ &= \sup_{x \in \mathbb{R}^d} |p_k(t_{k+1}, x, o_{1:k})L(o_{k+1}, x) - w_k(x, o_{1:k})L(o_{k+1}, x)| \\ &\leq C \sup_{x \in \mathbb{R}^d} |p_k(t_{k+1}, x, o_{1:k}) - w_k(x, o_{1:k})|. \end{aligned}$$

To analyze this difference, we recall the Feynman–Kac formulation of p from Corollary 2.1 and the simplified f in (11) which follows by (10). We replace t_{k+1} by $t_{k,N}$ and t_k by $t_{k,0}$ to obtain

$$p_k(t_{k+1}, x) = \mathbb{E} \left[g_k(X_{t_{k,N}}^{k,x}) + \int_{t_{k,0}}^{t_{k,N}} f(X_s^{k,x}, Y_s^{k,x}) ds \right].$$

For convenience, we have dropped the $o_{1:k}$ and o_{k+1} from the notation and will continue to do so whenever possible. Likewise, we see that the approximation w_k , defined by (16), satisfies

$$\begin{aligned} w_k(x) &= \mathbb{E}[w_k(\mathcal{X}_0^k) | \mathcal{X}_0^k = x] \\ &= \mathbb{E} \left[\mathcal{Y}_N^{k,x} + \sum_{\ell=0}^{N-1} \left(f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x})(t_{k,\ell+1} - t_{k,\ell}) + (\mathcal{Z}_\ell^{k,x})^\top \sigma(\mathcal{X}_\ell^{k,x})(W_{t_{k,\ell+1}} - W_{t_{k,\ell}}) \right) \right] \\ &= \mathbb{E} \left[\mathcal{Y}_N^{k,x} + \sum_{\ell=0}^{N-1} f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x})(t_{k,\ell+1} - t_{k,\ell}) \right]. \end{aligned}$$

Looking at the difference, rearranging the terms and applying the triangle inequality, we see that

$$\begin{aligned}
& |p_k(t_{k+1}, x) - w_k(x)| \\
&= \left| \mathbb{E} \left[g_k(X_{t_{k,N}}^{k,x}) + \int_{t_{k,0}}^{t_{k,N}} f(X_s^{k,x}, Y_s^{k,x}) ds \right] - \mathbb{E} \left[\mathcal{Y}_N^{k,x} + \sum_{\ell=0}^{N-1} f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x})(t_{k,\ell+1} - t_{k,\ell}) \right] \right| \\
&\leq \left| \mathbb{E} \left[g_k(X_{t_{k,N}}^{k,x}) - \mathcal{Y}_N^{k,x} \right] \right| + \left| \sum_{\ell=0}^{N-1} \mathbb{E} \left[\int_{t_{k,\ell}}^{t_{k,\ell+1}} f(X_s^{k,x}, Y_s^{k,x}) - f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x}) ds \right] \right| \\
&= \text{I} + \text{II}.
\end{aligned}$$

We begin by looking at the first term I, in which we add and subtract the auxiliary solution (\tilde{Y}^k) that satisfies $\tilde{Y}_{t_{k,N}}^{k,x} = \bar{g}_k(X_{t_{k,N}}^{k,x})$. Together with the triangle inequality, we obtain

$$\begin{aligned}
\text{I} &= \left| \mathbb{E} \left[g_k(X_{t_{k,N}}^{k,x}) - \tilde{Y}_{t_{k,N}}^{k,x} \right] + \mathbb{E} \left[\tilde{Y}_{t_{k,N}}^{k,x} - \mathcal{Y}_N^{k,x} \right] \right| \\
&\leq \mathbb{E} \left[\left| g_k(X_{t_{k,N}}^{k,x}) - \bar{g}_k(X_{t_{k,N}}^{k,x}) \right| \right] + \mathbb{E} \left[\left| \tilde{Y}_{t_{k,N}}^{k,x} - \mathcal{Y}_N^{k,x} \right| \right] \\
&= \text{I}_1 + \text{I}_2.
\end{aligned}$$

For the first term, we recall g_k from (13) and \bar{g}_k from (17). By applying Hölder's inequality and (4) we get

$$\begin{aligned}
\text{I}_1 &= \mathbb{E} \left[\left| p_{k-1}(t_k, X_{t_{k,N}}^{k,x}) L(o_k, X_{t_{k,N}}^{k,x}) - w_{k-1}(X_{t_{k,N}}^{k,x}) L(o_k, X_{t_{k,N}}^{k,x}) \right| \right] \\
&\leq C \mathbb{E} \left[\left| p_{k-1}(t_k, X_{t_{k,N}}^{k,x}) - w_{k-1}(X_{t_{k,N}}^{k,x}) \right| \right] \\
&\leq C \sup_{x \in \mathbb{R}^d} |p_{k-1}(t_k, x) - w_{k-1}(x)|.
\end{aligned}$$

This is the error from the previous time step multiplied with a constant. For the term I_2 we apply the Cauchy–Schwarz inequality and (22) to obtain

$$\text{I}_2 \leq \mathbb{E} \left[\left| \tilde{Y}_{t_{k,N}}^{k,x} - \mathcal{Y}_N^{k,x} \right|^2 \right]^{\frac{1}{2}} \leq C_2 \left(\tau^{\frac{1}{2}} + \mathbb{E} \left[\left| \bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x} \right|^2 \right]^{\frac{1}{2}} \right).$$

By collecting I_1 and I_2 , we get

$$\text{I} \leq C \left(\sup_{x \in \mathbb{R}^d} |p_{k-1}(t_k, x) - w_{k-1}(x)| + \tau^{\frac{1}{2}} + \mathbb{E} \left[\left| \bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x} \right|^2 \right]^{\frac{1}{2}} \right).$$

Next we rewrite the term II by applying the Fubini–Tonelli theorem and using the triangle inequality to obtain

$$\begin{aligned}
\text{II} &= \left| \sum_{\ell=0}^{N-1} \mathbb{E} \left[\int_{t_{k,\ell}}^{t_{k,\ell+1}} f(X_s^{k,x}, Y_s^{k,x}) - f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x}) ds \right] \right| \\
&\leq \sum_{\ell=0}^{N-1} \int_{t_{k,\ell}}^{t_{k,\ell+1}} \mathbb{E} \left[\left| f(X_s^{k,x}, Y_s^{k,x}) - f(\mathcal{X}_\ell^{k,x}, \mathcal{Y}_\ell^{k,x}) \right| \right] ds.
\end{aligned}$$

The absolute integrability of the integrand, required for the use of the Fubini–Tonelli theorem, can be shown by the properties of f and by standard moment estimates. We continue by using the Lipschitz continuity of f and applying the Cauchy–Schwarz inequality

$$\begin{aligned}
\text{II} &\leq C \sum_{\ell=0}^{N-1} \int_{t_{k,\ell}}^{t_{k,\ell+1}} \left(\mathbb{E} \left[\left\| X_s^{k,x} - \mathcal{X}_\ell^{k,x} \right\| + \left| Y_s^{k,x} - \mathcal{Y}_\ell^{k,x} \right| \right] \right) ds \\
(25) \quad &\leq C \sum_{\ell=0}^{N-1} \int_{t_{k,\ell}}^{t_{k,\ell+1}} \left(\mathbb{E} \left[\left\| X_s^{k,x} - \mathcal{X}_\ell^{k,x} \right\|^2 \right]^{\frac{1}{2}} + \mathbb{E} \left[\left| Y_s^{k,x} - \mathcal{Y}_\ell^{k,x} \right|^2 \right]^{\frac{1}{2}} \right) ds.
\end{aligned}$$

The first term in the integral, is the strong error of the Euler–Maruyama approximation of a time homogeneous SDE, and satisfies order $\frac{1}{2}$ convergence in τ . For the second term, we add and

subtract the auxiliary process \tilde{Y}^k , defined as the solution to (19), inside the norm. Applying the triangle inequality and taking the supremum over $[t_k, t_{k+1}]$ yields

$$\begin{aligned} \mathbb{E}\left[|Y_s^{k,x} - \mathcal{Y}_\ell^{k,x}|^2\right]^{\frac{1}{2}} &\leq \mathbb{E}\left[|Y_s^{k,x} - \tilde{Y}_s^{k,x}|^2\right]^{\frac{1}{2}} + \mathbb{E}\left[|\tilde{Y}_s^{k,x} - \mathcal{Y}_\ell^{k,x}|^2\right]^{\frac{1}{2}} \\ &\leq \sup_{t \in [t_k, t_{k+1}]} \mathbb{E}\left[|Y_t^{k,x} - \tilde{Y}_t^{k,x}|^2\right]^{\frac{1}{2}} + \sup_{t \in [t_k, t_{k+1}]} \mathbb{E}\left[|\tilde{Y}_t^{k,x} - \mathcal{Y}_t^{k,x}|^2\right]^{\frac{1}{2}}. \end{aligned}$$

By applying (20) and (22) for the respective term, we get

$$\mathbb{E}\left[|Y_s^{k,x} - \mathcal{Y}_\ell^{k,x}|^2\right]^{\frac{1}{2}} \leq C \left(\mathbb{E}\left[|g_k(X_{t_k, N}^{k,x}) - \bar{g}_k(X_{t_k, N}^{k,x})|^2\right]^{\frac{1}{2}} + \tau^{\frac{1}{2}} + \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right).$$

Inserting this bound into (25), we obtain

$$\begin{aligned} \text{II} &\leq C \sum_{\ell=0}^{N-1} \int_{t_k, \ell}^{t_{k+1}} \left(\tau^{\frac{1}{2}} + \mathbb{E}\left[|g_k(X_{t_k, N}^{k,x}) - \bar{g}_k(X_{t_k, N}^{k,x})|^2\right]^{\frac{1}{2}} + \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right) ds \\ &\leq C \frac{T}{K} \left(\tau^{\frac{1}{2}} + \mathbb{E}\left[|g_k(X_{t_k, N}^{k,x}) - \bar{g}_k(X_{t_k, N}^{k,x})|^2\right]^{\frac{1}{2}} + \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right) ds. \end{aligned}$$

Adapting the same arguments as for I_1 on the second term, we have

$$\text{II} \leq C \left(\tau^{\frac{1}{2}} + \sup_{x \in \mathbb{R}^d} |p_{k-1}(x) - w_{k-1}(x)| + \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right),$$

which up to a constant is the same to the bound for I . By defining the error term $e_k := \sup_{x \in \mathbb{R}^d} |p_k(t_{k+1}, x) - w_k(x)|$ and combining the bounds of I and II we get for some constant $C > 0$ depending on f, T, K, C_1, C_2 , and L

$$|p_k(t_{k+1}, x) - w_k(x)| \leq C e_{k-1} + C \left(\tau^{\frac{1}{2}} + \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right).$$

Taking supremum over $x \in \mathbb{R}^d$ on the right-hand side, followed by the left-hand side, we obtain the recursive bound

$$e_k \leq C e_{k-1} + C \left(\tau^{\frac{1}{2}} + \sup_{x \in \mathbb{R}^d} \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right).$$

Applying this bound recursively, we get

$$\begin{aligned} e_k &\leq \sum_{j=0}^k C^{k-j+1} \left(\tau^{\frac{1}{2}} + \sup_{x \in \mathbb{R}^d} \mathbb{E}\left[|\bar{g}_j(\mathcal{X}_N^{j,x}) - \mathcal{Y}_N^{j,x}|^2\right]^{\frac{1}{2}} \right) \\ &\leq C \left(\tau^{\frac{1}{2}} + \sum_{j=0}^{K-1} \sup_{x \in \mathbb{R}^d} \mathbb{E}\left[|\bar{g}_j(\mathcal{X}_N^{j,x}) - \mathcal{Y}_N^{j,x}|^2\right]^{\frac{1}{2}} \right), \end{aligned}$$

for some constant C depending on f, T, K, C_1, C_2 and L . We insert this bound into (24) and finish by taking the supremum over $o_{1:K} \in \mathbb{O}_K$ on the right-hand side, moving it inside the sum and finally taking the supremum on the left-hand side to obtain

$$\|p_{k+1}(t_{k+1}) - \hat{p}_{k+1}\|_{L^\infty(\mathbb{O}_k; L^\infty(\mathbb{R}^d; \mathbb{R}))} \leq C \left(\tau^{\frac{1}{2}} + \sum_{k=0}^{K-1} \sup_{o_k \in \mathbb{O}_k} \sup_{x \in \mathbb{R}^d} \mathbb{E}\left[|\bar{g}_k(\mathcal{X}_N^{k,x}, o_{1:k}) - \mathcal{Y}_N^{k,x}|^2\right]^{\frac{1}{2}} \right).$$

Since the bound holds for all $k \in \{0, \dots, K-1\}$, the proof is complete. \square

4. NUMERICAL RESULTS

In this section, we investigate the convergence result of Theorem 3.1 numerically. To this end we begin in Section 4.1 by outlining the final approximation steps for an employable algorithm. Section 4.2 contains the two numerical examples in which we evaluate the error terms of the bound (23) numerically.

4.1. Numerical approximation. In (16), we defined the optimization problem over all continuous functions, which is not feasible in practice. Instead, we consider classes of neural networks denoted $\mathbf{NN}^{\Theta,p,k} \subset L^\infty(\mathbb{O}_k; C(\mathbb{R}^d; \mathbb{R}^p))$ for $(p, k) \in \{1, d\} \times \{0, 1, \dots, K-1\}$. The specifications of these classes, and the set Θ of learnable parameters, are presented in Appendix A. By parameterizing the continuous functions (w, \bar{v}) using neural networks $(w^\theta, \bar{v}^\theta) \in \mathbf{NN}^{\Theta,1,k} \times \mathbf{NN}^{\Theta,d,k}$ and optimizing over their weights θ , the scheme becomes fully implementable. This approach leverages the expressive power of neural networks, enabling a flexible and efficient way of approximating the filtering density. The new recursive scheme is given, for $k = 0, \dots, K-1$, by

$$(26) \quad \min_{\substack{w^\theta \in \mathbf{NN}^{\Theta,1,k} \\ (\bar{v}^\theta)_{n=0}^{N-1} \in \prod_{n=0}^{N-1} \mathbf{NN}^{\Theta,d,k}}} \mathbb{E} \left[\left| \mathcal{Y}_N^k - \bar{g}_k(\mathcal{X}_N^k, O_{1:k}) \right|^2 \right]$$

$$\mathcal{X}_{n+1}^k = \mathcal{X}_n^k + b(\mathcal{X}_n^k)(t_{k,n+1} - t_{k,n}) + \sigma(\mathcal{X}_n^k)(W_{t_{k,n+1}} - W_{t_{k,n}}), \quad n = 0, \dots, N-1,$$

$$\mathcal{Z}_n^k = \bar{v}_n^\theta(\mathcal{X}_n^k, O_{1:k}), \quad n = 0, \dots, N-1,$$

$$\mathcal{Y}_{n+1}^k = w^\theta(\mathcal{X}_0^k, O_{1:k}) - \sum_{\ell=0}^n \left(f_b(\mathcal{X}_\ell^k, \mathcal{Y}_\ell^k, \mathcal{Z}_\ell^k)(t_{k,\ell+1} - t_{k,\ell}) - (\mathcal{Z}_\ell^k)^\top \sigma(\mathcal{X}_\ell^k)(W_{t_{k,\ell+1}} - W_{t_{k,\ell}}) \right), \quad n = 0, \dots, N-1.$$

Likewise, the expected value in the objective function cannot be evaluated in closed form and must instead be approximated via Monte Carlo sampling. To this end, we generate a set of M_{train} independent samples $(S^{(m)}, O^{(m)}, X^{(m)}, W^{(m)})_{m=1}^{M_{\text{train}}}$ from the system (1), (2), and the auxiliary process (8). These samples define M_{train} trajectories over which we evaluate the loss functional appearing in (26). The resulting empirical average used during training is defined as

$$(27) \quad \min_{\substack{w^\theta \in \mathbf{NN}^{\Theta,1,k} \\ (\bar{v}^\theta)_{n=0}^{N-1} \in \prod_{n=0}^{N-1} \mathbf{NN}^{\Theta,d,k}}} \frac{1}{M_{\text{train}}} \sum_{m=1}^{M_{\text{train}}} \left| \mathcal{Y}_N^{k,(m)} - \bar{g}_k(\mathcal{X}_N^{k,(m)}, O_{1:k}^{(m)}) \right|^2.$$

A schematic overview of the method is presented in Figure 2.

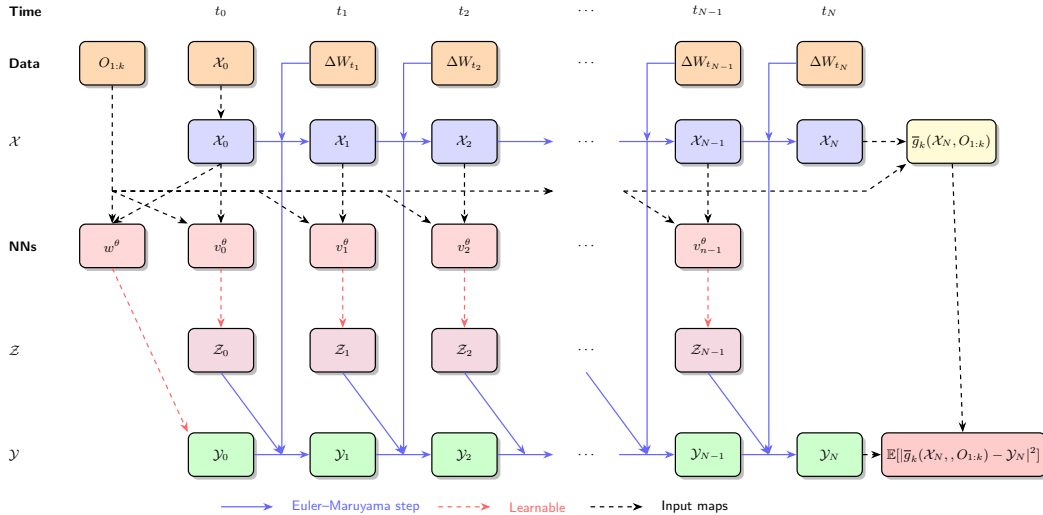


Figure 2. The figure illustrates the numerical scheme for a fixed $k = 1, \dots, K-1$ in the optimization problem (26). The color of the arrows between the boxes indicate if the map consists of an Euler–Maruyama step, a parameterized neural network step or simply an input to a function.

In the experiments in Section 4.2 we use the normalized versions of \bar{g}_k , $k = 1, \dots, K-1$, rather than the unnormalized counterparts. More precisely, we redefine the target function in (27), for

$k = 1, \dots, K - 1$, $o_{1:k} \in \mathbb{O}_k$, as

$$\begin{aligned}\bar{g}_k(x, o_{1:k}) &= \frac{w_{k-1}^*(x, o_{1:k-1})L(o_k, x)}{C(o_{1:k})}, \quad x \in \mathbb{R}^d, \\ C(o_{1:k}) &= \int_{\mathbb{R}^d} w_{k-1}^*(z, o_{1:k-1})L(o_k, z) dz.\end{aligned}$$

In the one-dimensional examples, the normalizing constant $C(O_{1:k}^{(m)})$ required for each training sample $m = 1, \dots, M_{\text{train}}$ is approximated using quadrature. Specifically, we define

$$\widehat{C}^{(m)} = \sum_{j=1}^J w_{k-1}^*(z_j, o_{1:k-1})L(o_k, z_j)\Delta z,$$

where $(z_j)_{j=1}^J \subset \mathbb{R}$ is a fixed equidistant spatial grid with $\Delta z = z_j - z_{j-1}$ for $j = 2, \dots, J$. The grid is chosen to appropriately cover the support of the solution and to resolve the relevant features of the integrand. This quadrature-based approximation of the normalizing constant introduces an additional source of numerical error. To mitigate its effect, the number of grid points J must be chosen sufficiently large to ensure accurate integration. Further implementation details are provided in Appendix A.

4.2. Numerical experiments. In this section we empirically examine the numerical convergence of our approximate filter applied to (1)–(2). We test the method on two one-dimensional examples, one linear and one nonlinear. In both examples we set $T = 1$, $K = 10$, $d = d' = 1$, $\sigma(x) = 1$, $L(o, x) = \mathcal{N}(o | h(x), R)$ with $h(x) = x$ and $R = 1$. We note that this is a fairly low signal-to-noise ratio possibly making the filtering problem harder.

Recalling the error bound (23) from Theorem 3.1, we now turn to estimating the quantities appearing in the bound. Specifically, we study convergence numerically by evaluating the error between p_k and \widehat{p}_k in the norm $L^\infty(\mathbb{O}_k; L^\infty(\mathbb{R}^d; \mathbb{R}))$, along with the corresponding *a posteriori* term, across a range of discretizations. In the figures below, we denote these two quantities by

$$\begin{aligned}e_k &= \|p_k(t_k) - \widehat{p}_k\|_{L^\infty(\mathbb{O}_k; L^\infty(\mathbb{R}^d; \mathbb{R}))}, \quad k = 1, \dots, K, \\ E_k &= \sup_{o_{1:k} \in \mathbb{O}_k} \sup_{x \in \mathbb{R}^d} \mathbb{E} \left[|\bar{g}_{k-1}(\mathcal{X}_N^{k-1, x}, o_{1:k}) - \mathcal{Y}_N^{k-1, x}|^2 \right]^{\frac{1}{2}}, \quad k = 1, \dots, K.\end{aligned}$$

Here, e_k measures the pointwise approximation error of the filtering density at time t_k , whereas E_k reflects the residual error in the learning objective, serving as an *a posteriori* error of how far the obtained solution is from the theoretical optimum at time t_k . See Appendix A for further details on the numerical approximation of these error terms.

Ornstein–Uhlenbeck process. The first equation we consider is a linear SDE with the drift coefficient $\mu(x) = -x$ and is solved by the Ornstein–Uhlenbeck process. In this linear setting, with a linear measurement function, the solution p to (5)–(7) is computed exactly by the Kalman filter. In this example, the coefficients satisfy conditions ((i)) and ((ii)), and the convergence in Theorem 3.1 applies. In Figure 3, we report the pointwise errors e_k and E_k , evaluated at each observation time $t_k = \frac{k}{K}$ for $k = 1, \dots, K$. The experiment is repeated for seven discretizations with $N = 2^j$, $j = 0, \dots, 6$, corresponding to the number of time steps between successive observation times. We note that both error terms decrease at all time steps as we increase the number of intermediate discretization steps. As expected, both error metrics decrease uniformly in time as N increases. We also note that, for coarse discretizations, e_k increases over time, indicating a cumulative error effect. However, for sufficiently fine resolutions ($N \geq 16$), the error remains nearly constant. The behavior of E_k exhibits a similar trend, although not as clear.

To investigate the convergence rate, we fix $k = K$ and plot both the final time error e_K and the accumulated *a posteriori* term $E := \sum_{k=1}^K E_k$ in Figure 4. These quantities are shown together with a reference slope of $N^{-\frac{1}{2}}$ on a logarithmic scale. We see that both terms converge at least with order $\frac{1}{2}$. These results are consistent with the theoretical error bound in (23), where e_K is dominated by the discretization error $N^{-\frac{1}{2}}$ and the residual E .

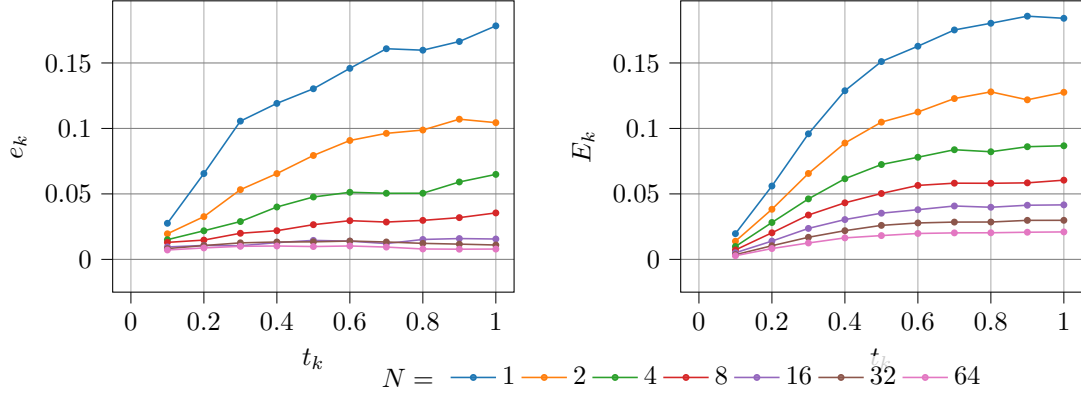


Figure 3. Ornstein-Uhlenbeck process. The figure illustrates the error trajectories for seven different discretizations. To the left we see the error e_k and to the right we see the residual E_k , $k = 1, \dots, K$.

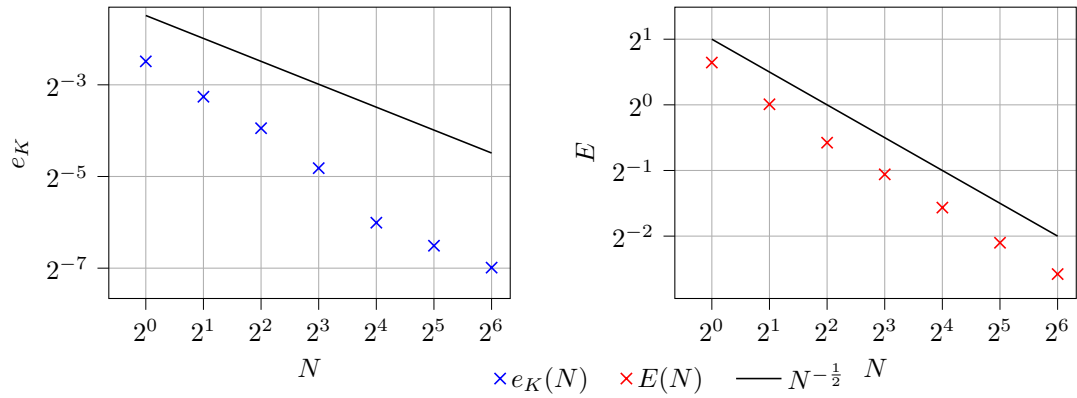


Figure 4. Ornstein-Uhlenbeck process. The figure presents the error and the accumulated residual at the final time for seven different discretizations together with a reference line of order $\frac{1}{2}$. To the left we have e_K and to the right the accumulated residual $E = \sum_{k=1}^K E_k$.

Bistable process. The second example concerns an SDE with nonlinear drift given by $\mu(x) = \frac{2}{5}(5x - x^3)$. Here, condition ((i)) is not satisfied due to the cubic drift, and no theoretical convergence guarantee is available. The drift corresponds to the gradient of a double-well potential, leading to a bimodal invariant distribution for S_t as $t \rightarrow \infty$. The dynamics (1) of S , initialized from π_0 , tend to drive the state S toward the two stable modes of the potential landscape. In this nonlinear setting, no closed-form solution is available for the coupled system (5)–(7). As a result, we employ a particle filter to compute a reference solution; implementation details are provided in Appendix A. In Figure 5, we present the error measures e_k and E_k at each observation time t_k , for $k = 1, \dots, K$, across seven discretizations with $N = 2^j$, $j = 0, \dots, 6$. Similar to the linear case, the error decreases uniformly as N increases. Moreover, for the finest discretizations ($N = 32$ and $N = 64$), the error appears to stabilize over time.

In Figure 6, we examine the convergence behavior at the final time t_K . Both the final-time error e_K and the accumulated *a posteriori* term $E = \sum_{k=1}^K E_k$ are plotted against a reference slope of $N^{-\frac{1}{2}}$ on a logarithmic scale. The error e_K exhibits an order $\frac{1}{2}$ convergence rate for all but the finest discretization. In contrast, the *a posteriori* term E follows an order $\frac{1}{2}$ decay up to $N = 16$, after which the convergence stagnates. This behavior aligns with the theoretical structure of the error bound in (23): once E ceases to decrease, the total error e_K likewise plateaus, reflecting its dependence on E as one of the dominant terms. We emphasize, however, that the coefficients in this example do not satisfy the assumptions of Theorem 3.1.

The observed plateau of E for large N is likely due to limitations in the practical implementation of the method. Potential causes include insufficient Monte Carlo sampling to estimate the expected

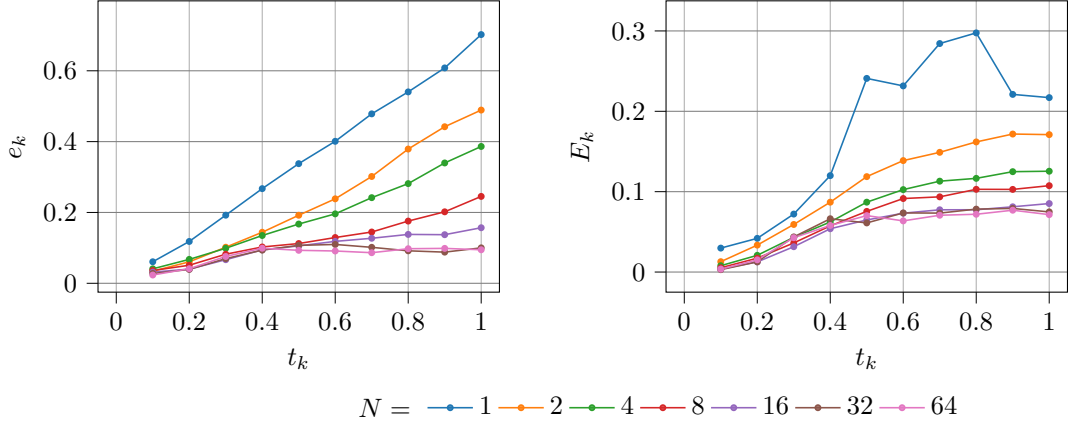


Figure 5. Bistable process. The figure illustrates the error trajectories for seven different discretizations. To the left we see the error e_k and to the right we see the residual E_k , $k = 1, \dots, K$.

value in (27), suboptimal network architectures, or challenges in optimization during training. These factors may dominate the discretization error beyond a certain resolution, thus limiting further convergence.

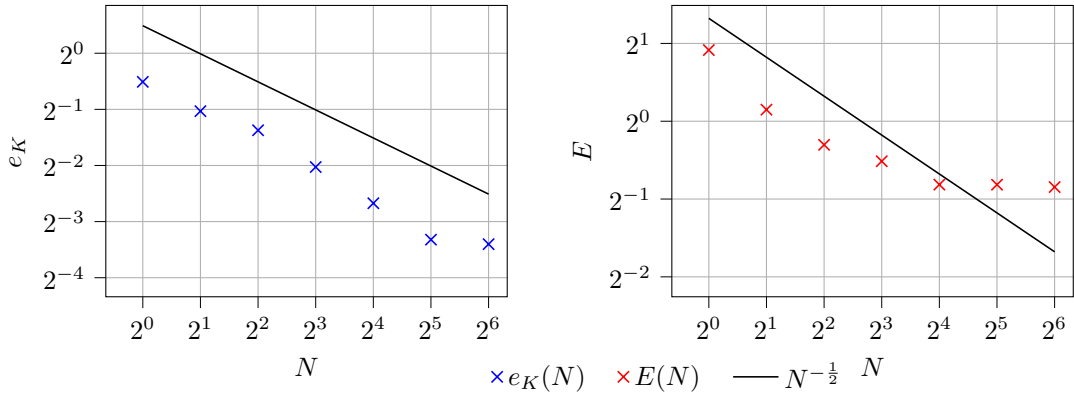


Figure 6. Bistable process. The figure presents the error and the accumulated residual at the final time for seven different discretizations together with a reference line of order $\frac{1}{2}$. To the left we have e_K and to the right the accumulated residual $E = \sum_{k=1}^K E_k$.

5. CONCLUSION AND OUTLOOK

In this paper, considering the filtering problem in a continuous-discrete setting, we introduce an approximate method, prove an error bound for the method, and empirically confirm the theoretical rate. The objective is to develop a filtering approach suitable for high-dimensional and strongly nonlinear settings, where neither Kalman filters nor particle filters are sufficient. This work establishes the theoretical foundation, while a complementary numerical study focusing on high-dimensional problems was conducted concurrently by the first author confirming its numerical robustness [4].

The numerical results confirm the convergence predicted by Theorem 3.1. In both the linear and nonlinear examples, the accuracy improves with finer temporal discretization, and for sufficiently fine resolutions, the error, for both left-hand side and right-hand side, remains essentially constant over time. The observed rates match the theoretical prediction until the improvement is limited by factors such as network capacity or training accuracy, rather than by the discretization itself. This indicates that the method achieves the theoretically predicted rates until it is constrained by implementation aspects.

Acknowledgements. The authors would like to thank Melker Bild for his contributions through initial experiments conducted as part of his master’s thesis, which explored related aspects of deep BSDE filtering. The work of K.B. and S.L. was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers e-Commons partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

REFERENCES

- [1] K. Andersson, A. Andersson, and C. W. Oosterlee. The deep multi-FBSDE method: a robust deep learning method for coupled FBSDEs. *arXiv:2503.13193*, 2025.
- [2] K. Bågmark, A. Andersson, and S. Larsson. An energy-based deep splitting method for the nonlinear filtering problem. *Partial Differ. Equ. Appl.*, 4(2):14, 2023.
- [3] K. Bågmark, A. Andersson, S. Larsson, and F. Rydin. A convergent scheme for the Bayesian filtering problem based on the Fokker–Planck equation and deep splitting. *arXiv:2409.14585*, 2024.
- [4] K. Bågmark and F. Rydin. High-dimensional Bayesian filtering through deep density approximation. *arXiv:2511.07261*, 2025.
- [5] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
- [6] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. Deep learning based numerical approximation algorithms for stochastic partial differential equations and high-dimensional nonlinear filtering problems. *arXiv:2012.01194*, 2020.
- [7] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. Deep splitting method for parabolic PDEs. *SIAM J. Sci. Comput.*, 43:A3135–A3154, 2021.
- [8] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.
- [9] S. Challa and Y. Bar-Shalom. Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions. *IEEE Trans. Aerosp. Electron. Syst.*, 36:309–315, 2000.
- [10] N. Chopin, A. Fulop, J. Heng, and A. H. Thiery. Computational Doob h-transforms for online filtering of discretely observed diffusions. In *Int. Conf. Mach. Learn.*, pages 5904–5923. PMLR, 2023.
- [11] A. Corenflos and A. Finke. Particle-MALA and particle-mGRAD: Gradient-based MCMC methods for high-dimensional state-space models. *arXiv:2401.14868*, 2024.
- [12] A. Corenflos, Z. Zhao, T. B. Schön, S. Särkkä, and J. Sjölund. Conditioning diffusion models by explicit forward-backward bridging. In *Int. Conf. Artif. Intell. Stat.*, pages 3709–3717. PMLR, 2025.
- [13] D. Crisan, A. Lobbe, and S. Ortiz-Latorre. An application of the splitting-up method for the computation of a neural network representation for the solution for the filtering equations. *Stoch. Partial Differ. Equ.: Anal. Comput.*, 10:1050–1081, 2022.
- [14] N. Cui, L. Hong, and J. R. Layne. A comparison of nonlinear filtering approaches with an application to ground target tracking. *Signal Processing*, 85:1469–1492, 2005.
- [15] B. Demissie, M. A. Khan, and F. Govaers. Nonlinear filter design using Fokker-Planck propagator in Kronecker tensor format. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2016.
- [16] 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.*, 5:349–380, Nov. 2017.
- [17] W. E and B. Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.*, 1:1–12, 2018.
- [18] N. El Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Math. Finance*, 7(1):1–71, 1997.
- [19] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2009.
- [20] A. Finke and A. H. Thiery. Conditional sequential Monte Carlo in high dimensions. *Ann. Statist.*, 51:437–463, 2023.
- [21] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers Group, Dordrecht, 1997.
- [22] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.
- [23] F. K. Gustafsson, M. Danelljan, G. Bhat, and T. B. Schön. Energy-based models for deep probabilistic regression. In *European Conference on Computer Vision*, pages 325–343. Springer, 2020.
- [24] J. Han, W. Hu, J. Long, and Y. Zhao. Deep Picard iteration for high-dimensional nonlinear PDEs. *SIAM J. Sci. Comput.*, 48(1):C1–C24, 2026.
- [25] J. Han and J. Long. Convergence of the deep BSDE method for coupled FBSDEs. *Probab. Uncertain. Quant. Risk*, 5:Paper No. 5, 33, 2020.
- [26] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *Int. J. Comput. Vis.*, 29(1):5–28, 1998.
- [27] A. Jasra, D. A. Stephens, and C. C. Holmes. Population-based reversible jump Markov chain Monte Carlo. *Biometrika*, 92(4):803–820, 2005.

- [28] M. S. Johannes and N. G. Polson. MCMC Methods for Continuous-Time Financial Econometrics. In *Handbook of Financial Econometrics*, pages 1–72. Elsevier, 2009.
- [29] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng.*, 83:95–108, 1961.
- [30] L. Kapllani and L. Teng. A backward differential deep learning-based algorithm for solving high-dimensional nonlinear backward stochastic differential equations. *IMA J. Numer. Anal.*, 2025.
- [31] K. P. Körding and D. M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247, 2004.
- [32] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.*, 34:26548–26560, 2021.
- [33] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3:218–229, 2021.
- [34] K. Luo, J. Zhao, Y. Wang, J. Li, J. Wen, J. Liang, H. Soekmadji, and S. Liao. Physics-informed neural networks for PDE problems: a comprehensive review. *Artif. Intell. Rev.*, 58(10):1–43, 2025.
- [35] P. S. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, 1979.
- [36] C. A. Naesseth, F. Lindsten, and T. B. Schön. High-dimensional filtering using nested sequential Monte Carlo. *IEEE Trans. Signal Process.*, 67:4177–4188, 2019.
- [37] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer Science & Business Media, 2003.
- [38] 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.
- [39] M. Schauer, F. van der Meulen, and H. van Zanten. Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli*, 23(4A):2917–2950, 2017.
- [40] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [41] C. Snyder. Particle filters, the “optimal” proposal and high-dimensional systems. In *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean*, pages 1–10, 2011.
- [42] C. Snyder, T. Bengtsson, and M. Morzfeld. Performance bounds for particle filters using the optimal proposal. *Mon. Weather Rev.*, 143:4750–4761, 2015.
- [43] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [44] Z. Zhao, Z. Luo, J. Sjölund, and T. B. Schön. Conditional sampling within generative diffusion models. *Philos. Trans. R. Soc. A*, 383(2299):20240329, 2025.

APPENDIX A. IMPLEMENTATION DETAILS

In this section, we present the implementation details. Section A.1 is on the architecture and parameters and on how the models are trained. Section A.2 briefly describes the reference solutions. Section A.3 describes how the strong error term is evaluated.

A.1. Networks and training. The spaces $\mathbf{NN}^{\Theta,p,k}$, $p = 1$ or $p = d$, and $k = 0, \dots, K - 1$, which define our models, consist of fully connected feed-forward neural networks with three hidden layers, one input layer, and one output layer. The training is organized into K sequential minimization problems, one for each prediction step $k = 0, \dots, K - 1$. In each minimization problem, we employ a single network for the w component of the solution and N distinct networks for the v components, one for each time step in the discretization. The architectural differences between the w and v networks are summarized in Table 1. To allow parameter transfer between consecutive minimization problems, all models share the same fixed input dimension, independent of the number of observational inputs available at a given step. For time points without observations, the corresponding input components are set to zero. We choose a smaller architecture for the v -networks, since each of them models only a short propagation of the process \mathcal{Y} , whereas the w -network represents the full predictive density after training. The w -network uses an exponential output, similar to that of energy-based methods [2, 23]. Training was carried out with the Adam optimizer with standard

	w -network	v -networks
Input dimension	$d + (K - 1) \times d'$	$d + (K - 1) \times d'$
Hidden layer size	128	32
Activation function (hidden)	ReLU	ReLU
Activation function (output)	Exponential	None
Output dimension	1	d
Number of networks	1	N

Table 1. Architectural details of the models.

parameters, using a fixed learning rate of 10^{-4} and a constant batch size of 512. Each epoch began by generating 200 batches of simulated trajectories from the process (S, O, \mathcal{X}, W) , acting as input to the scheme, and evaluating the terminal condition \bar{y} , acting as the target label in this learning task. Training was stopped after at most 100 epochs, or earlier if the averaged loss over the epoch stopped decreasing for 5 consecutive epochs (early stopping). Thus, we got different M_{train} values (number of Monte Carlo samples) depending on the number of epochs that were required. We found that the models with the coarser discretizations converged with fewer epochs. Additionally, the cost of training is at least proportional to the number of discretization steps due to the sequential nature of the Euler–Maruyama step when calculating \mathcal{Y}_N . The normalization steps performed in both the training and inference time were performed with $J = 10^3$ discretization points over a spatial grid on $[-5, 5]$. All experiments were run on an NVIDIA A40 GPU (48GB memory).

A.2. Reference solutions. For the linear example, the Ornstein–Uhlenbeck process, the reference solution was obtained using a discrete-time Kalman filter [29]. To reduce discretization error in the prediction step, we used 128 intermediate prediction steps in each interval $[t_k, t_{k+1}]$; this should be compared to the largest value of the discretization parameter of the model, which was $N = 64$.

For the nonlinear example, the bistable process, we employed a bootstrap particle filter with the same refinement of 128 intermediate prediction steps for improved accuracy [22]. We used a Gaussian kernel density estimator to recover a probability density [40]. A total of 10^5 particles were used to ensure a high-accuracy estimate of the filtering density and to reduce Monte Carlo variance to a negligible level.

A.3. Evaluation of errors. We approximate the left hand side by considering new samples $(O^{(m)})_{m=1}^{M_{\text{eval}}}$ from (2) for $M_{\text{eval}} > 0$ and a spatial discretization by equidistant $x_i \in \mathbb{R}$, $i = 1, \dots, I$, for $I > 0$. We emphasize that we must consider sampled approximations over \mathbb{O}_k since this space is of dimension 10 for $k = 10$ and $d' = 1$. For both examples, it suffices to consider the domain $[-5, 5] \subset \mathbb{R}$ for the points $(x_i)_{i=1}^I$. By these approximations, for $k = 1, \dots, K$, we get

$$\|p_k(t_k) - \hat{p}_k\|_{L^\infty(\mathbb{O}_k; L^\infty(\mathbb{R}^d; \mathbb{R}))} \approx \max_{\substack{m=1, \dots, M_{\text{eval}} \\ i=1, \dots, I}} \left| p_k(t_k, x_i, O_{1:k}^{(m)}) - \hat{p}_k(x_i, O_{1:k}^{(m)}) \right|.$$

Similarly, we use the samples $(O^{(m)})_{m=1}^{M_{\text{eval}}}$ to evaluate the a posteriori terms E_k . This is done through the same Euler–Maruyama steps as in the minimization formulation (26). In the evaluation, we set $I = 10^3$ and $M_{\text{eval}} = 10^4$.

KASPER BÄGMARK, DEPARTMENT OF MATHEMATICAL SCIENCES, CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG, SE-412 96 GOTHENBURG, SWEDEN

Email address: bagmark@chalmers.se

ADAM ANDERSSON, DEPARTMENT OF MATHEMATICAL SCIENCES, CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG, S-412 96 GOTHENBURG, SWEDEN, AND SAAB AB, S-412 76 GOTHENBURG, SWEDEN

Email address: adam.andersson@chalmers.se

STIG LARSSON, DEPARTMENT OF MATHEMATICAL SCIENCES, CHALMERS UNIVERSITY OF TECHNOLOGY AND UNIVERSITY OF GOTHENBURG, SE-412 96 GOTHENBURG, SWEDEN

Email address: stig@chalmers.se