

Deep Neural Networks for Mean-Field Stochastic Optimal Control with Applications to Mean Variance Optimization

Hamid OULGHAZI

h.oulghazi@umi.ac.ma

Université Moulay Ismail de Meknes, FSTE

Mustapha Hamchichi

Université Moulay Ismail de Meknes, FSTE

Sidi Mohamed Douiri

Université Moulay Ismail de Meknes, FSTE

Research Article

Keywords: Deep Neural Networks, Mean-Field Stochastic Optimal Control, Stochastic Maximum Principle, Hamiltonian system, Forward Backward Stochastic Differential Equation, Mean Variance Optimization

Posted Date: October 1st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11075283/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Deep Neural Networks for Mean-Field Stochastic Optimal Control with Applications to Mean Variance Optimization

xxxxx xxxx^{1*}, xxxx xxxx¹ and xxxxx xxxxxx¹
¹xxxxxx, xxxxxxx, xxxxxx, xxxxxxxxxxx, xxxxxxxx.

*Corresponding author(s). E-mail(s): [xxxxxx](#);
Contributing authors: [xxxxxx](#); [xxxxxxxxxx](#);

Abstract

This paper addresses high-dimensional optimal control problems for mean-field stochastic differential equations. By means of the stochastic maximum principle, we derive an extended Hamiltonian system in the form of a forward-backward stochastic differential equation. We develop three deep learning based algorithms for its numerical solution. The proposed methods are tested on the optimal mean-variance portfolio selection problem and show good accuracy and computational efficiency in high dimensions.

Keywords: Deep Neural Networks, Mean-Field Stochastic Optimal Control, Stochastic Maximum Principle, Hamiltonian system, Forward Backward Stochastic Differential Equation, Mean Variance Optimization

MSC Classification: 93E20 , 49K45 , 60H10 , 60H35 , 68T07 , 91G10

1 Introduction

Mean-field models arise in a wide range of applications, including statistical mechanics and physics, such as the derivation of the Boltzmann and Vlasov equations in kinetic gas theory, as well as quantum mechanics and quantum chemistry through density functional theory and the Hartree-Fock models [1]. Building on these developments, Lasry and Lions extended the mean-field paradigm to economics, finance, and stochastic differential games [2]. Recent studies have demonstrated that probabilistic methods provide an effective framework for analyzing solutions of linear McKean-Vlasov partial

differential equations (see. [3]). Building on these developments, Buckdahn, Djehiche, and Peng [4] investigated a class of mean-field problems using a purely stochastic approach and introduced a new class of backward stochastic differential equations (BSDEs), referred to as mean-field backward stochastic differential equations (MFBSDEs). In subsequent works, the authors addressed stochastic control problems in which the state dynamics are governed by mean-field-type stochastic differential equations (SDEs). By applying Pontryagin’s maximum principle, they derived optimality conditions for both regular and singular control problems, where the coefficients of the SDE depend not only on the state process but also on its expectation [5].

More generally, stochastic optimal control theory relies on Pontryagin’s stochastic maximum principle [6], which was developed in the stochastic setting by Peng [7, 8] and provides necessary conditions for optimality. In the mean-field framework, this maximum principle differs from the classical one in that the first-order adjoint equation is formulated as a linear mean-field backward stochastic differential equation, whereas the second-order adjoint equation retains the same form as in Peng’s stochastic maximum principle.

By the end of the twentieth century, numerous numerical methods had been developed for solving stochastic optimal control problems. However, most of these approaches suffer from the well-known curse of dimensionality, which makes them impractical for high-dimensional settings, as their computational complexity increases exponentially with the dimension of the problem. Recent advances in deep learning have opened new perspectives for overcoming this limitation. Deep learning-based methods have achieved remarkable success in solving high-dimensional problems across a wide range of applications and have shown great potential for tackling complex stochastic control problems (see. [9], [10], [11]).

The main idea behind these works is to represent the control process by a deep neural network (DNN) and to determine the optimal set of network parameters using stochastic gradient descent (SGD) algorithms. Building upon this framework, several researchers have developed neural network-based approaches for solving stochastic optimal control problems. For instance, the authors in [12, 13] proposed deep learning algorithms based on dynamic programming principles to tackle such control problems. In [14], hybrid architectures combining feedforward and recurrent neural networks were introduced to solve a class of mean-field-type stochastic differential equations through the associated forward-backward stochastic differential equations (FBSDEs).

In the present work, we investigate an optimal control problem governed by a mean-field-type stochastic differential equation (SDE), whose coefficients may depend not only on the state process but also on certain functionals of its probability distribution from the perspective of the stochastic maximum principle (SMP) by employing deep learning techniques.

We mainly focus on the following stochastic optimal control problem, which was originally introduced and studied from an analytical viewpoint in [4, 5]

$$\inf_{u \in \mathcal{U}} \mathbb{E} \left[\int_0^T h(t, x_t, \mathbb{E}[\varphi(x_t)], u_t) dt + g(x_T, \mathbb{E}[\chi(x_T)]) \right] \quad (1)$$

Subject to

$$x_t = x_0 + \int_0^t b(s, x_s, \mathbb{E}[\psi(x_s)], u_s) ds + \int_0^t \sigma(s, x_s, \mathbb{E}[\phi(x_s)], u_s) dB_s$$

Where the functions h and g depend on the law of the state process. The coefficients b , σ , ψ , ϕ , φ , and χ are given functions, and $(B_t)_{t \geq 0}$ denotes a Brownian motion. The control process $(u_t)_{t \geq 0}$ takes its values in the admissible control set $\mathcal{U}$

Moreover, the cost functional is of mean-field type, as it depends on the distribution of the state process. This dependence introduces time inconsistency and may lead to a nonlinear performance criterion, in contrast to classical stochastic control problems, whose cost functionals are typically expressed as linear expectations.

According to Pontryagin's maximum principle, any optimal control corresponds to the optimal state variables around the Hamiltonian system, with the maximum condition of H with respect to the control variable u . This yields the so-called the extended Hamiltonian system, which is expressed in FBSDE form with a minimum condition [4, 14].

$$\begin{cases} dx_t^* = & H_p(t, x_t^*, u^*, p_t^*, q_t^*) dt + H_q(t, x_t^*, u^*, p_t^*, q_t^*) dB_t \\ dp_t^* = & -H_x(t, x_t^*, u^*, p_t^*, q_t^*) dt + q_t^* dB_t \\ x^*(0) = & x_0. \\ p_T^* = & g_x(T) + \mathbb{E}[g_y(T)\chi_x(T)]. \quad \text{There } y = \chi(x_T) \\ u^* = & \underset{v \in \mathcal{U}}{\operatorname{argmin}} H(t, x_t^*, v, p_t^*, q_t^*) \end{cases} \quad (2)$$

The main idea of our work is to employ deep learning to solve this type of mean-field stochastic control problems by reformulating the original problem into an new problem equivalent such that, we first reformulate (2) as a new optimal control problem [15]:

$$\inf_{\hat{p}_0 \{\hat{q}_t\}_{0 \leq t \leq T}} \mathbb{E} [| \hat{p}_T - g_x(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) - \mathbb{E}[g_y(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)])\chi_x(\hat{x}_T)] |^2] \quad (3)$$

Subject to

$$\begin{cases} \hat{x}_t = x_0 + \int_0^t b(s, \hat{x}_s, \mathbb{E}[\psi(\hat{x}_s)], \hat{u}_s) ds + \int_0^t \sigma(s, \hat{x}_s, \mathbb{E}[\phi(\hat{x}_s)], \hat{u}_s) dB_s \\ \hat{p}_t = p_0 - \int_0^t H_x(s, \hat{x}_s, \hat{u}_s, \hat{p}_s, \hat{q}_s) ds + \int_0^t \hat{q}_s dB_s \\ \hat{u}_t = \underset{v \in \mathcal{U}}{\operatorname{argmin}} H(t, \hat{x}_t, v, \hat{p}_t, \hat{q}_t), \end{cases} \quad (4)$$

where $s \in [0, t]$. The processes $\hat{q}_t$ and $\hat{p}_0$, as well as the initial state, are regarded as controls. The resulting optimization problem (3) involves a simple quadratic cost functional at the terminal time T , providing a more tractable framework for determining whether a state control pair is optimal, that is, to check whether $\mathbb{E} [| \hat{p}_T - g_x(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) - \mathbb{E}[g_y(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)])\chi_x(\hat{x}_T)] |^2]$ equals to 0.

However, the main challenge of this transformation lies in handling an additional constraint, namely the maximum condition. To address the resulting control problem (1.3), we develop three deep learning-based algorithms, each designed for a specific setting and offering distinct advantages in terms of accuracy, efficiency, and implementation.

In the first algorithm, a single deep neural network (DNN) is employed to approximate the control process $\hat{q}_t$. After training, the network provides an approximation of $\hat{q}_t$, which is then used to construct an approximate solution $(\hat{x}_t, \hat{p}_t, \hat{q}_t, \hat{u}_t)_{0 \leq t \leq T}$ of problem (3). To satisfy the maximum condition and determine the corresponding optimal control $\hat{u}_t$, the L-BFGS optimization method [16] is employed.

To further improve the computational efficiency of the approximation of the optimal control $\hat{u}_t$, we propose a second algorithm. The main idea consists in reformulating the maximum condition as the constraint

$$H_u(t, x, u, p, q) = 0.$$

Two neural networks are then introduced to approximate the control processes $(\hat{q}_t)_{0 \leq t \leq T}$ and $(\hat{u}_t)_{0 \leq t \leq T}$, respectively. Moreover, the integral of the constraint $H_u(t, x, u, p, q)$ over the interval $[0, T]$ is incorporated into the original loss function as a penalty term. This strategy considerably reduces the computational cost, especially in high-dimensional settings where the optimal control $\hat{u}_t$ cannot be obtained in closed form.

Furthermore, when the function $\bar{H}$ defined in (10) is explicitly available, a broad class of high-dimensional stochastic optimal control problems can be addressed, even if the optimal control $\hat{u}_t$ admits no explicit representation. In this situation, we introduce a third algorithm. It is worth noting that whenever $\hat{u}_t$ can be expressed explicitly, as in Algorithm (1), the function $\bar{H}$ can also be computed explicitly. Consequently, Algorithm (1) with an explicit expression of $\hat{u}_t$ may be viewed as a particular case of Algorithm (3).

The numerical results demonstrate the effectiveness and accuracy of the proposed algorithms in solving high-dimensional mean field stochastic optimal control problems. The remainder of this paper is organized as follows. Section 2 presents the general formulation of the stochastic optimal control problem governed by mean-field-type stochastic differential equations (SDEs) and introduces the main assumptions ensuring the existence and uniqueness of solutions. We then reformulate the original stochastic control problem into an equivalent control framework. In Section 3, we develop the proposed deep learning-based numerical methods for solving the reformulated problem and describe the neural network architectures employed. Finally, in Section 4, we illustrate the effectiveness of the proposed approach through the optimal mean variance portfolio selection problem. The three algorithms are implemented and compared in terms of accuracy and computational performance.

2 Mathematical Formulation

2.1 Statement of the problem

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, \mathbb{P})$ be a complete filtered probability space satisfying the usual conditions and supporting a standard Brownian motion $B = (B_t)_{t \geq 0}$. The filtration $(\mathcal{F}_t)_{t \geq 0}$ is assumed to be the augmented natural filtration generated by B . We consider

the following mean field stochastic differential equation:

$$\begin{cases} dx_t = b(t, x_t, \mathbb{E}[\psi(x_t)], u_t) dt + \sigma(t, x_t, \mathbb{E}[\phi(x_t)], u_t) dB_t \\ x(0) = x_0. \end{cases} \quad (5)$$

For any $u \in \mathcal{U}$ such that $\mathcal{U}$ denotes the class of measurable, $\mathcal{F}_t$ -adapted, and square-integrable processes

$u : [0, T] \times \Omega \rightarrow U$ and the action space U is a non empty, closed and convex subset of $\mathbb{R}$.

And where

$$\begin{aligned} b &: [0, T] \times \mathbb{R}^d \times \mathbb{R}^d \times U \longrightarrow \mathbb{R}^d \\ \sigma &: [0, T] \times \mathbb{R}^d \times \mathbb{R}^d \times U \longrightarrow \mathbb{R}^{d \times d} \\ \psi &: \mathbb{R}^d \longrightarrow \mathbb{R}^d \\ \phi &: \mathbb{R}^d \longrightarrow \mathbb{R}^d. \end{aligned}$$

The cost functional over the time horizon $[0, T]$ is of mean field type and is defined by

$$J(u) = \mathbb{E} \left[\int_0^T h(t, x_t, \mathbb{E}[\varphi(x_t)], u_t) dt + g(x_T, \mathbb{E}[\chi(x_T)]) \right]. \quad (6)$$

Where

$$\begin{aligned} g &: \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R} \\ h &: [0, T] \times \mathbb{R}^d \times \mathbb{R}^d \times U \longrightarrow \mathbb{R} \\ \chi &: \mathbb{R}^d \longrightarrow \mathbb{R}^d \\ \varphi &: \mathbb{R}^d \longrightarrow \mathbb{R}^d. \end{aligned}$$

We also use the following notation:

- $\langle \cdot, \cdot \rangle :=$ the usual scalar product in $\mathbb{R}^n$.
- $\|\cdot\| :=$ the Euclidean norm in $\mathbb{R}^n$, such that $\|Z\| = \{tr(ZZ^T)\}^{\frac{1}{2}}$.
- $M^2([0, T]; \mathbb{R}^n) :=$ the space of all $\mathcal{F}_t$ progressively measurable processes $v = (v_t)_{0 \leq t \leq T}$ with values in $\mathbb{R}^n$ such that

$$\mathbb{E} \left[\int_0^T |v_t|^2 dt \right] < \infty.$$

We consider now the following assumption:

Assumption 1

Assume that the functions b , σ , ψ , and ϕ are bounded and continuous. Moreover, suppose

that there exists a positive constant K such that, for any $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\begin{aligned} |b(t, x_1, y_1, u) - b(t, x_2, y_2, u)| &\leq K(|x_1 - x_2| + |y_1 - y_2|), \\ |\sigma(t, x_1, y_1, u) - \sigma(t, x_2, y_2, u)| &\leq K(|x_1 - x_2| + |y_1 - y_2|), \\ |\Psi(x_1) - \Psi(x_2)| &\leq K|x_1 - x_2|, \\ |\Phi(x_1) - \Phi(x_2)| &\leq K|x_1 - x_2|. \end{aligned}$$

Assumption 2

We assume that the functions h , g , χ , and ϕ are bounded and continuous. In addition, h satisfies a Lipschitz condition with respect to (x, y) : there exists a constant $K > 0$ such that, for every $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$|h(t, x_1, y_1, u) - h(t, x_2, y_2, u)| \leq K(|x_1 - x_2| + |y_1 - y_2|),$$

Theorem 1 [14]

Under Assumptions (1) and (2), the mean-field stochastic differential equation (MFSDE) (5) admits a unique strong solution. Furthermore, for every admissible control $u \in \mathcal{U}$, the state equation (6) possesses a unique solution, and the cost functional J is well defined as a mapping from $\mathcal{U}$ into $\mathbb{R}$. An admissible control $u^* \in \mathcal{U}$ is said to be optimal if it satisfies

$$J(u^*) = \inf_{v \in \mathcal{U}} J(v).$$

To simplify the presentation, we adopt the following notation throughout the paper. For each function $f \in \{b, \sigma, h\}$ and for each function $l \in \{\psi, \phi, \varphi, \chi\}$, we set

$$f_x(t) = \frac{\partial f}{\partial x}(t, x_t, \mathbb{E}[\psi(t)], u_t) \quad \text{and} \quad l_x(t) = l_x(x_t)$$

Proposition 1 [14]

If u^* denotes the optimal control for (5) and the corresponding trajectories are written as x^* , then the pair (p^*, q^*) represents the solution to the associated adjoint system.

$$\begin{cases} dp_t &= -(b_x p_t + \mathbb{E}[b_y(t)p_t]\psi_x(t) + \sigma_x(t)q_t + \mathbb{E}[\sigma_y(t)q_t]\phi_x(t) + \mathbb{E}[h_y(t)]\varphi_x(t) + h_x(t))dt + q_t dB_t \\ p_T &= g_x(T) + \mathbb{E}[g_y(T)\chi_x(T)]. \end{cases} \quad (7)$$

Finally, we define the Hamiltonian corresponding to the state variable x as follows:

$$H : [0, T] \times \mathbb{R}^n \times U \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$H(t, x, u, p, q) := h(t, x, \mathbb{E}[\varphi(x)], u) + b(t, x, \mathbb{E}[\psi(x)], u)p + \sigma(t, x, \mathbb{E}[\phi(x)], u)q$$

Let H be differentiable, and use its expression to express the adjoint system in terms of H

$$\begin{cases} dp_t &= -H_x(t, x, u, p, q)dt + q_t dB_t \\ p_T &= g_x(T) + \mathbb{E}[g_y(T)\chi_x(T)]. \end{cases} \quad (8)$$

Given that the Hamiltonian is differentiable, it follows that

$$\langle H_v(t, x_t^*, v, p_t^*, q_t^*), v - u^* \rangle \geq 0 \quad \forall v \in U \quad \text{i.e.} \quad u^* = \arg \min_{v \in U} H(t, x_t^*, v, p_t^*, q_t^*)$$

The combination of (5) and (7) can lead us to a stochastic Hamiltonian system, which can be written as follows:

$$\begin{cases} dx_t^* = & H_p(t, x_t^*, u^*, p_t^*, q_t^*)dt + H_q(t, x_t^*, u^*, p_t^*, q_t^*)dB_t \\ dp_t^* = & -H_x(t, x_t^*, u^*, p_t^*, q_t^*)dt + q_t^* dB_t \\ x^*(0) = & x_0. \\ p_T^* = & g_x(T) + \mathbb{E}[g_y(T)\chi_x(T)] \\ u^* = & \arg \min_{v \in \mathcal{U}} H(t, x_t^*, v, p_t^*, q_t^*) \end{cases} \quad \text{Therey} = \chi(x_T) \quad (9)$$

Using the uniqueness of u^* we can define a function $\hat{H}$, for all $(x_t^*, p_t^*, q_t^*) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$ such that

$$\bar{H}(t, x_t^*, p_t^*, q_t^*) = \min_{v \in \mathcal{U}} H(t, x_t^*, v, p_t^*, q_t^*) = H(t, x_t^*, u^*, p_t^*, q_t^*) \quad (10)$$

2.2 Problem formulation

We make use of the extended stochastic Hamiltonian system (9), expressed as a forward backward stochastic differential equation (FBSDE) coupled with an associated minimum condition. This framework enables us to reformulate the original optimal control problem (6) into an equivalent characterization derived from the Stochastic Maximum Principle (see. [7, 15]).

$$\begin{cases} d\hat{x}_t = & b(t, \hat{x}_t, \mathbb{E}[\psi(\hat{x}_t)], \hat{u}_t)dt + \sigma(t, \hat{x}_t, \mathbb{E}[\phi(\hat{x}_t)], \hat{u}_t)dB_t \\ d\hat{p}_t = & -H_x(t, \hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)dt + \hat{q}_t dB_t \\ \hat{x}_0 = & x_0 \quad \hat{p}_0 = p_0 \\ \hat{u}_t = & \arg \min_{v \in U} H(t, \hat{x}_t, v, \hat{p}_t, \hat{q}_t) \end{cases} \quad (11)$$

The control problem (1) is then reformulated as a new variational problem.

$$\inf_{\hat{p}_0 \in \mathcal{P}_0} \mathbb{E} [| \hat{p}_T - g_x(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) - \mathbb{E}[g_y(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)])\chi_x(\hat{x}_T)] |^2] \quad (12)$$

Subject to

$$\begin{cases} \hat{x}_t = & x_0 + \int_0^t b(s, \hat{x}_s, \mathbb{E}[\psi(\hat{x}_s)], \hat{u}_s)ds + \int_0^t \sigma(s, \hat{x}_s, \mathbb{E}[\phi(\hat{x}_s)], \hat{u}_s)dB_s \\ \hat{p}_t = & p_0 - \int_0^t H_x(s, \hat{x}_s, \hat{u}_t, \hat{p}_t, \hat{q}_s)ds + \int_0^t \hat{q}_s dB_s \\ \hat{u}_t = & \arg \min_{v \in U} H(t, \hat{x}_t, v, \hat{p}_t, \hat{q}_t) \end{cases}$$

Theorem 2 [17]

Assume that the functions g_x and g_y are continuous with respect to their arguments, the function $\bar{H}$ defined in (10) is continuously differentiable with respect to (x, p, q) . Moreover,

its partial derivatives are continuous in (x, p, q) and satisfy the following Lipschitz condition: there exists a constant $K > 0$ such that, for each

$$f \in \{\bar{H}_x, \bar{H}_p, \bar{H}_q\},$$

$$\begin{aligned} |f(t, x, p, q) - f(t, x', p', q')| &\leq K(|x - x'| + |p - p'| + |q - q'|), \\ |f(t, 0, 0, 0)|, |H_q(t, x, p, 0)| &\leq K, \\ x, x' &\in \mathbb{R}^d, \quad p, p' \in \mathbb{R}^d, \quad q, q' \in \mathbb{R}^{d \times d}. \end{aligned}$$

Then the FBSDE (11) admits a solution on $[0, T]$ if and only if (12) converges to 0.

Proposition 2 Suppose that the assumptions of Theorems 1 and 2 are satisfied. If there exists a solution

$$(\hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)_{0 \leq t \leq T}$$

to system (11) such that

$$\mathbb{E} \left[\left| \hat{p}_T - g_x(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) - \mathbb{E}[g_y(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)])\chi_x(\hat{x}_T)] \right|^2 \right] = 0. \quad (13)$$

Then $(\hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)_{0 \leq t \leq T}$ is a solution of (9). Furthermore, the cost functional 6 admits the following representation:

$$J(u^*(\cdot)) = J(\hat{u}(\cdot)) = \mathbb{E} \left[\int_0^T h(t, \hat{x}_t, \mathbb{E}[\varphi(\hat{x}_t)], \hat{u}_t) dt + g(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) \right]. \quad (14)$$

By virtue of Theorem 6 and Proposition 2, the optimal control u^* can be obtained through the solution of problem (12), showing that this reformulated problem is equivalent to the original one. Moreover, the quadratic terminal cost functional arising in the new formulation offers a convenient criterion for identifying the optimal state control pair $(\hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)_{0 \leq t \leq T}$. Our main focus is now to solve the problem (12) by means of deep learning methods.

3 Numerical Methods

In the previous section, we established sufficient optimality conditions based on the stochastic maximum principle and reformulated the Hamiltonian system associated with problem (5) as an equivalent variational problem.

In the present section, we develop three deep neural network (DNN)-based algorithms to solve the variational problem (12).

Prior to presenting these approximation schemes, we first discretize the stochastic Hamiltonian system (11), which serves as the foundation for the proposed numerical methods.

3.1 Algorithm 1: Numerical algorithm with 1-NN

Let τ be a partition of the time interval $[0, T]$ defined by

$$\tau : t_0 = 0 < t_1 < t_2 < \dots < t_{N-1} < t_N = T, \quad N \in \mathbb{N},$$

and let $\delta = \max_{0 \leq i \leq N-1} (t_{i+1} - t_i)$ denote the mesh size of this partition. For each $i = 0, 1, \dots, N-1$, define $\Delta t_i = t_{i+1} - t_i$ and $\Delta B_{t_i} = B_{t_{i+1}} - B_{t_i}$, where ΔB_{t_i} is a Gaussian random variable with distribution $\mathcal{N}(0, \Delta t_i)$.

Applying the Euler Maruyama discretization scheme to system (11), the maximum condition can be expressed as follows:

$$\begin{cases} \hat{x}_{t_{i+1}}^\tau = \hat{x}_{t_i}^\tau + b(t_i, \hat{x}_{t_i}^\tau, \mathbb{E}[\psi(\hat{x}_{t_i}^\tau)], \hat{u}_{t_i}^\tau) \Delta t_i + \sigma(t_i, \hat{x}_{t_i}^\tau, \mathbb{E}[\phi(\hat{x}_{t_i}^\tau)], \hat{u}_{t_i}^\tau) \Delta B_{t_i} \\ \hat{p}_{t_{i+1}}^\tau = \hat{p}_{t_i}^\tau - H_x(t_i, \hat{x}_{t_i}^\tau, \hat{u}_{t_i}^\tau, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau) \Delta t_i + \hat{q}_{t_i}^\tau \Delta B_{t_i} \\ \hat{x}_{t_0}^\tau = x_0, \quad \hat{p}_{t_0}^\tau = \hat{p}_0^\tau \\ \hat{u}_{t_i}^\tau = \arg \min_{u \in U} H(t_i, \hat{x}_{t_i}^\tau, u, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau) \end{cases} \quad (15)$$

We consider $(\hat{q}_{t_i}^\tau)_{0 \leq i < N}$, as a control and assume that it satisfies

$$\hat{q}_{t_i}^{l,\tau} = \mathfrak{R}(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_q^l)$$

To approximate the feedback control $\hat{q}_{t_i}^{l,\tau}$, we construct a deep neural network, in which the time variable t_i is used as an input. The network architecture consists of five layers: an input layer of dimension $(1 + d + d)$, three hidden layers, and an output layer of dimension $(d \times d)$. The collection of all trainable parameters of the network is denoted by θ . The associated loss function is defined as follows:

$$\text{Loss} = \frac{1}{M} \sum_{j=1}^M |\hat{p}_T^\tau - g_x(\hat{x}_T^\tau, \mathbb{E}[\chi(\hat{x}_T^\tau)]) - \mathbb{E}[g_y(\hat{x}_T^\tau, \mathbb{E}[\chi(\hat{x}_T^\tau)]) \chi_x(\hat{x}_T^\tau)]|^2. \quad (16)$$

Here, M denotes the number of simulated trajectories of the underlying Brownian motion. It is worth noting that the minimum condition leads to an optimization problem involving multivariable functions. In general, such problems do not admit closed-form solutions, making numerical optimization techniques necessary to approximate the optimal control. In this work, the corresponding optimization problem is solved numerically, and the overall procedure for solving the stochastic optimal control problem is summarized in Algorithm 1. The minimization condition results in a multivariate optimization problem that generally lacks an analytical solution. Consequently, numerical optimization techniques, such as the BFGS algorithm and its extensions [16], Sequential Least Squares Programming (SLSQP), and other gradient-based methods, are required to approximate the optimal control. In contrast, when the optimal control $\hat{u}_t$ admits an explicit representation, the Hamiltonian system (11) becomes equivalent to a forward backward stochastic differential equation (FBSDE). This particular setting is examined in Subsection 3.2.

Algorithm 1

Require: ΔB_{t_i} , initial parameters $(\theta_q^0, \hat{p}_0^{0,\tau})$, learning rate η

Ensure: The processes $\hat{x}_{t_i}^{l,\tau}$, $\hat{p}_{t_i}^{l,\tau}$ and $\hat{q}_{t_i}^{l,\tau}$

```
1: for  $l = 0$  to maxstep do
2:    $\hat{x}_{t_0}^{l,\tau} = x_0, \hat{p}_{t_0}^{l,\tau} = \hat{p}_0^{l,\tau}$ 
3:   for  $i = 0$  to  $N - 1$  do
4:      $\hat{q}_{t_i}^{l,\tau} = \mathfrak{R}(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_q^l)$ 
5:      $\hat{u}_{t_i}^{l,\tau} = \underset{u \in U}{\operatorname{argmin}} H(t_i, \hat{x}_{t_i}^{l,\tau}, u, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau})$ 
6:      $\alpha_i = \frac{1}{d} \sum_{j=1}^d \left[ \psi(\hat{x}_{t_i,j}^{l,\tau}) \right]$ 
7:      $\beta_i = \frac{1}{d} \sum_{j=1}^d \left[ \phi(\hat{x}_{t_i,j}^{l,\tau}) \right]$ 
8:      $\zeta_i = \frac{1}{d} \sum_{j=1}^d \left[ \varphi(\hat{x}_{t_i,j}^{l,\tau}) \right]$ 
9:      $\hat{x}_{t_{i+1}}^{l,\tau} = \hat{x}_{t_i}^{l,\tau} + b(t_i, \hat{x}_{t_i}^{l,\tau}, \alpha_i, \hat{u}_{t_i}^{l,\tau}) \Delta t_i + \sigma(t_i, \hat{x}_{t_i}^{l,\tau}, \beta_i, \hat{u}_{t_i}^{l,\tau}) \Delta B_{t_i}$ 
10:     $\hat{p}_{t_{i+1}}^{l,\tau} = \hat{p}_{t_i}^{l,\tau} - H_x(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{u}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau}) \Delta t_i + \hat{q}_{t_i}^{l,\tau} \Delta B_{t_i}$ 
11:     $\delta = \frac{1}{d} \sum_{j=1}^d \left[ \chi(\hat{x}_{t_i,j}^{l,\tau}) \right]$ 
12:  end for
13:   $J(\hat{u}_{(\cdot)}^{l,\tau}) = \frac{1}{M} \sum_{j=1}^M \left[ \frac{T}{N} \sum_{i=0}^{N-1} h(t_i, \hat{x}_{t_i}^{l,\tau}, \zeta_i, \hat{u}_{t_i}^{l,\tau}) + g(\hat{x}_T, \delta) \right]$ 
14:   $\text{Loss} = \frac{1}{M} \sum_{j=1}^M \left[ \left| \hat{p}_T^{l,\tau} - g_x(\hat{x}_T^{l,\tau}, \delta) - [g_y(\hat{x}_T^{l,\tau}, \delta) \chi_x(\hat{x}_T^{l,\tau})] \right|^2 \right]$ 
15:   $(\theta_q^{l+1}, \hat{p}_0^{l+1,\tau}) = (\theta_q^l, \hat{p}_0^{l,\tau}) - \eta \nabla \text{loss}$ 
16: end for
```

3.2 Algorithm 2 with explicit expression of $\bar{H}$

We now present a second algorithm for the same control problem. When the function $\bar{H}$ defined in (10) is known explicitly, the associated Hamiltonian system can be reformulated as a forward backward stochastic differential equation (FBSDE). Since this FBSDE satisfies the standard existence and uniqueness conditions [5, 18], it can be solved in high dimensions using deep learning-based numerical methods.

$$\begin{cases} \hat{x}_{t_{i+1}}^\tau = \hat{x}_{t_i}^\tau + \hat{H}_p(t_i, \hat{x}_{t_i}^\tau, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau) \Delta t_i + \hat{H}_q(t_i, \hat{x}_{t_i}^\tau, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau) \Delta B_{t_i} \\ \hat{p}_{t_{i+1}}^\tau = \hat{p}_{t_i}^\tau - \hat{H}_x(t_i, \hat{x}_{t_i}^\tau, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau) \Delta t_i + \hat{q}_{t_i}^\tau \Delta B_{t_i} \\ \hat{x}_{t_0}^\tau = x_0, \quad \hat{p}_{t_0}^\tau = \hat{p}_{t_0}^\tau \end{cases} \quad (17)$$

If the system (17) satisfies the monotonicity conditions introduced in Assumptions (1) and (2), then the corresponding problem can be solved in high dimensions using deep learning techniques developed for forward backward stochastic differential equations (FBSDEs). To this end, a single deep neural network (DNN) is employed to approximate q^* . So, the triplets (x^*, p^*, q^*) have been obtained, the optimal control u^* can be recovered through the minimum condition

$$u_t^* = \arg \min_{u \in U} H(t, x_t^*, u, p_t^*, q_t^*).$$

The corresponding pseudo-code is presented in Algorithm 2.

Algorithm 2

Require: ΔB_{t_i} , initial parameters $(\theta_q^0, \hat{p}_0^{0,\tau})$, learning rate η

Ensure: The processes $\hat{x}_{t_i}^{l,\tau}$, $\hat{p}_{t_i}^{l,\tau}$ and $\hat{q}_{t_i}^{l,\tau}$

```

1: for  $l = 0$  to maxstep do
2:    $\hat{x}_{t_0}^{l,\tau} = x_0, \hat{p}_0^{l,\tau} = \hat{p}_0^{0,\tau}$ 
3:   for  $i = 0$  to  $N - 1$  do
4:      $\hat{q}_{t_i}^{l,\tau} = \mathfrak{R}(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_q^l)$ 
5:      $\alpha_i = \frac{1}{d} \sum_{j=1}^d [\psi(\hat{x}_{t_i,j}^{l,\tau})]$ 
6:      $\beta_i = \frac{1}{d} \sum_{j=1}^d [\phi(\hat{x}_{t_i,j}^{l,\tau})]$ 
7:      $\zeta_i = \frac{1}{d} \sum_{j=1}^d [\varphi(\hat{x}_{t_i,j}^{l,\tau})]$ 
8:      $\hat{x}_{t_{i+1}}^{l,\tau} = \hat{x}_{t_i}^{l,\tau} + \hat{H}_p(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau}) \Delta t_i + \hat{H}_q(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau}) \Delta B_{t_i}$ 
9:      $\hat{p}_{t_{i+1}}^{l,\tau} = \hat{p}_{t_i}^{l,\tau} - \hat{H}_x(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau}) \Delta t_i + \hat{q}_{t_i}^{l,\tau} \Delta B_{t_i}$ 
10:   end for
11:    $\delta = \frac{1}{d} \sum_{j=1}^d [\chi(\hat{x}_{T,j}^{l,\tau})]$ 
12:    $\text{Loss} = \frac{1}{M} \sum_{j=1}^M (|\hat{p}_T^{l,\tau} - g_x(\hat{x}_T^{l,\tau}, \delta) - [g_y(\hat{x}_T^{l,\tau}, \delta) \chi_x(\hat{x}_T^{l,\tau})]|^2)$ 
13:    $(\theta_q^{l+1}, \hat{p}_0^{l+1,\tau}) = (\theta_q^l, \hat{p}_0^{l,\tau}) - \eta \nabla \text{loss}$ 
14: end for
15:  $\hat{u}_{t_i}^{l,\tau} = \underset{u \in \mathcal{U}}{\operatorname{argmin}} H(t_i, \hat{x}_{t_i}^{l,\tau}, u, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau})$ 
16:  $J(\hat{u}_{(\cdot)}^{l,\tau}) = \frac{1}{M} \sum_{j=1}^M \left[ \frac{T}{N} \sum_{i=0}^{N-1} h(t_i, \hat{x}_{t_i}^{l,\tau}, \zeta_i, \hat{u}_{t_i}^{l,\tau}) + g(\hat{x}_T, \delta) \right]$ 

```

A key advantage of this approach is that the above optimization problem needs to be solved only once, after the optimal state processes have been computed through the solution of the Hamiltonian system (17) by means of deep learning. This significantly reduces the overall computational cost compared with approaches in which the optimization problem must be solved repeatedly at each iteration or time step as algorithm 1.

3.3 Algorithm 3: Numerical algorithm with 2-NN

The previous algorithm relies on the explicit knowledge of the function $\bar{H}$ defined in (10). However, in many practical situations, obtaining an explicit expression of $\bar{H}$ is either difficult or impossible. To overcome this limitation, we develop a third numerical algorithm based on two deep neural networks.

We consider a class of stochastic optimal control problems for which the control domain is convex and given by $\mathcal{U} = \mathbb{R}^k$. Furthermore, all coefficients are assumed to be continuously differentiable with respect to the control variable u , and the optimal control $\hat{u}$ is assumed to belong to the interior of the control domain. Under these assumptions, the maximum principle implies that the optimal control satisfies the

first-order optimality condition

$$H_u(t, x, u, p, q) = 0 \quad \forall u \in \mathbb{U} \quad (18)$$

Consequently, instead of requiring an explicit expression of $\bar{H}$, we incorporate this condition directly into the learning procedure by adding a penalty term to the loss function and approximate simultaneously the processes $\hat{q}$ and $\hat{u}$ using two neural networks.

So, the system 9 can be reformulated as

$$\begin{cases} dx_t^* = & H_p(t, x_t^*, u^*, p_t^*, q_t^*)dt + H_q(t, x_t^*, u^*, p_t^*, q_t^*)dB_t \\ dp_t^* = & -H_x(t, x_t^*, u^*, p_t^*, q_t^*)dt + q_t^* dB_t \\ x^*(0) = & x_0. \\ p_T^* = & g_x(T) + \mathbb{E}[g_y(T) \chi_x(T)] & \text{Therey} = \chi(x_T) \\ & H_u(t, x_t^*, u^*, p_t^*, q_t^*) = 0 \quad \forall u \in \mathbb{U} \end{cases} \quad (19)$$

Following the methodology developed for the optimal control of fully coupled forward backward stochastic systems, the Hamiltonian system (19) can be solved by reformulating it as the following variational control problem, which is particularly suitable for deep learning-based numerical approximation:

$$\inf_{\hat{p}_0, \{\hat{q}_t, \hat{u}_t\}_{0 \leq t \leq T}} \mathbb{E} \left[\left| \hat{p}_T - g_x(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)]) - \mathbb{E}[g_y(\hat{x}_T, \mathbb{E}[\chi(\hat{x}_T)])\chi_x(\hat{x}_T)] \right|^2 + \lambda \int_0^T [H_u(t, \hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)]^2 dt \right] \quad (20)$$

Subject to

$$\begin{aligned} \hat{x}_t &= x_0 + \int_0^t b(s, \hat{x}_s, \mathbb{E}[\psi(\hat{x}_s)], \hat{u}_s)ds + \int_0^t \sigma(s, \hat{x}_s, \mathbb{E}[\phi(\hat{x}_s)], \hat{u}_s)dB_s \\ \hat{p}_t &= p_0 - \int_0^t H_x(s, \hat{x}_s, \hat{u}_s, \hat{p}_s, \hat{q}_s)ds + \int_0^t \hat{q}_s dB_s \end{aligned}$$

where $\hat{p}_0$, $\{\hat{q}_t\}_{0 \leq t \leq T}$, and $\{\hat{u}_t\}_{0 \leq t \leq T}$ serve as the control variables of the new problem, and λ denotes a hyper-parameter. Provided that the cost functional (20) converges to zero, the quadruple $(\hat{x}_t, \hat{u}_t, \hat{p}_t, \hat{q}_t)$ converges to the optimal solution $(x_t^*, u_t^*, p_t^*, q_t^*)$. As in the previous algorithms, we employ a discrete Euler scheme, where τ denotes a partition of the time interval:

$$\begin{cases} \hat{x}_{t_{i+1}}^\tau = \hat{x}_{t_i}^\tau + b(t_i, \hat{x}_{t_i}^\tau, \mathbb{E}[\psi(\hat{x}_{t_i}^\tau)], \hat{u}_{t_i}^\tau)\Delta t_i + \sigma(t_i, \hat{x}_{t_i}^\tau, \mathbb{E}[\phi(\hat{x}_{t_i}^\tau)], \hat{u}_{t_i}^\tau)\Delta B_{t_i} \\ \hat{p}_{t_{i+1}}^\tau = \hat{p}_{t_i}^\tau - H_x(t_i, \hat{x}_{t_i}^\tau, \hat{u}_{t_i}^\tau, \hat{p}_{t_i}^\tau, \hat{q}_{t_i}^\tau)\Delta t_i + \hat{q}_{t_i}^\tau \Delta B_{t_i} \\ \hat{x}_{t_0}^\tau = x_0, \quad \hat{p}_{t_0}^\tau = \hat{p}_0^\tau \end{cases} \quad (21)$$

To approximate the control processes $\hat{u}_t$ and $\hat{q}_t$, we introduce two deep neural networks. The controls $\hat{u}_{t_i}^\tau$ and $\hat{q}_{t_i}^\tau$ are viewed as feedback policies parameterized by neural networks, whose inputs are the current time t_i and the state variables $(\hat{x}_{t_i}^\tau, \hat{p}_{t_i}^\tau)$. Rather

than constructing a separate network at each time step, we employ a single shared network over the entire time horizon for each control, with parameters denoted by θ .

$$\hat{u}_{t_i}^{l,\tau} = \mathfrak{R}^1(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_u^l) \quad \text{and} \quad \hat{q}_{t_i}^{l,\tau} = \mathfrak{R}^2(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_q^l) \quad (22)$$

Each of the two neural networks (2-NNets) is composed of an input layer of dimension $(2d + 1)$, three hidden layers, and an output layer. The output dimension is $d \times d$ for $\hat{q}_{t_i}^{l,\tau}$ and d for $\hat{u}_{t_i}^{l,\tau}$. The loss function is defined in (20), and the corresponding pseudo-code is given in Algorithm 3.

Algorithm 3

Require: ΔB_{t_i} , initial parameters $(\theta_q^0, \hat{p}_0^{0,\tau})$, learning rate η

Ensure: The processes $\hat{x}_{t_i}^{l,\tau}$, $\hat{p}_{t_i}^{l,\tau}$ and $\hat{q}_{t_i}^{l,\tau}$

```

1: for  $l = 0$  to maxstep do
2:    $\hat{x}_{t_0}^{l,\tau} = x_0$ ,  $\hat{p}_0^{l,\tau} = \hat{p}_0^{l,\tau}$  and  $H = 0$ 
3:   for  $i = 0$  to  $N - 1$  do
4:      $\hat{u}_{t_i}^{l,\tau} = \mathfrak{R}^1(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_u^l)$ 
5:      $\hat{q}_{t_i}^{l,\tau} = \mathfrak{R}^2(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \theta_q^l)$ 
6:      $\alpha_i = \frac{1}{d} \sum_{j=1}^d [\psi(\hat{x}_{t_i,j}^{l,\tau})]$ 
7:      $\beta_i = \frac{1}{d} \sum_{j=1}^d [\phi(\hat{x}_{t_i,j}^{l,\tau})]$ 
8:      $\zeta_i = \frac{1}{d} \sum_{j=1}^d [\varphi(\hat{x}_{t_i,j}^{l,\tau})]$ 
9:      $\hat{x}_{t_{i+1}}^{l,\tau} = \hat{x}_{t_i}^{l,\tau} + b(t_i, \hat{x}_{t_i}^{l,\tau}, \alpha_i, \hat{u}_{t_i}^{l,\tau})\Delta t_i + \sigma(t_i, \hat{x}_{t_i}^{l,\tau}, \beta_i, \hat{u}_{t_i}^{l,\tau})\Delta B_{t_i}$ 
10:     $\hat{p}_{t_{i+1}}^{l,\tau} = \hat{p}_{t_i}^{l,\tau} - H_x(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{u}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau})\Delta t_i + \hat{q}_{t_i}^{l,\tau} \Delta B_{t_i}$ 
11:     $H = H + H_u(t_i, \hat{x}_{t_i}^{l,\tau}, \hat{u}_{t_i}^{l,\tau}, \hat{p}_{t_i}^{l,\tau}, \hat{q}_{t_i}^{l,\tau})^2$ 
12:  end for
13:   $\delta = \frac{1}{d} \sum_{j=1}^d [\chi(\hat{x}_{T,j}^{l,\tau})]$ 
14:   $J(\hat{u}_{(\cdot)}^{l,\tau}) = \frac{1}{M} \sum_{j=1}^M \left[ \frac{T}{N} \sum_{i=0}^{N-1} h(t_i, \hat{x}_{t_i}^{l,\tau}, \zeta_i, \hat{u}_{t_i}^{l,\tau}) + g(\hat{x}_T, \delta) \right]$ 
15:   $Loss = \frac{1}{M} \sum_{j=1}^M \left| \hat{p}_T^{l,\tau} - g_x(\hat{x}_T^{l,\tau}, \delta) - \left[ g_y(\hat{x}_T^{l,\tau}, \delta) \chi_x(\hat{x}_T^{l,\tau}) \right] \right|^2 + \lambda H.$ 
16:   $(\theta_q^{l+1}, \theta_u^{l+1}, \hat{p}_0^{l+1,\tau}) = (\theta_q^l, \theta_u^l, \hat{p}_0^{l,\tau}) - \eta \nabla loss$ 
17: end for
```

Algorithm 3 can efficiently handle a large class of high-dimensional stochastic optimal control problems, even when the optimal control is not available in closed form. Rather than solving the extremum condition approximately, the algorithm only requires enforcing the first-order optimality condition $H_u(t, x, u, p, q) = 0$, which is incorporated as a penalty term in the loss function.

4 Results and Discussions

In this section, we illustrate the performance of the proposed algorithms by applying them to the optimal mean variance portfolio selection problem. We consider a financial market consisting of two continuously traded assets: a risk-free asset (bank account) and a risky asset. Let N_t^0 and N_t^1 denote their respective prices at time t . The dynamics of these asset prices are governed by the following equations:

$$\begin{cases} dN_t^0 = \rho_t N_t^0 dt, \\ dN_t^1 = \mu_t N_t^1 dt + \sigma_t N_t^1 dB_t. \end{cases} \quad (23)$$

Let u_t denote the amount invested in the risky asset at time t . Under the self-financing assumption, the corresponding wealth process $(x_t)_{0 \leq t \leq T}$ satisfies

$$\begin{cases} dx_t = (\rho_t x_t + (\mu_t - \rho_t)u_t)dt + \sigma_t u_t dB_t, \\ x(0) = x_0. \end{cases} \quad (24)$$

where μ_t , σ_t , and ρ_t are bounded deterministic functions representing the market coefficients. The classical mean variance portfolio selection problem aims to determine an admissible control process that minimizes the variance of the terminal wealth while achieving a prescribed expected terminal wealth level. More precisely, the objective is to minimize

$$\text{Var}(x_T) = \mathbb{E}[x_T^2] - (\mathbb{E}[x_T])^2,$$

subject to the constraint

$$\mathbb{E}[x_T] = a,$$

where a is a given target level of expected terminal wealth. Various methods have been proposed in the literature to solve this problem [19, 20]. In the present work, we adopt the equivalent formulation obtained from the sufficient optimality conditions. The resulting objective functional is

$$J(u) = \mathbb{E}\left[\frac{\gamma}{2}x_T^2 - x_T\right] - \frac{\gamma}{2}(\mathbb{E}[x_T])^2,$$

where $\gamma > 0$ denotes the risk-aversion coefficient.

We observe that this cost functional is of the form (6) with $h = 0$, and the corresponding Hamiltonian of the system is given by

$$H(t, x, u, p, q) = (\rho_t x + (\mu_t - \rho_t)u)p + \sigma_t uq. \quad (25)$$

The adjoint process satisfies the backward stochastic differential equation

$$\begin{cases} dp_t = -\rho_t p_t dt + q_t dB_t, & t \in [0, T], \\ p_T = \gamma(x_T - \mathbb{E}[x_T]) - 1. \end{cases} \quad (26)$$

And the corresponding Hamiltonian system takes the following form:

$$\begin{cases} dx_t^* = [\rho_t x_t^* + (\mu_t - \rho_t) u_t^*] dt + \sigma_t u_t^* dB_t \\ dp_t^* = -\rho_t p_t^* dt + q_t^* dB_t \\ p_T^* = \gamma(x_T^* - \mathbb{E}[x_T^*]) - 1 \\ x^*(0) = x_0 \\ u_t^* = \arg \min_{v \in U} H(t, x_t^*, v, p_t^*, q_t^*) \end{cases} \quad (27)$$

Before starting to solve the example numerically, we first present the variational problem that corresponds to our original problem:

$$\inf_{p_0, \{q_t\}_{0 \leq t \leq T}} \mathbb{E} [|p_T - \gamma(x_T - \mathbb{E}[x_T]) + 1|^2] \quad (28)$$

$$\begin{cases} x_t = x_0 + \int_0^t (\rho_s x_s + (\alpha_s - \rho_s) u_s) ds + \int_0^t \sigma_s u_s dB_s \\ p_t = p_0 - \int_0^t \rho_s p_s ds + \int_0^t q_s dB_s \\ x(0) = x_0 \\ u_t = \arg \min_{v \in U} H(t, x_t, v, p_t, q_t) \end{cases} \quad (29)$$

In this application, only the cost functional J is of mean-field type, whereas neither the state equation nor the adjoint equations involve mean field interactions. Consequently, several functions related to the mean field terms vanish and are omitted in the implementation, namely

$$\psi = \varphi = \phi = 0.$$

We investigate this mean variance portfolio optimization problem in order to assess the performance and effectiveness of the proposed algorithms.

The numerical simulations are carried out with the following parameters: a risk free rate $\rho = 0.05$, an expected return $\mu = 0.10$, a volatility $\sigma = 0.20$, a risk-aversion coefficient $\gamma = 2.0$, an investment horizon $T = 1$, an initial wealth $x_0 = 1.0$, and an adjusted Sharpe ratio $\lambda = 0.0625$.

All networks are 3-hidden-layer with 64 units and Tanh activations. Tanh is preferred for BSDE problems: it is smooth, bounded and avoids saturation in the tail regions of the state space.

To compare the convergence properties of the three proposed algorithms, we plot the training loss as a function of the epoch number over 8000 training iterations. Algorithm 1 minimizes the objective functional (28) using a regularized control approximation. Algorithm 2 relies on the same loss function but exploits the closed-form expression of the optimal control given by [5] to solve the corresponding FBSDE. In contrast, Algorithm 3 augments the loss function with the penalty term $\lambda \int_0^T |H_u(t)|^2 dt$, thereby enforcing the first-order optimality condition directly within the learning procedure.

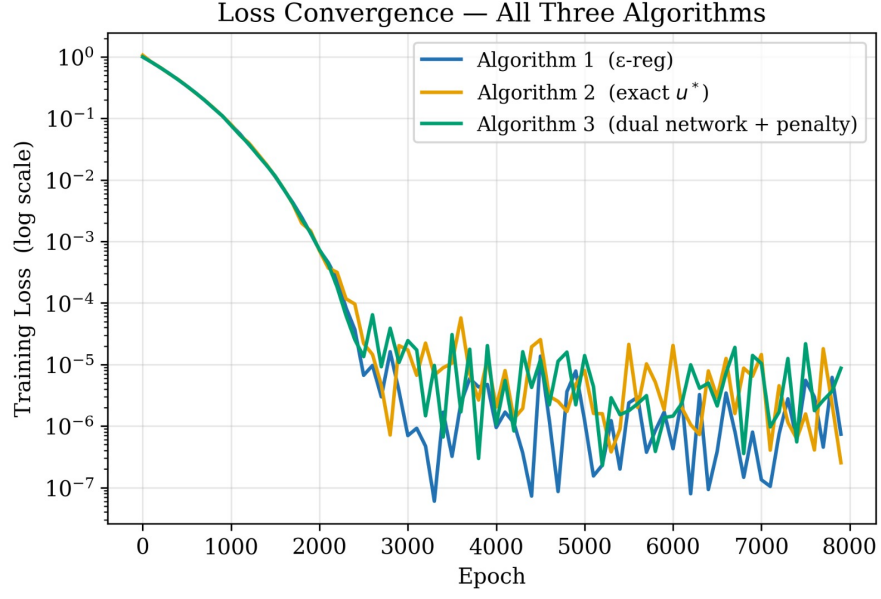


Fig. 1 Training Loss Convergence over Iterations for All Algorithms

We also plot the evolution of the estimated portfolio cost functional $J(u)$ during the training process for the three proposed algorithms, together with the analytical optimal value J^* .

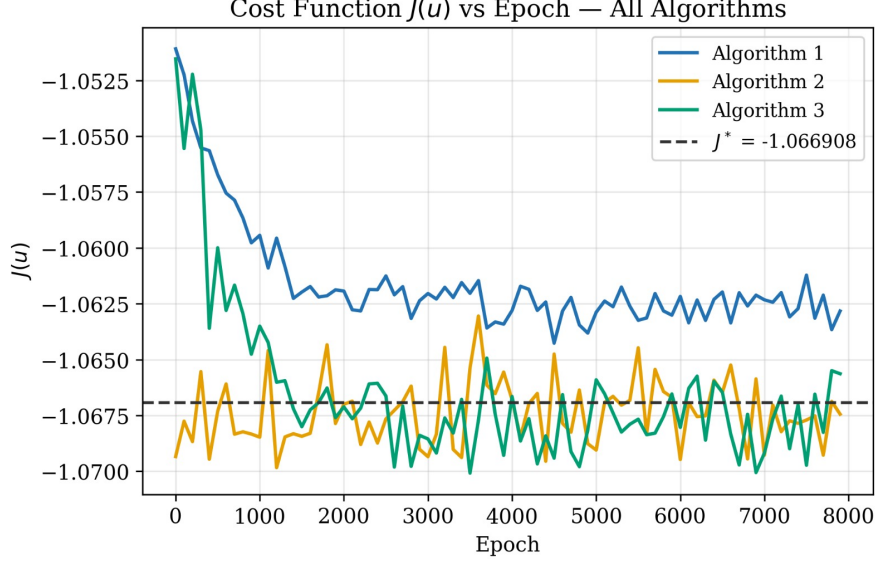


Fig. 2 Convergence of the Estimated Cost Functional $J(u)$ to the Analytical Optimal Value J^* for the Three Proposed Algorithms

As illustrated in Figure 2, the estimated cost functionals produced by the three algorithms converge toward the analytical optimal value J^* . However, Algorithm 2 consistently provides the most accurate approximation and achieves the highest level of accuracy among the proposed methods, yielding the smallest relative error. This superior performance can be attributed to the use of the explicit optimal control, which simplifies the learning task and enhances the convergence properties of the numerical scheme.

Algorithm 3 also delivers a highly accurate approximation. Indeed, its estimated cost remains very close to the analytical optimal value J^* . This improved performance is due to the inclusion of the penalty term

$$\lambda \int_0^T |H_u(t, x, u, p, q)|^2 dt,$$

which explicitly enforces the first-order optimality condition derived from the stochastic maximum principle. This additional constraint guides the training process toward solutions that better satisfy the optimality conditions, thereby improving both the accuracy and the stability of the approximation. In contrast, Algorithm 1 exhibits the lowest level of accuracy among the three methods, as reflected by its larger deviation from the analytical optimal value J^* . Nevertheless, it still converges to a satisfactory approximation and may serve as a viable alternative when the explicit optimal control is unavailable or when the additional penalty term is not incorporated into the learning procedure.

We now compare the numerical solutions obtained by each algorithm with the corresponding analytical solution. The following figure illustrates the convergence behavior of Algorithm 1 with $\varepsilon = 0.1$.

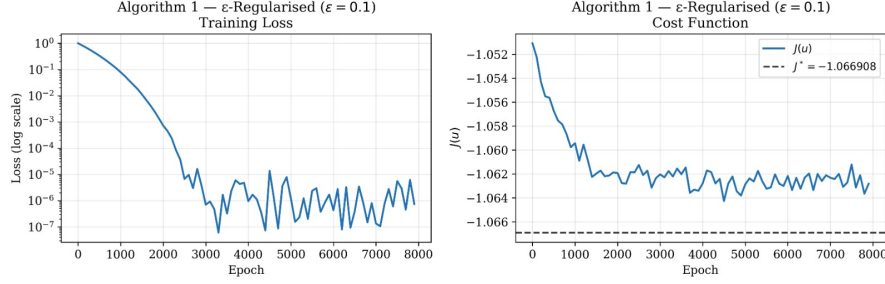


Fig. 3 Training loss (left) and evolution of the estimated cost functional toward the analytical optimum J^* (right) over the training iterations for Algorithm 1

The discrepancy between the numerical and analytical solutions obtained with Algorithm 1 is due to the ε -regularization error, which vanishes as $\varepsilon \rightarrow 0$. However, smaller values of ε generally require a significantly larger number of training iterations to achieve stable convergence. Consequently, Algorithm 1 converges more slowly than Algorithms 2 and 3

Algorithm 2 benefits from the availability of the analytical optimal control given by [5]. As a result, the associated FBSDE can be solved directly, and the neural network only needs to learn the adjoint process $\hat{q}_t$. This significantly simplifies the optimization procedure and improves convergence toward the analytical optimum of J^* . Compared with Algorithm 1, a more accurate and stable convergence is observed, as shown in the following figure.

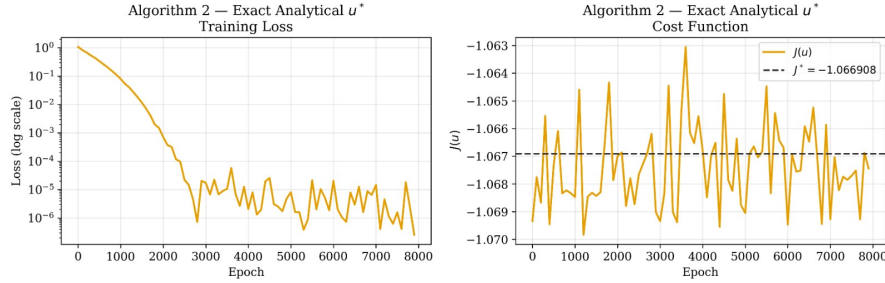


Fig. 4 Training loss (left) and evolution of the estimated cost functional toward the analytical optimum J^* (right) over the training iterations for Algorithm 2.

Algorithm 3 simultaneously trains two neural networks to approximate the optimal control process $\hat{u}_t$ and the adjoint diffusion process $\hat{q}_t$. The loss function incorporates

the first-order optimality condition derived from the stochastic maximum principle. Consequently, the proposed algorithm does not require an explicit analytical expression for the optimal control u^* .

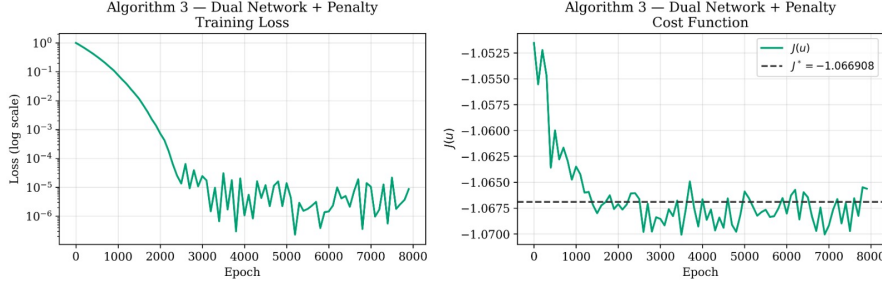


Fig. 5 Training loss (left) and evolution of the estimated cost functional toward the analytical optimum J^* (right) over the training iterations for Algorithm 3

The following table presents a comparison between the analytical solution and the numerical results obtained by Algorithms 1, 2, and 3. For each algorithm, the corresponding relative error with respect to the analytical solution and the final training loss are also reported. These results provide a quantitative assessment of the accuracy and convergence performance of the proposed approaches.

Algorithm	$J(u)$	J^*	Relative Error	Final Loss
Algorithm 1	-1.062827	-1.066908	0.382%	7.39e-07
Algorithm 2	-1.067443	-1.066908	0.050%	2.53e-07
Algorithm 3	-1.065633	-1.066908	0.119%	8.72e-06

Fig. 6 Summary of Numerical Results and Relative Errors

From Table 6, several conclusions can be drawn. First, Algorithm 2 achieves the highest level of accuracy among the three methods, yielding the smallest relative error and the closest approximation to the analytical solution. This superior performance can be attributed to the use of the explicit optimal control, which simplifies the learning task and improves the convergence of the numerical scheme.

The results also highlight the effectiveness and robustness of the framework proposed in Algorithm 3. Although its accuracy is slightly lower than that of Algorithm 2, it remains highly competitive while offering a much greater degree of flexibility. In particular, Algorithm 3 does not require any explicit analytical expression of the optimal control and accurately reproduces both the optimal wealth trajectories and the time evolution of the control process. Consequently, it constitutes the most general approach among the three proposed methods.

Overall, these results confirm the effectiveness of the proposed deep learning framework for solving stochastic optimal control problems. They also demonstrate the potential

of Algorithm 3 for tackling high-dimensional mean field stochastic control problems, where analytical solutions or explicit optimal controls are generally unavailable.

References

- [1] Lasry, J.M., Lions, P.L.: Mean field games. *Japan. J. Math* **2**, 229–260 (2007) <https://doi.org/10.1007/s11537-007-0657-8>
- [2] Buckdahn, J.L.: Stochastic differential games and viscosity solutions of hamilton jacobi bellman isaacs equations. *SIAM. J. Control. Optim.* **47**(1), 444–475 (2008) <https://doi.org/10.1137/060671954>
- [3] Méléard, S.: Asymptotic behaviour of some interacting particle systems; mckean–vlasov and boltzmann models. In: *Probabilistic Models for Nonlinear PDE's*, Lecture Notes in Mathematics, pp. 42–95. Springer, Berlin, Heidelberg, New York (2006). <https://doi.org/10.1007/BFb0093177>
- [4] Buckdahn, R., Djehiche, B., Li, J., Peng, S.: Mean-field backward stochastic differential equations: a limit approach. *Ann. Probab.* **37**(4), 1524–1565 (2009) <https://doi.org/10.1214/08-AOP442>
- [5] Andersson, D., Djehiche, B.: A maximum principle for sdes of mean-field type. *Appl. Math. Optim.* **63**(3), 341–356 (2011) <https://doi.org/10.1007/s00245-010-9123-8>
- [6] Pontryagin, L.S.: *Mathematical Theory of Optimal Processes*. CRC Press, Taylor and Francis Group, vol. 4 (2018)
- [7] Peng, S.: A general stochastic maximum principle for optimal control problems. *SIAM J. Control Optim* **2**(4), 966–979 (1990) <https://doi.org/10.1137/0328054>
- [8] Bismut, J.M.: An introductory approach to duality in optimal stochastic control. *SIAM Review* **20**(1), 62–78 (2007) <https://doi.org/10.1137/1020004>
- [9] Weinan, E., Jiequn, H., Arnulf, J.: 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) <https://doi.org/10.1007/s40304-017-0117-6>
- [10] Goodfellow, I., Bengio, Y., Courville, A.: Deep feedforward networks. *Deep learning* **1**, 161–217 (2016)
- [11] Oulghazi, H.: A new approach for solving a control stochastic problem driven by a diffusion process with jumps. *Mathematical Modeling and Computing* **13**(1), 77–90 (2026) <https://doi.org/10.23939/mmc2026.01.077>

- [12] Huré, C., Pham, H., Bachouch, A., Langrené, N.: Deep neural networks algorithms for stochastic control problems on finite horizon convergence analysis. *SIAM Journal on Num. Anal.* **59**(1), 525–557 (2021) <https://doi.org/10.1137/20M1316640>
- [13] Bachouch, A., Huré, C., Langrené, N., Pham, H.: Deep Neural Networks Algorithms for Stochastic Control Problems on Finite Horizon: Numerical Applications, (2021). <https://doi.org/10.1007/s11009-019-09767-9>
- [14] Andersson, D., Djehiche, B.: A maximum principle for sdes of mean-field type. *Applied Mathematics and Optimization* **63**(3), 341–356 (2011) <https://doi.org/10.1007/s00245-010-9123-8>
- [15] Ji, S., Peng, S., Peng, Y., Zhang, X.: deep learning method for solving stochastic optimal control problems driven by fully coupled fbsdes. *Zhongtai Securities Institute for Financial Studies, Shandong University* (2022) <https://doi.org/10.48550/arXiv.2204.05796>
- [16] Liu, D.C., Noceda, J.I.: On the limited memory method for large scale optimization. *Mathematical Programming* **45**(1), 503–528 (1989)
- [17] Ma, J., Yong, J.: Forward Backward Stochastic Differential Equations and Their Applications. *Lecture Notes in Mathematics*, vol. 1702. Springer, Berlin, Heidelberg (1999). <https://doi.org/10.1007/978-3-540-48831-6-8>
- [18] El Karoui, N., Peng, S., Quenez, M.C.: Backward stochastic differential equations in finance. *Mathematical finance* **7**(1), 1–71 (1997) <https://doi.org/10.1111/1467-9965.00022>
- [19] Zhou, X.Y., Li, D.: Continuous-time mean-variance portfolio selection: a stochastic lq framework. *Appl. Math. Optim.* **42**, 19–33 (2000) <https://doi.org/10.1007/s002450010003>
- [20] Basak, S., Chabakauri, G.: Dynamic mean variance asset allocation. *The Review of Financial Studies* **23**(8), 2970–3016 (2010) <https://doi.org/10.1093/rfs/hhq028>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [HighlightsDNNMFSOCP.txt](#)