mathematics

MDPI

Article

Martingale Deep Reinforcement Learning for High-Dimensional Financial Problems

Hang Qi [1], Dingli Xia [1], Mingxiang Deng [2,*] and Xiaofei Guan [1]

[1] School of Mathematical Sciences, Tongji University, Shanghai 200092, China; 2111155@tongji.edu.cn (H.Q.); 2533735@tongji.edu.cn (D.X.); guanxf@tongji.edu.cn (X.G.)
[2] School of Mathematics and Information of Science, Guangzhou University, Guangzhou 510006, China
* Correspondence: dmx@gzhu.edu.cn

Abstract

We develop a martingale-residual framework for continuous-time reinforcement learning in high-dimensional financial problems, linking policy evaluation (PE), policy gradient (PG), and entropy-regularized Actor–Critic (AC) learning. First, continuous-time PE and PG are often developed through separate constructions. We formulate a common martingale residual for the Critic and Actor, and show that, as the local time step vanishes, its rescaled policy derivative recovers the classical continuous-time PG. We also quantify how Critic approximation errors affect policy-gradient accuracy. Second, martingale-orthogonality temporal-difference (TD) methods require prescribed test functions, and their performance can depend on that choice. We introduce a weak-martingale adversarial formulation in which the test function is learned rather than prescribed. The resulting minimax problem avoids the sensitivity to hand-chosen test functions and provides a unified computational representation of the associated moment conditions. Third, direct high-dimensional Hamilton–Jacobi–Bellman (HJB) implementations can require explicit Hessians and other higher-order differential operators. Our algorithms work directly with sampled martingale residuals and first-order automatic differentiation, yielding Hessian-free implementations for both offline training from complete trajectories and online updating from sequential transitions. Numerical experiments on high-dimensional basket-option pricing and mean–variance portfolio selection demonstrate accurate PE and effective PG. In high-dimensional mean–variance portfolio-selection experiments, the proposed method achieves competitive or superior performance on several key metrics relative to benchmarks: the offline variant achieves a mean annualized return of 16.223%, the online variant attains a mean Sharpe ratio of 2.337, and fixed-budget training time increases by 19.3% as the asset dimension grows from 10 to 1000.

Keywords: high-dimensional reinforcement learning; martingale; policy evaluation; policy gradient; Actor–Critic; adversarial network

MSC: 93E35; 91G60

Check for updates

Academic Editor: Maria C. Mariani

Received: 30 September 2026
Revised: 26 October 2026
Accepted: 29 October 2026
Published: 3 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Reinforcement learning (RL) provides a data-driven approach to sequential decision making under uncertainty and has become increasingly relevant to financial problems involving dynamic portfolio choice, derivative valuation, and stochastic control [1,2]. In continuous time, these problems are naturally formulated through controlled diffusion

processes. Classical stochastic-control methods characterize optimal decisions through Hamilton–Jacobi–Bellman (HJB) equations, but their direct use becomes difficult when the underlying drift and volatility are unknown or when the state dimension is large [3,4]. In particular, grid-based PDE methods deteriorate rapidly with dimension, while model-based implementations require explicit knowledge of the infinitesimal generator and, in many cases, derivatives of the value function that are expensive to evaluate in high-dimensional systems [5,6].

A natural alternative is to learn directly from sampled trajectories. Discretizing a continuous-time problem allows one to apply standard discrete-time RL algorithms [7], but the resulting procedures may be sensitive to the time discretization: a fine grid substantially increases computational cost, whereas a coarse grid can distort the underlying continuous-time dynamics [8,9]. More fundamentally, the deterministic continuous-time temporal-difference (TD) construction of [10] does not extend to diffusion systems as a pathwise squared-residual objective: local increments contain a quadratic-variation component induced by diffusion noise rather than only value-function approximation error [11]. This established observation motivates formulations that exploit the martingale structure of continuous-time value processes rather than pathwise TD errors.

Recent work has pursued both statistical guarantees and scalable function approximation. For instance, Li et al. established minimax-optimal sample complexity for synchronous Q-learning [12], while Mao et al. proposed a model-free nonstationary RL algorithm with near-optimal regret [13]. Deep learning approaches use neural networks to approximate value functions or policies in high-dimensional systems, including mean-field reinforcement learning, dynamic pricing, offline zero-sum Markov games, and reflected Brownian motion control [14–17]. Despite these advances, model dependence, sensitivity to stochastic noise, fixed test functions in moment-based methods, and the cost of high-order differential operators remain important obstacles for high-dimensional continuous-time problems [18–20].

Martingale methods provide an alternative formulation. For a fixed policy, the Feynman–Kac representation implies that the appropriately compensated value process is a martingale, allowing value estimation from sampled trajectories [21]. This observation has been used to construct continuous-time policy evaluation (PE) methods based on martingale loss and martingale orthogonality. Subsequent work has extended continuous-time RL to randomized policies and entropy regularization and derived policy gradient (PG) representations for stochastic control problems [11,22,23]. These developments provide ingredients for continuous-time RL. In the present implementation, policy evaluation can use observed trajectories, whereas the Actor's pathwise derivative uses a differentiable relaxed transition model. Separate PE and PG constructions, test-function choices, and Hessian costs motivate the framework below.

This paper addresses these issues through a common martingale-residual viewpoint. Starting from the martingale characterization of the value function, we use short-interval martingale increments as the common object underlying both PE and PG. For PE, the short-interval martingale condition is rewritten in weak form and enforced through an adversarially learned test function. For randomized policies, the same martingale residual is differentiated with respect to the policy parameters, leading to a stochastic PG estimator. Under an exact current-policy Critic and the stated regularity conditions, we show that the rescaled population derivative approaches the continuous-time PG as the time step vanishes. This connects the ideal Critic residual with the Actor update. The learned-Critic error identity and the local analysis of anchored adversarial updates quantify the approximation and optimization effects in this connection.

Our work builds directly on the continuous-time martingale RL framework of [11,22,23]. In particular, martingale-loss PE, martingale-orthogonality conditions, and entropy-regularized randomized policies are established ingredients used in our development. The contributions of the present paper lie in the following three extensions.

- *A unified martingale-residual framework for continuous-time PE, PG, and Actor–Critic (AC) learning.* The central theoretical result establishes asymptotic equivalence between the rescaled population residual derivative and the classical continuous-time PG under an exact Critic as the local interval tends to zero. We also quantify how Critic approximation errors affect policy-gradient accuracy. These results establish an explicit link between martingale-based value estimation by the Critic and the PG update by the Actor, providing a coherent basis for the proposed AC construction.
- *A weak-martingale adversarial formulation with a learnable test function.* We rewrite the short-interval martingale condition as a weak moment condition and formulate PE as a minimax problem over the value function and a test-function class. Parameterizing the test function by an adversarial network replaces the hand-chosen test functions used in martingale-orthogonality TD-type methods by a learned test function. We establish local stability of the alternating Critic/Test updates under the stated conditions. The one-dimensional benchmark illustrates why this distinction matters: different fixed test functions can generate different moment-equation fixed points, whereas the adversarial formulation can adapt the test function toward the value-approximation objective.
- *Hessian-free implementation for offline and online learning.* The proposed weak-martingale Critic and martingale-residual policy gradient are implemented through sampled state transitions, neural-network forward/backward propagation, and first-order automatic differentiation. They therefore avoid the explicit Hessians and higher-order differential operators that appear in direct HJB-based methods. The same martingale-residual construction accommodates both offline learning from complete trajectories and online updating from sequential transitions, providing a common implementation principle for high-dimensional financial problems.

The computational framework is evaluated in two financial settings. High-dimensional basket-option pricing serves as a policy-evaluation benchmark for the martingale Critic, while dynamic mean–variance portfolio selection tests the complete randomized-policy and Actor–Critic framework. In the latter experiment, the offline method achieves a mean annualized return of 16.223% and the online method attains a mean Sharpe ratio of 2.337, across 100 training seeds on the common market path. The corresponding timing comparison and complexity discussion are reported in Section 5.3.

The remainder of the paper is organized as follows. Section 2 introduces the continuous-time stochastic-control and martingale foundations. Section 3 develops the martingale-residual formulation for PE, PG, and AC learning. Section 4 presents the corresponding offline and online schemes, including a one-dimensional test-function illustration. Section 5 contains the financial applications and the mean–variance scalability results. Section 6 concludes. Proofs, algorithms, and supplementary numerical results are collected in the appendix.

2. Continuous-Time Stochastic Control and Martingale Foundations

This section introduces the continuous-time stochastic-control framework used throughout the paper. We first specify the controlled diffusion and the randomized-policy representations used in the entropy-regularized problem. We then specialize to a fixed-policy evaluation, whose PDE and martingale characterization provide the foundation for the local weak formulation in Section 3.

2.1. Controlled Diffusions and General Control Objectives

We first introduce the notation and definitions needed in the subsequent analysis. Following [22], we consider a general stochastic optimal-control problem in continuous time and continuous space. Let d and m be positive integers, let $T > 0$, and let

$$\mu : [0, T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d, \qquad \sigma : [0, T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^{d \times m},$$

where $\mathcal{A}$ is the action space, which may be continuous or discrete. The controlled state process $X_s^a \in \mathbb{R}^d$ follows the stochastic differential equation

$$\mathrm{d}X_s^a = \mu(s, X_s^a, a_s)\mathrm{d}s + \sigma(s, X_s^a, a_s)\mathrm{d}W_s, \quad a_s \in \mathcal{A}, \qquad s \in [0, T]. \tag{1}$$

This equation is defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_s^W\}_{s \geq 0})$, where W_s is a standard m-dimensional Brownian motion and $a_s \in \mathcal{A}$ denotes the action taken by the agent at time s.

In trajectory-based policy evaluation, the agent forms updates from observed states and realized rewards. The agent interacts with the environment, observes the state process generated by a sequence of actions, and collects discounted running rewards and a terminal reward. For an admissible $\{\mathcal{F}_s\}_{s \geq t}$-progressively measurable action process $a = \{a_s, t \leq s \leq T\}$, define

$$\begin{aligned} J(t, x; a) = \mathbb{E}\Bigg[& \int_t^T e^{-\beta(s-t)} f(s, X_s^a, a_s)\, \mathrm{d}s \\ & + e^{-\beta(T-t)} g(X_T^a) \,\Big|\, X_t^a = x \Bigg]. \end{aligned} \tag{2}$$

The associated optimal value is $v^*(t, x) = \sup_a J(t, x; a)$. Here, $f : [0, T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ is the running reward, $g : \mathbb{R}^d \to \mathbb{R}$ is the terminal reward, and $\beta \geq 0$ is the discount factor. Smoothness and growth conditions are imposed below only when they are needed for a PDE or martingale characterization.

2.2. Randomized Policies and Relaxed Dynamics

Let $\pi : (t, x) \mapsto \pi(\cdot|t, x) \in \mathcal{P}(\mathcal{A})$ be a randomized policy. At each time s, the action a_s^π is sampled from $\pi(\cdot|s, X_s^{\pi,a})$. The sampled-action representation of the state process is

$$\begin{aligned} \mathrm{d}X_s^{\pi,a} &= \mu(s, X_s^{\pi,a}, a_s^\pi)\mathrm{d}s + \sigma(s, X_s^{\pi,a}, a_s^\pi)\mathrm{d}W_s, \\ & s \in [t, T], \qquad X_t^{\pi,a} = x. \end{aligned} \tag{3}$$

To balance exploration and exploitation, we add an entropy-regularization term γp to the reward, typically with $p = -\log \pi(a|t, x)$ and temperature $\gamma \geq 0$:

$$\begin{aligned} v(t, x; \pi) = \mathbb{E}\Bigg[& \int_t^T e^{-\beta(s-t)} \Big[f(s, X_s^{\pi,a}, a_s^\pi) \\ & + \gamma p(s, X_s^{\pi,a}, a_s^\pi, \pi(\cdot|s, X_s^{\pi,a})) \Big]\, \mathrm{d}s \\ & + e^{-\beta(T-t)} g(X_T^{\pi,a}) \,\Big|\, X_t^{\pi,a} = x \Bigg]. \end{aligned} \tag{4}$$

This sampled-action representation describes interaction with the controlled environment in (1). For the policy-level PDE and martingale analysis, we use the corresponding relaxed representation. Conditional on the current state, its drift and quadratic covariance are obtained by averaging those of the sampled-action dynamics with respect to the action

distribution. On a possibly enlarged filtered probability space, let $\bar{W}$ be a standard d-dimensional Brownian motion and let the relaxed state process X^π evolve as

$$\mathrm{d}X_s^\pi = \bar{\mu}(s, X_s^\pi, \pi(\cdot|s, X_s^\pi))\mathrm{d}s + \bar{\sigma}(s, X_s^\pi, \pi(\cdot|s, X_s^\pi))\mathrm{d}\bar{W}_s, \tag{5}$$

where

$$\begin{aligned}\bar{\mu}(s,x,\pi) &= \int_{\mathcal{A}} \mu(s,x,a)\pi(a|s,x)\,\mathrm{d}a,\\ \bar{a}(s,x,\pi) &= \int_{\mathcal{A}} \sigma(s,x,a)\sigma(s,x,a)^\top \pi(a|s,x)\,\mathrm{d}a,\\ \bar{\sigma}(s,x,\pi)\bar{\sigma}(s,x,\pi)^\top &= \bar{a}(s,x,\pi).\end{aligned}$$

Thus, $\bar{\sigma}$ is a measurable $d \times d$ square root of the averaged covariance matrix $\bar{a}$. The sampled-action and relaxed processes need not be coupled pathwise; under the standard relaxed-control equivalence [21], they give the same policy-level value. Accordingly, the following is an alternative representation of the value in (4), rather than a second definition:

$$\begin{aligned}v(t,x;\pi) = \mathbb{E}\Bigg[\int_t^T \int_{\mathcal{A}} e^{-\beta(s-t)}\Big[f(s, X_s^\pi, a)\\ + \gamma p(s, X_s^\pi, a, \pi(\cdot|s, X_s^\pi))\Big]\\ \times \pi(a|s, X_s^\pi)\,\mathrm{d}a\,\mathrm{d}s + e^{-\beta(T-t)}g(X_T^\pi)\,\Big|\, X_t^\pi = x\Bigg].\end{aligned} \tag{6}$$

The RL objective is to find

$$v^*(t,x) = \max_{\pi\in\Pi} v(t,x;\pi), \tag{7}$$

where Π denotes the admissible policy class.

2.3. Fixed-Policy Evaluation, PDEs, and Martingale Characterization

We now specialize the general controlled dynamics to a prescribed Markov policy α. Writing $a_s = \alpha(s, X_s)$, let $\mu^\alpha(t,x) = \mu(t,x,\alpha(t,x))$ and $\sigma^\alpha(t,x) = \sigma(t,x,\alpha(t,x))$. For policy evaluation we suppress the entropy term and, to simplify notation, write the induced coefficients again as μ and σ. For the fixed-policy PE formulation in this subsection and its undiscounted martingale constructions below, we set $\beta = 0$; when discounting is present, the corresponding residual includes the term $-\beta v$. The state process then satisfies

$$\mathrm{d}X_s = \mu(s, X_s)\mathrm{d}s + \sigma(s, X_s)\mathrm{d}W_s, \tag{8}$$

and the corresponding value function is

$$v(t,x) = \mathbb{E}\Big[\int_t^T f(s, X_s)\mathrm{d}s + g(X_T)\,\Big|\, X_t = x\Big]. \tag{9}$$

This is the fixed-policy evaluation problem. The value function (9), with terminal condition $v(T,x) = g(x)$, satisfies by the Feynman–Kac formula [24] the partial differential equation

$$\mathcal{L}v(t,x) + f(t,x) = 0, \quad (t,x) \in [0,T) \times \mathbb{R}^d, \tag{10}$$

where the state x evolves according to the SDE (8). The infinitesimal generator is

$$\begin{aligned}\mathcal{L}v(t,x) &= \partial_t v(t,x) + \mu(t,x)\cdot\nabla_x v(t,x)\\ &\quad + \frac{1}{2}\operatorname{Tr}\Big(\sigma(t,x)\sigma(t,x)^\top \operatorname{Hess}_x v(t,x)\Big).\end{aligned} \tag{11}$$

Here, ∂_t denotes the time derivative, ∇_x the spatial gradient, Hess_x the Hessian matrix, and Tr the trace operator. This notation comes from stochastic control theory [4] and provide the basis for the martingale construction below.

For a candidate value function v, define the whole-horizon compensated value process and its increment by

$$M_t^v := v(t,X_t) + \int_0^t f(r,X_r)\mathrm{d}r, \qquad t\in[0,T], \tag{12}$$

$$\begin{aligned}\Delta M_{s,t}^v &:= M_t^v - M_s^v\\ &= v(t,X_t) - v(s,X_s) + \int_s^t f(r,X_r)\mathrm{d}r, \qquad 0\le s\le t\le T.\end{aligned} \tag{13}$$

The following result characterizes the value function $v(t,x)$ by a martingale property.

Theorem 1. *Assume the regularity and integrability conditions for the Feynman–Kac representation and uniqueness of the fixed-policy value function. A candidate u in the stated function class equals the value function in (9) if and only if $u(T,x) = g(x)$ for every x and, for every admissible initial state and time, the compensated process M^u in (12) is a square-integrable martingale. Equivalently, every increment $\Delta M_{s,t}^u$ has conditional expectation zero given $\mathcal{F}_s$. The terminal condition is necessary because adding a constant leaves the martingale property unchanged.*

With the specified terminal payoff, the martingale property yields the Feynman–Kac representation and identifies the fixed-policy value function. The short-interval increments in (13) provide the basic objects for the weak-martingale construction below. We next recall the quadratic variation process associated with M^v.

Definition 1. *When v is the value function, M^v is a continuous square-integrable martingale. Its quadratic variation is*

$$[M^v]_t = \lim_{\|\Delta\|\to 0}\sum_{i=1}^{K}\left(M_{\tau_i}^v - M_{\tau_{i-1}}^v\right)^2 < \infty,$$

where $\Delta : 0 = \tau_0 < \cdots < \tau_K = t$ is an arbitrary partition of $[0,t]$, and $\|\Delta\| = \max_{1\le i\le K}(\tau_i - \tau_{i-1})$ is its mesh size. According to (12),

$$\langle M^v\rangle_t = [M^v]_t = \int_0^t |Z_s|^2\mathrm{d}s, \quad t\in[0,T],$$

where $Z_s = \sigma(s,X_s)^\top\nabla_x v(s,X_s)\in\mathbb{R}^m$ is the diffusion coefficient of the martingale and captures the impact of stochastic fluctuations.

The predictable quadratic variation $\langle M^v\rangle_t$ describes the volatility of the martingale and is related to the smoothness of the value function. In a diffusion setting, it also explains why one works with martingale moment conditions rather than a pathwise squared TD increment [11]; it will be used implicitly in the residual construction below.

3. A Unified Martingale-Residual Framework

Building on the martingale characterizations in Section 2, we develop a common residual formulation for policy evaluation, policy gradient, and Actor–Critic learning. We

first formulate the local weak-martingale condition and its adversarial implementation, then extend the residual to randomized policies and derive the associated policy-gradient and Actor–Critic formulations.

3.1. Martingale-Adversarial Policy Evaluation Under Fixed Policies

We formulate the short-interval martingale-increment condition on intervals $[s, s+h]$, following [25]. The local formulation permits independent short-path simulations and recovers the whole-horizon martingale characterization as $h \to 0$. Under a fixed policy, a pilot process provides starting points for these local simulations.

We introduce a pilot process $\hat{X} : [0,T] \times \Omega \to \mathbb{R}^d$ to explore the state space $\mathbb{R}^d$:

$$\hat{X}_t = \hat{X}_0 + \int_0^t \hat{\mu}(s, \hat{X}_s)\mathrm{d}s + \int_0^t \hat{\sigma}(s, \hat{X}_s)\mathrm{d}W_s, \tag{14}$$

where typically $\hat{\mu}(t,x) := \mu(t,x)$ and $\hat{\sigma}(t,x) := \sigma(t,x)$. The pilot process can be generated in advance and used to provide initial points for subsequent local simulations. For any starting time s, define the system process X_t^s, starting from $\hat{X}_s$ and evolving on $[s,T]$, by

$$X_t^s = \hat{X}_s + \int_s^t \mu(r, X_r^s)\mathrm{d}r + \int_s^t \sigma(r, X_r^s)\mathrm{d}W_r. \tag{15}$$

For the system process $\{X_r^s : r \in [s,T]\}$, initialized at $X_s^s = \hat{X}_s$, the corresponding martingale increment is

$$\Delta M_{s,t}^v = v(t, X_t^s) - v(s, \hat{X}_s) + \int_s^t f(r, X_r^s)\mathrm{d}r, \qquad t \in [s,T]. \tag{16}$$

Theorem 2. *The system process $\{X_t^s : t \in [s,T]\}$ is a Markov process relative to the filtration $\mathcal{F}$ [26]. Moreover, the Itô integral in (13) is an $\mathcal{F}$-martingale on $[s,T]$ [27]. Assume that $R(\cdot,\cdot;v)$ belongs to the domain of the time–space generator $\mathcal{L}$ and that $\mathcal{L}R$ satisfies the local boundedness and growth conditions required by Dynkin's formula. For $t = s+h$, taking the conditional expectation $\mathbb{E}[\cdot \mid \hat{X}_s]$ of the short-interval increment $\Delta M_{s,s+h}^v$ gives*

$$\begin{aligned}\mathbb{E}[\Delta M_{s,s+h}^v \mid \hat{X}_s] &= \int_s^t \mathbb{E}[R(r, X_r^s; v) \mid \hat{X}_s]\mathrm{d}r \\ &= hR(s, \hat{X}_s; v) + O(h^2),\end{aligned} \tag{17}$$

where the residual is $R(t,x;v) := \mathcal{L}v(t,x) + f(t,x)$.

Equation (17) shows that the conditional expectation of the short-interval increment is approximately $hR + O(h^2)$. The local conditional moment condition is

$$\mathbb{E}[\Delta M_{t,t+h}^v \mid \hat{X}_t] = 0. \tag{18}$$

It expresses Bellman consistency along the short path $[t, t+h]$ and follows from the whole-horizon martingale property. To avoid evaluating pointwise conditional expectations, we formulate weak moments. Let $\mathcal{T}_b$ be a class of smooth test functions $\rho : [0,T] \times \mathbb{R}^d \to \mathbb{R}^r$ satisfying $\|\rho\|_\infty \le C_\rho$ for a fixed constant C_ρ. Multiplying (18) by $\rho(t, \hat{X}_t)$ and integrating over time gives, for every $\rho \in \mathcal{T}_b$,

$$\int_0^{T-h} \mathbb{E}\big[\rho(t, \hat{X}_t)\mathbb{E}[\Delta M_{t,t+h}^v \mid \hat{X}_t]\big]\mathrm{d}t = 0. \tag{19}$$

The upper limit $T-h$ keeps each local interval within the terminal horizon. Conversely, if $\mathcal{T}_b$ separates conditional residuals under the joint time–pilot-state distribution, the family of weak conditions (19) implies (18) almost everywhere under that distribution.

Since $\rho(t, \hat{X}_t)$ is measurable with respect to $\hat{X}_t$, the tower property shows that, for each $\rho \in \mathcal{T}_b$, (19) is equivalent to

$$\int_0^{T-h} \mathbb{E}\big[\rho(t, \hat{X}_t)\Delta M^v_{t,t+h}\big]\,\mathrm{d}t = 0. \tag{20}$$

This form replaces pointwise conditional expectations with moments that can be approximated from samples. The corresponding population martingale-adversarial formulation is

$$\min_{u\in\mathcal{V},\, u(T,\cdot)=g} \sup_{\rho\in\mathcal{T}_b} \|G(u,\rho)\|^2 = 0, \tag{21}$$

where

$$G(v,\rho) := \int_0^{T-h} \mathbb{E}\big[\rho(t, \hat{X}_t)\Delta M^v_{t,t+h}\big]\,\mathrm{d}t. \tag{22}$$

This population identity uses exact increments and assumes that the true value function belongs to $\mathcal{V}$. The bounded-test identification is quantified in (A23). The neural architecture selects the indexed moment family for computation, while the anchor in (45) supplies direct regression control.

In computation, we approximate the exact increment $\Delta M^{v^\theta}_{t,t+h}$ by the following one-step Euler residual. With v^θ denoting the value-network approximation of v,

$$\begin{aligned}\widehat{\Delta M}^\theta_{t,h}(\hat{X}_t,\varepsilon) := v^\theta\Big(t+h, \hat{X}_t + \mu(t,\hat{X}_t)h + \sigma(t,\hat{X}_t)\sqrt{h}\,\varepsilon\Big)\\ - v^\theta(t,\hat{X}_t) + hf(t,\hat{X}_t).\end{aligned} \tag{23}$$

Here, $\varepsilon \sim N(0, I_m)$ and $\hat{X}_t + \mu(t,\hat{X}_t)h + \sigma(t,\hat{X}_t)\sqrt{h}\,\varepsilon$ is the Euler approximation of the local state at time $t+h$. The hat distinguishes this Euler residual from the exact continuous-time increment. At nonzero h, the Euler weak residual includes the propagation defect quantified by the local consistency expansion. The computational moment is

$$G_h(v^\theta,\rho) := \int_0^{T-h} \mathbb{E}\big[\rho(t,\hat{X}_t)\widehat{\Delta M}^\theta_{t,h}(\hat{X}_t,\varepsilon)\big]\,\mathrm{d}t. \tag{24}$$

In high-dimensional environments, this quantity is computed by forward propagation, and neural-network automatic differentiation avoids explicit high-order derivative calculations. The adversarial optimization learns the test function, while the expectation in (24) is approximated by samples. The detailed adversarial-learning configuration and pseudocode are reported in Appendix B.1, Algorithm A1.

3.2. Martingale-Adversarial Policy Evaluation Under Randomized Policies

The fixed-policy residual in the preceding subsection provides the local prototype. We now pass to a randomized policy π, replacing the fixed coefficients and running reward by their relaxed, entropy-regularized counterparts. The associated value function $v(\cdot,\cdot;\pi)$ is characterized by a PDE based on the Feynman–Kac formula; see [24].

Lemma 1. *Assume that there exists a unique viscosity solution $v \in C([0,T]\times\mathbb{R}^d)$ satisfying, for $(t,x)\in[0,T)\times\mathbb{R}^d$,*

$$\begin{aligned} 0 = \int_{\mathcal{A}} \Big[& \mathcal{L}^a v(t,x) + f(t,x,a) \\ & + \gamma p(t,x,a,\pi(\cdot|t,x)) - \beta v(t,x) \Big] \pi(a|t,x)\,\mathrm{d}a, \end{aligned} \tag{25}$$

with terminal condition $v(T,x)=g(x)$, $x\in\mathbb{R}^d$. Suppose further that $|v(t,x)|\le C(1+|x|^\mu)$ for some constants $C>0$ and $\mu\ge 1$. Then, v is the value function of the randomized policy, namely, $v(t,x)=v(t,x;\pi)$ for all $(t,x)\in[0,T)\times\mathbb{R}^d$. Here, $\mathcal{L}^a$ is the analogue of the generator $\mathcal{L}$ in (11), with the action variable a included.

To avoid unnecessary technical details, we assume below that $v\in C^{1,2}([0,T)\times\mathbb{R}^d)\cap C([0,T]\times\mathbb{R}^d)$. Conditions ensuring uniqueness, existence, and regularity of viscosity solutions to equations of the form (25) have been studied extensively; see, for example, [22,28], where deterministic-policy settings are extended to randomized policies.

Theorem 3. *The function $v(\cdot,\cdot;\pi)$ is the value function under the policy π if and only if it satisfies the terminal condition $v(T,x;\pi)=g(x)$ and, for any initial pair $(t,x)\in[0,T)\times\mathbb{R}^d$, the process*

$$\begin{aligned} & e^{-\beta s} v(s, X_s^\pi;\pi) \\ & \quad + \int_t^s e^{-\beta r} \int_{\mathcal{A}} \Big[f(r, X_r^\pi, a) \\ & \qquad + \gamma p(r, X_r^\pi, a, \pi(\cdot|r, X_r^\pi)) \Big] \\ & \qquad \times \pi(a|r, X_r^\pi)\,\mathrm{d}a\,\mathrm{d}r \end{aligned} \tag{26}$$

is an $(\mathcal{F}^{X^\pi},\mathbb{P})$-martingale on $[t,T]$. Moreover, this is equivalent to the martingale-orthogonality condition

$$0 = \mathbb{E}\left[\int_0^T \xi_t \Big(\mathrm{d}v(t, X_t^\pi;\pi) + \widetilde{q}(t, X_t^\pi, \pi)\,\mathrm{d}t - \beta v(t, X_t^\pi;\pi)\,\mathrm{d}t \Big) \right], \tag{27}$$

for every scalar adapted process $\xi\in L^2_{\mathcal{F}^{X^\pi}}([0,T];\mathbb{R})$.

In reinforcement learning, function approximation is usually used to learn the objects of interest. For randomized-policy evaluation, we use a parametric family $v^\theta = v^\theta(\cdot,\cdot;\pi)$ defined on $[0,T]\times\mathbb{R}^d$ to approximate v, where $\theta\in\Theta\subseteq\mathbb{R}^{L_\theta}$. The problem is to find the best parameter θ in a suitable sense. We impose the following regularity assumption on the approximators. When no confusion is possible, we suppress the dependence of $v^\theta(\cdot,\cdot;\pi)$ on π.

Assumption 1. *For all $\theta\in\Theta$, $v^\theta\in C^{1,2}([0,T)\times\mathbb{R}^d)\cap C([0,T]\times\mathbb{R}^d)$ and satisfies a polynomial growth condition in x. Moreover, $v^\theta(t,x)$ is smooth in θ, and $\partial v^\theta/\partial\theta$ and $\partial^2 v^\theta/\partial\theta^2$ also belong to $C^{1,2}([0,T)\times\mathbb{R}^d)\cap C([0,T]\times\mathbb{R}^d)$ and satisfy polynomial growth conditions in x.*

The martingale representation above also underlies the established martingale-loss and fixed-test-function TD schemes described in Section 4. For the randomized-policy Critic used in the proposed Actor–Critic construction, we use the following weak-martingale adversarial formulation:

$$\min_{\theta\in\Theta}\sup_{\rho\in\mathcal{T}_b}\|G_h(v^\theta,\rho;\pi)\|^2, \tag{28}$$

where

$$G_h(v^\theta,\rho;\pi) := \int_0^{T-h}\mathbb{E}\big[\rho(t,X_t^\pi)\mathcal{R}_h^\theta(t,X_t^\pi,\varepsilon;\pi)\big]\,\mathrm{d}t, \tag{29}$$

and, for a generic state $x\in\mathbb{R}^d$,

$$\begin{aligned}\mathcal{R}_h^\theta(t,x,\varepsilon;\pi) := v^\theta\big(t+h, x+h\bar{\mu}(t,x,\pi)\\ +\bar{\sigma}(t,x,\pi)\sqrt{h}\,\varepsilon\big) - v^\theta(t,x)\\ +h\widetilde{q}(t,x,\pi)-\beta h v^\theta(t,x),\end{aligned} \tag{30}$$

with $\varepsilon\sim N(0,I_d)$ and

$$\begin{aligned}\widetilde{q}(t,x,\pi) := \int_{\mathcal{A}}\Big[f(t,x,a)+\gamma p(t,x,a,\pi(\cdot|t,x))\Big]\\ \times\pi(a|t,x)\,\mathrm{d}a.\end{aligned} \tag{31}$$

At a fixed time step, the population objective may remain positive because of Euler discretization, function approximation, or the chosen test class. Its empirical counterpart is also affected by sampling and optimization error. This formulation uses forward propagation and a learned test function. In randomized-policy residuals, $\varepsilon\sim N(0,I_d)$ is compatible with the $d\times d$ relaxed diffusion factor $\bar{\sigma}$. Its offline and online implementations are specified in Section 4.

3.3. Martingale-Residual Policy Gradient

Building on the randomized-policy residual above, we first recall the randomized-policy gradient identities of [22]. Given an admissible parametric policy family π^ϕ with parameter $\phi\in\Phi\subset\mathbb{R}^{L_\phi}$, suppose that a policy-evaluation step has been performed and an estimate of the value function $v(t,x;\pi^\phi)$ is available. Following [22], X^{π^ϕ} in the sampled-action identities below denotes $X^{\pi^\phi,a}$ in the notation of Section 2.2. The policy-gradient (PG) step aims to compute

$$\mathfrak{g}(t,x;\phi) := \frac{\partial}{\partial\phi}v(t,x;\pi^\phi)\in\mathbb{R}^{L_\phi}, \tag{32}$$

which is then used to update ϕ at the current time–state pair (t,x) in order to maximize the value function [22].

Theorem 4 ([22]). *For a parametric policy π^ϕ, the policy gradient $\mathfrak{g}(t,x;\phi)$ admits the representation*

$$\begin{aligned}\mathfrak{g}(t,x;\phi) = \mathbb{E}\Bigg[\int_t^T e^{-\beta(s-t)}\Bigg\{\frac{\partial}{\partial\phi}\log\pi^\phi(a_s^{\pi^\phi}|s,X_s^{\pi^\phi})\\ \times\Big(\mathrm{d}v(s,X_s^{\pi^\phi};\pi^\phi)+\big[f(s,X_s^{\pi^\phi},a_s^{\pi^\phi})\\ +\gamma p(s,X_s^{\pi^\phi},a_s^{\pi^\phi},\pi^\phi(\cdot|s,X_s^{\pi^\phi}))\\ -\beta v(s,X_s^{\pi^\phi};\pi^\phi)\big]\,\mathrm{d}s\Big)\\ +\gamma q_\phi(s,X_s^{\pi^\phi},a_s^{\pi^\phi};\phi)\,\mathrm{d}s\Bigg\}\,\Big|\,X_t^{\pi^\phi}=x\Bigg],\end{aligned} \tag{33}$$

where $q_\phi(t,x,a;\phi) = \partial_\phi p(t,x,a,\pi^\phi(\cdot|t,x))$.

The proof is given in Appendix A.3. All terms in (33) can be computed from future sample trajectories and the estimated value function v. Since (33) involves future trajectories starting from t, it is naturally suited to offline implementation.

Online learning extension.

Theorem 5 ([22]). *Suppose that* ϕ^* *is an interior optimal parameter and that the first-order condition* $\mathfrak{g}(t,x;\phi^*) = 0$ *holds for every* $(t,x) \in [0,T] \times \mathbb{R}^d$. *Then,*

$$\begin{aligned} 0 = \mathbb{E}\Bigg[\int_0^T \kappa_s \Bigg\{ \Bigg[\frac{\partial}{\partial \phi} \log \pi^{\phi^*}(a_s^{\pi^{\phi^*}} | s, X_s^{\pi^{\phi^*}}) + \zeta_s \Bigg] \\ \times \Big(\mathrm{d}v(s, X_s^{\pi^{\phi^*}}; \pi^{\phi^*}) + \big[f(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}) \\ + \gamma p(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}, \pi^{\phi^*}(\cdot|s, X_s^{\pi^{\phi^*}})) \\ - \beta v(s, X_s^{\pi^{\phi^*}}; \pi^{\phi^*}) \big] \mathrm{d}s \Big) \\ + \gamma q_\phi(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}; \phi^*) \, \mathrm{d}s \Bigg\} \Bigg| \, X_0^{\pi^{\phi^*}} = x \Bigg]. \end{aligned} \tag{34}$$

for any $\kappa \in L^2_{\mathcal{F}^{X^{\pi^{\phi^*}}}}([0,T];\mathbb{R})$ *and* $\zeta \in L^2_{\mathcal{F}^{X^{\pi^{\phi^*}}}}([0,T];\mathbb{R}^{L_\phi})$.

The proof is given in Appendix A.4. The first-order policy conditions can therefore be written as a system of moment equations. If $\kappa_s = e^{-\beta s}$, then (34) agrees with $\mathfrak{g}(0,x;\phi^*)$; if κ_s is set to zero on future intervals $[t,T]$, then the condition depends only on past observations and supports online learning.

We now construct the proposed martingale-residual gradient estimator from the same randomized-policy residual. The implementation uses neural-network parameterizations of the value function and the randomized policy. Together with the learnable-test-function construction in Section 3.2, it is based on sampled residuals and first-order automatic differentiation rather than an explicit expansion of higher-order generator terms.

Recall from Lemma 1 that, for a given policy, the value function $v(t,x;\pi)$ satisfies (25). In policy-gradient learning, the objective is to maximize the expected cumulative reward by searching over admissible policies:

$$\begin{aligned} v^*(t,x) = \sup_\pi \mathbb{E}\Bigg[\int_t^T e^{-\beta(s-t)} \widetilde{q}(s, X_s^\pi, \pi) \, \mathrm{d}s \\ + e^{-\beta(T-t)} g(X_T^\pi) \, \Big| \, X_t^\pi = x \Bigg], \end{aligned} \tag{35}$$

where

$$\begin{aligned} \widetilde{q}(s, X_s^\pi, \pi) = \int_{\mathcal{A}} \Big[f(s, X_s^\pi, a) + \gamma p(s, X_s^\pi, a, \pi(\cdot|s, X_s^\pi)) \Big] \\ \times \pi(a|s, X_s^\pi) \, \mathrm{d}a \end{aligned}$$

is the entropy-regularized averaged reward. This control problem is equivalent to the HJB equation

$$\begin{aligned} 0 = {} & \partial_t v^*(t,x) - \beta v^*(t,x) \\ & + \sup_{\pi} \int_{\mathcal{A}} \Big[\mu(t,x,a) \cdot \nabla_x v^*(t,x) \\ & \quad + \frac{1}{2} \operatorname{Tr}\Big(\sigma(t,x,a)\sigma(t,x,a)^\top \nabla_x^2 v^*(t,x) \Big) \\ & \quad + f(t,x,a) + \gamma p(t,x,a,\pi(\cdot|t,x)) \Big] \pi(a|t,x)\, \mathrm{d}a, \end{aligned} \tag{36}$$

with terminal condition $v^*(T,x) = g(x)$.

For a fixed policy π, the HJB equation reduces to a linear PDE, which is represented by the randomized-policy martingale residual in Section 3.2. Using this residual, we approximate the value function v by v^θ and the policy π by π^ϕ, and consider the adversarial loss

$$\min_{\theta \in \Theta} \sup_{\rho \in \mathcal{T}_b} \| G_h(v^\theta, \rho; \pi^\phi) \|^2. \tag{37}$$

Here,

$$G_h(v^\theta, \rho; \pi^\phi) := \int_0^{T-h} \mathbb{E}\big[\rho(t, \hat{X}_t^{\pi^\phi}) \mathcal{R}_h^\theta(t, \hat{X}_t^{\pi^\phi}, \varepsilon; \pi^\phi) \big]\, \mathrm{d}t, \tag{38}$$

where $\mathcal{R}_h^\theta$ is the randomized-policy Euler residual defined in (30).

Inspired by [25], we take the conditional expectation of the randomized-policy residual in (30). The following theorem shows the essential connection between the martingale-residual gradient and the existing randomized policy-gradient identities recalled above.

Theorem 6. *Let ϕ be the current policy-network parameter. Assume that the Critic is exact at the current policy, that is, $v^\theta(t,x) = v(t,x;\pi^\phi)$. Let the pilot process $\hat{X}^{\pi^\phi}$ be sampled from the relaxed dynamics of π^ϕ, with $\hat{X}_0^{\pi^\phi} = x_0$. For a candidate policy parameter ψ, define*

$$L_h(\psi \mid \phi) = \mathbb{E}\left[\int_0^{T-h} e^{-\beta t} \mathcal{R}_h^\theta(t, \hat{X}_t^{\pi^\phi}, \varepsilon; \pi^\psi)\, \mathrm{d}t \right], \tag{39}$$

where the current-policy Critic and the law of $\hat{X}^{\pi^\phi}$ are held fixed when differentiating with respect to ψ, $\mathcal{R}_h^\theta$ is defined in (30), and $\varepsilon \sim N(0, I_d)$. Suppose the policy-dependent drift and a chosen diffusion factor are differentiable, with integrable envelopes permitting differentiation, expectation, and Gaussian integration by parts. Assume the derivative-level Euler bounds (A34) with coefficient-weighted integrable envelopes uniform in h, and the terminal-kernel bound stated in Appendix A.7. Then,

$$\frac{1}{h} \nabla_\psi L_h(\psi \mid \phi)\Big|_{\psi=\phi} = \mathfrak{g}(0, x_0; \phi) + O(h), \tag{40}$$

where $\mathfrak{g}(0, x_0; \phi)$ is the classical continuous-time randomized policy gradient defined by the existing identities recalled above. Consequently,

$$\phi \leftarrow \phi + \alpha_\phi \frac{1}{h} \nabla_\psi L_h(\psi \mid \phi)\Big|_{\psi=\phi} \approx \phi + \alpha_\phi \big(\mathfrak{g}(0, x_0; \phi) + O(h) \big), \tag{41}$$

so the residual derivative approximates the current-policy value gradient with order-one consistency. Proposition 1 extends the estimate to a learned Critic, and Corollary 1 quantifies the resulting Actor updates.

Under its stated assumptions, Theorem 6 shows that the rescaled current-policy derivative of $L_h(\psi \mid \phi)$ approaches the classical policy gradient as $h \to 0$. It can be computed by backpropagation through sampled residuals, without explicitly expanding model-dependent higher-order generator terms such as $\mathcal{L}^a$. This motivates the Hessian-free implementation used below. Section 3.5 quantifies Critic approximation and sampling errors and connects them to policy improvement.

3.4. Martingale-Residual Actor–Critic Formulation

Combining the randomized-policy adversarial Critic in Section 3.2 with the martingale-residual Actor update in Section 3.3 yields the Actor–Critic formulation. Both updates are computed from sampled residuals by first-order automatic differentiation.

The algorithm alternates between optimizing the Critic and optimizing the Actor. The value function is parameterized by a network $v^\theta(t,x)$, the policy distribution by a network $\pi^\phi(\cdot \mid t,x)$, and the test function by a network $\rho_\eta(t,x)$. Under the stated weak-martingale conditions, the Critic targets martingale consistency of the value approximation. Using the randomized-policy adversarial formulation in Section 3.2, the value evaluation problem is represented as a martingale-residual minimization problem:

$$\min_\theta \sup_\eta \Big\{ a\mathbb{E}\big[\widehat{G}_h^{(1)}(\theta,\phi,\eta)^\top \widehat{G}_h^{(2)}(\theta,\phi,\eta)\big] + \lambda A(\theta) \Big\}, \tag{42}$$

where $\widehat{G}_h^{(1)}$ and $\widehat{G}_h^{(2)}$ are empirical versions of $G_h(v^\theta,\rho_\eta;\pi^\phi)$. The positive anchor A and its weights are specified in (45). For parameters fixed before sampling, independent identically distributed fresh batches give $\mathbb{E}[(\widehat{G}_h^{(1)})^\top \widehat{G}_h^{(2)}] = \|G_h(v^\theta,\rho_\eta;\pi^\phi)\|^2$. Appendix A.8 gives the variance identity and the conditional empirical objectives used by the two applications. The Critic alternates minimization in θ and maximization in η. Theorem 7 establishes local stability of the anchored population recursion, and Proposition 1 quantifies the effect of the learned Critic on the residual Actor.

The Actor step is based on the policy-gradient construction in Section 3.3. It estimates the gradient through the derivative of the martingale residual with respect to the policy parameter:

$$\widehat{\mathfrak{g}}_\phi \approx \frac{1}{h}\int_0^{T-h} e^{-\beta t}\mathbb{E}\Big[\nabla_\psi \mathcal{R}_h^\theta(\cdot;\pi^\psi)\Big|_{\psi=\phi}\Big]\mathrm{d}t, \tag{43}$$

and then updates $\phi \leftarrow \phi + \alpha_\phi \widehat{\mathfrak{g}}_\phi$. This update uses automatic differentiation of the residual, avoiding an explicit expansion of Itô's formula or the computation of high-dimensional Hessian matrices. The entropy-regularization term is included in the residual. The pathwise derivative uses the policy-dependent relaxed coefficients and sampled Euler noise, or a differentiable observed transition such as the portfolio allocation-to-wealth map in (A39).

Overall, the algorithm adopts an alternating optimization scheme: in each cycle, several adversarial Critic iterations are first performed to stabilize value evaluation, followed by an Actor update based on the latest Critic. The offline and online implementations of this scheme are specified in Section 4. The pseudocode is given in Appendix B.2, Algorithm A2.

In high-dimensional settings, the implementation avoids explicit Hessians and uses automatic differentiation of sampled residuals. The corresponding operation counts for the mean–variance application are discussed in Section 5.

3.5. Accuracy and Stability of the Learned Residual Updates

We now quantify how adversarial value evaluation and Critic accuracy enter policy improvement. Throughout the fixed-policy analysis, the policy, its pilot distribution and the mean–variance multiplier are held fixed. Let Z denote the conditioning time and state,

Z' its sampled successor, and w the represented Critic functions. Write the normalized residual as $r_h[w] = \mathcal{B}_h w + c_h$, where $\mathcal{B}_h$ is linear in w. With ρ_η a bounded vector test, define

$$m_h(w,\eta) = \mathbb{E}[\rho_\eta(Z) r_h[w](Z,Z')], \qquad D_h(w) = \sup_{\eta \in K} \|m_h(w,\eta)\|^2, \tag{44}$$

$$\Psi_h(w) = aD_h(w) + \lambda A(w), \qquad A(w) = \mathbb{E}\|w(Z) - Y\|_W^2, \qquad a, \lambda > 0. \tag{45}$$

Here, W is positive definite and Y is a trajectory target. Time and test-output normalizations are included in a or m_h. The weak term learns martingale consistency, while the trajectory anchor controls regression accuracy. If $q(Z) = \mathbb{E}[Y \mid Z]$, conditional orthogonality gives

$$A(w) = A(q) + \|w - q\|_{L^2_W}^2. \tag{46}$$

The bounded-test identification and the resulting approximation bound are established in Appendix A.6. In particular, when $D_h(q) = 0$, the population objective controls the represented-function error by $\Psi_h(w) - \Psi_h(q) \geq \lambda \|w - q\|_{L^2_W}^2$.

For basket-option evaluation, $w(t,S) = e^{-rt} v(t,S)$ and $Y = e^{-rT} g(S_T)$; the implemented weights are $a = 0.1$ and $\lambda = 0.3$. For portfolio evaluation, $w = (m_1, m_2, m_E)$ represents the first two terminal gross-return moments and the future entropy reward, with $W = \mathrm{diag}(1, 1/4, 1)$ and $\lambda = 0.25$. The weak cross-moment is averaged over its 14 outputs, with coefficient one. For fixed multiplier ℓ and target z, the corresponding value is affine in these functions:

$$v_w(t,x) = -x^2 m_2 + 2\ell x m_1 - \ell^2 + (\ell - z)^2 + m_E. \tag{47}$$

The following theorem describes how the anchor stabilizes alternating Critic/Test updates near a compatible value representation.

Theorem 7 (Local stability of anchored adversarial evaluation)**.** *Set* $F(\theta,\eta) = a\|m_h(w_\theta,\eta)\|^2 + \lambda A(w_\theta)$. *Let* K *be a compact convex test-parameter neighborhood with nonempty interior. Suppose that, on a neighborhood of* $\{\theta_*\} \times K$, *the moment map and anchor are twice-continuously differentiable with uniformly bounded derivatives, and*

$$m_h(w_{\theta_*},\eta) = 0 \quad (\eta \in K), \qquad \nabla A(\theta_*) = 0, \qquad \nabla^2 A(\theta_*) \succeq a_0 I, \quad a_0 > 0. \tag{48}$$

The last condition is imposed in locally identifiable Critic coordinates. There exist a Critic neighborhood and constants $0 < \mu \leq L$ *such that the alternating population updates*

$$\begin{aligned} \theta_{k+1} &= \theta_k - \alpha \nabla_\theta F(\theta_k, \eta_k), & 0 < \alpha \leq L^{-1}, \\ \eta_{k+1} &= \eta_k + \tau_k \nabla_\eta F(\theta_{k+1}, \eta_k), & 0 \leq \tau_k \leq \bar{\tau}, \end{aligned} \tag{49}$$

started sufficiently close to (θ_*, η_0), *with* $\eta_0 \in \operatorname{int} K$, *satisfy*

$$\|\theta_k - \theta_*\| \leq \kappa^k \|\theta_0 - \theta_*\|, \qquad \kappa = 1 - \alpha\mu < 1, \qquad \eta_k \longrightarrow \eta_\infty \in K. \tag{50}$$

The limit is a local saddle point. Moreover, $\|\nabla_\eta F(\theta,\eta)\| \leq C\|\theta - \theta_*\|^2$ *in that neighborhood.*

The anchor supplies local Critic curvature, and the Test gradient decreases quadratically as the Critic approaches a compatible value. This gives convergence to the saddle set even though the Test directions are flat at its limit. For a representable regression target, the curvature has the explicit form $\nabla^2 A(\theta_*) = 2\mathbb{E}[J_*^\top W J_*]$, where $J_* = D_\theta w_{\theta_*}$. The proof and the tracking estimate for a changing policy are given in Appendix A.6. The numerical

algorithms specify the adaptive optimizer, batch reuse, and update frequencies used to implement the residual objective.

For the policy-gradient analysis, abbreviate the relaxed coefficients as $b_\phi = \bar{\mu}(\cdot, \pi^\phi)$, $S_\phi = \bar{\sigma}(\cdot, \pi^\phi)$ and $C_\phi = S_\phi S_\phi^\top$. This diffusion covariance is distinct from the trajectory anchor $A(w)$. Let $g_h[w]$ be the rescaled derivative in Theorem 6, evaluated with Critic w, and write $v = v(\cdot, \cdot; \pi^\phi)$, $J(\phi) = v(0, x_0; \pi^\phi)$ and $\widehat{v} = v + e$. The current-policy Critic and pilot law remain fixed under differentiation.

The next result quantifies how errors in the learned Critic's spatial derivatives enter the Actor's residual-gradient estimate.

Proposition 1 (Finite-step Critic error in the residual gradient)**.** *Suppose b_ϕ and a chosen diffusion factor S_ϕ are differentiable in ϕ, and e has two spatial derivatives. Assume integrable envelopes permitting differentiation and Gaussian integration by parts. For $Y_h^\phi = X_t^{\pi^\phi} + h b_\phi(t, X_t^{\pi^\phi}) + \sqrt{h} S_\phi(t, X_t^{\pi^\phi})\varepsilon$, with independent standard Gaussian ε,*

$$g_h[\widehat{v}] - g_h[v] = \int_0^{T-h} e^{-\beta t} \mathbb{E}\big[(D_\phi b_\phi)^\top \nabla_x e(t+h, Y_h^\phi) + \tfrac{1}{2}(D_\phi C_\phi)^* \nabla_x^2 e(t+h, Y_h^\phi)\big] \mathrm{d}t. \tag{51}$$

Here, $[(D_\phi C)^* H]_j = (\partial_{\phi_j} C) : H$. With respect to $\mathrm{d}\mu_h = e^{-\beta t}\mathrm{d}t\, \mathrm{d}\mathbb{P}(X_t^{\pi^\phi}, \varepsilon)$, set $B_{1,h} = \|D_\phi b_\phi\|_{L^2(\mu_h;\mathrm{op})}$, $B_{2,h} = \|D_\phi C_\phi\|_{L^2(\mu_h;\mathrm{op})}$, and $E_{j,h} = \|D_x^j e(t+h, Y_h^\phi)\|_{L^2(\mu_h)}$. The covariance operator maps Euclidean policy directions to Frobenius matrices. Under the consistency conditions of Theorem 6,

$$\|g_h[\widehat{v}] - \nabla_\phi J(\phi)\| \le C_E h + B_{1,h} E_{1,h} + \tfrac{1}{2} B_{2,h} E_{2,h}. \tag{52}$$

Equation (52) separates propagation accuracy and the two Critic-derivative errors relevant to the Actor. The Hessian expresses diffusion sensitivity in the proof; the sampled update is evaluated by first-order differentiation through Y_h^ϕ. Appendix A.7 connects these derivative errors to local Critic convergence, and gives the sampling and observed-transition counterparts.

The following bound separates the effects of a finite update budget, gradient bias, and sampling variability on Actor stationarity.

Corollary 1 (Finite-iteration Actor stationarity)**.** *Let $J(\phi) = v(0, x_0; \pi^\phi)$ have an L_J-Lipschitz gradient on a region containing the update segments, and let $J(\phi_k) \le J_{\max}$. For $\phi_{k+1} = \phi_k + \alpha_\phi \widehat{g}_k$, $0 < \alpha_\phi \le L_J^{-1}$, assume*

$$\mathbb{E}[\widehat{g}_k \mid \mathcal{F}_k] = \nabla J(\phi_k) + b_k, \qquad \|b_k\| \le \epsilon_k, \qquad \mathbb{E}[\|\widehat{g}_k - \mathbb{E}[\widehat{g}_k \mid \mathcal{F}_k]\|^2 \mid \mathcal{F}_k] \le s_k^2.$$

Then,

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla J(\phi_k)\|^2 \le \frac{2(J_{\max} - J(\phi_0))}{\alpha_\phi K} + \frac{1}{K}\sum_{k<K} \mathbb{E}\epsilon_k^2 + \frac{L_J \alpha_\phi}{K} \sum_{k<K} \mathbb{E}s_k^2. \tag{53}$$

For a fresh residual-gradient batch, the bias is bounded by (52), with the pilot-law comparison term specified in Appendix A.7, and $s_k^2 \le \sigma_{h_k}^2 / N_k$.

Thus, Critic accuracy, the propagation step, and the sampling budget enter policy improvement through distinct terms. The finite-iteration term decreases as K^{-1}, while improved Critic derivatives and sampling precision reduce the attainable stationarity level.

4. Offline and Online Learning Schemes

This section describes how the martingale formulations in Section 3 are implemented from complete trajectories and from sequentially observed transitions. In the discretized schemes, the local horizon h is represented by the grid step Δt. We retain the martingale loss and fixed-test-function TD as policy-evaluation constructions under these two information regimes, and then specify the corresponding offline and online martingale-adversarial Actor–Critic updates.

4.1. Offline Learning: Martingale Loss and Adversarial Actor–Critic

With complete sampled trajectories, the martingale loss of [11] provides an offline policy-evaluation method. Its construction is based on the optimal L^2 projection property of martingales.

Lemma 2. *Let $\{X_t, 0 \le t \le T\}$ be the state process, and suppose that the value function is correctly characterized by the square-integrable martingality of $M_t^v = v(t, X_t) + \int_0^t f(s, X_s)\mathrm{d}s$. The martingale condition is equivalent to*

$$M_t^v = \mathbb{E}[M_T^v \mid \mathcal{F}_t], \quad \forall t \in [0, T]. \tag{54}$$

This means that, at any time before the terminal time T, the process is the conditional expectation of its terminal value given all information available at that time. Furthermore, a basic property of conditional expectation implies that M_t^v minimizes the L^2 error between M_T^v and any $\mathcal{F}_t$-measurable random variable, namely,

$$M_t^v \equiv \mathbb{E}[M_T^v \mid \mathcal{F}_t] = \underset{\xi \text{ is } \mathcal{F}_t\text{-measurable}}{\arg\min} \ \mathbb{E}\,|M_T^v - \xi|^2, \quad \forall t \in [0, T]. \tag{55}$$

For a value-network candidate, write $M_t^\theta := v^\theta(t, X_t) + \int_0^t f(s, X_s)\mathrm{d}s$. Based on this lemma, we define the martingale loss function as

$$\begin{aligned}
\mathrm{ML}(\theta) &:= \frac{1}{2}\|M_T^v - M_\cdot^\theta\|_{L^2}^2 = \frac{1}{2}\mathbb{E}\left[\int_0^T |M_T^v - M_t^\theta|^2 \mathrm{d}t\right] \\
&\approx \frac{1}{2}\mathbb{E}\left[\sum_{i=0}^{K-1}\left(g(X_T) - v^\theta(t_i, X_{t_i}) + \int_{t_i}^T f(s, X_s)\mathrm{d}s\right)^2 \Delta t\right] \\
&\approx \frac{1}{2}\mathbb{E}\left[\sum_{i=0}^{K-1}\left(g(X_{t_K}) - v^\theta(t_i, X_{t_i}) + \sum_{j=i}^{K-1} f(t_j, X_{t_j})\Delta t\right)^2 \Delta t\right] \\
&= \frac{1}{2}\mathbb{E}\left[\sum_{i=0}^{K-1}\left(g(X_{t_K}) + \sum_{j=0}^{K-1} f(t_j, X_{t_j})\Delta t \right.\right. \\
&\qquad \left.\left. - v^\theta(t_i, X_{t_i}) - \sum_{j=0}^{i-1} f(t_j, X_{t_j})\Delta t\right)^2 \Delta t\right].
\end{aligned} \tag{56}$$

Here, $0 = t_0 < t_1 < \cdots < t_K = T$ is the time grid. This loss requires only the realized cumulative reward $\sum_{j=0}^{i-1} f(t_j, X_{t_j})\Delta t$ and the terminal reward $g(X_T)$; it does not require explicit evaluation of the drift or diffusion coefficients. Hence, it can be implemented by replacing expectations with sample averages over trajectories.

For a randomized policy π, the corresponding offline objective is

$$\min_{\theta \in \Theta} \ \mathbb{E}\left[\int_0^T \left|\mathcal{R}_t^\theta\right|^2 \mathrm{d}t\right], \tag{57}$$

where

$$\begin{aligned}\mathcal{R}_t^\theta = e^{-\beta T} g(X_T^{\pi,a}) - e^{-\beta t} v^\theta(t, X_t^{\pi,a}) \\ + \int_t^T e^{-\beta s} \Big[f(s, X_s^{\pi,a}, a_s^\pi) \\ + \gamma p(s, X_s^{\pi,a}, a_s^\pi, \pi(\cdot|s, X_s^{\pi,a})) \Big] \mathrm{d}s.\end{aligned} \tag{58}$$

This is the entropy-regularized counterpart of the preceding martingale-loss construction.

The loss function compares the predicted value function with the realized reward over the entire trajectory. This method is a continuous-time analogue of Monte Carlo policy evaluation with function approximation in the classical MDP and RL literature [7]. Its trajectory-based implementation is given in Appendix B.3, Algorithm A3.

Finally, applying stochastic gradient descent to minimize the martingale loss yields the update rule

$$\begin{aligned}\theta \leftarrow \theta + \alpha \sum_{i=0}^{K-1} \left(g(X_{t_K}) + \sum_{j=i}^{K-1} f(t_j, X_{t_j}) \Delta t - v^\theta(t_i, X_{t_i}) \right) \\ \times \frac{\partial v^\theta}{\partial \theta}(t_i, X_{t_i}) \Delta t.\end{aligned} \tag{59}$$

When the state dynamics satisfy Lipschitz continuity and linear growth conditions, ensuring the uniqueness of the SDE solution, and when the parametric approximator v^θ is smooth and continuous in the parameter θ, the following theorem from [11] states that minimizing the martingale loss is equivalent to minimizing the mean-squared value error (MSVE) between the estimated value function v^θ and the true value function v:

$$\mathrm{MSVE}(\theta) = \mathbb{E}\left[\int_0^T |v(t, X_t) - v^\theta(t, X_t)|^2 \mathrm{d}t \right]. \tag{60}$$

Theorem 8 ([11]). *The following identity holds:*

$$\mathrm{ML}(\theta) = \frac{1}{2} \mathrm{MSVE}(\theta) + \frac{1}{2} \mathbb{E}\left[\int_0^T |M_T^v - M_t^v|^2 \mathrm{d}t \right], \tag{61}$$

where the second term is a constant independent of θ. Thus,

$$\arg\min_\theta \mathrm{ML}(\theta) = \arg\min_\theta \mathrm{MSVE}(\theta). \tag{62}$$

Moreover, any convergent subsequence of the minimizer of the discretized martingale loss function $\theta_{ML}^*(\Delta t) \in \arg\min_\theta \mathrm{ML}_{\Delta t}(\theta)$ converges to the minimizer of the martingale loss function, and the convergence rate depends on the Hölder continuity of the reward function.

MSVE is a theoretically reliable loss function, but it cannot be implemented directly because v is unobservable. The martingale loss realizes it through the observable terminal value $M_T^v = g(X_T) + \int_0^T f(s, X_s)\mathrm{d}s$. Recall that MSBE is defined as

$$\mathbb{E} \int_0^T [\mathcal{L} v^\theta(t, X_t) + f(t, X_t)]^2 \mathrm{d}t,$$

which is the mean square of the Bellman residual. Since the true value function v satisfies $\mathcal{L}v + f = 0$, MSBE is a natural population objective under the stated model. However, in a data-driven RL implementation, $\mathcal{L}$ involves environmental quantities such as the drift μ and volatility σ, which need not be evaluated explicitly. Under the assumptions of the

martingale-loss characterization, zero-population MSVE corresponds to the true value function and hence to zero MSBE. The ML construction uses complete trajectories and is retained here as an offline policy-evaluation reference method; it avoids a direct grid-based PDE solution.

For the proposed offline Actor–Critic method, complete trajectories are stored and sampled in disjoint batches. The Critic forms the empirical counterpart of $G_h(v^\theta, \rho)$ in (24) from one-step residuals along the sampled paths and combines it with the trajectory anchor in (45). It alternates a descent step in the value-network parameters with an ascent step in the test-network parameters. Under a randomized policy, these increments are given by the entropy-regularized local residual in Section 3.2. This is the randomized-policy martingale-adversarial Critic; the martingale loss above remains the offline PE reference. The Actor is updated using the martingale-residual gradient derived in Section 3.3. The general procedure is given in Appendix B.2; the mean–variance specialization is given in Algorithm A6.

4.2. Online Learning: Fixed-Test-Function TD and Adversarial Actor–Critic

Unlike the martingale loss, which uses a complete sample path over $[0, T]$, an online update is formed from sequentially observed transitions. In a diffusion setting, a pathwise squared TD objective is affected by the martingale quadratic variation [11]. We first recall the martingale-orthogonality method of [11], which enforces the martingale condition through selected test functions and sample moments.

Theorem 9 ([11]). *If $M_t^v = v(t, X_t) + \int_0^t f(s, X_s)\mathrm{d}s$ is a square-integrable martingale, then for any scalar adapted test process $\xi \in L^2_{\mathcal{F}}([0, T]; \mathbb{R})$,*

$$\mathbb{E}\left[\int_0^T \xi_t \mathrm{d}M_t^v\right] = 0. \tag{63}$$

where

$$\mathrm{d}M_t^v = [\mathcal{L}v(t, X_t) + f(t, X_t)]\mathrm{d}t + \left(\nabla_x v(t, X_t)\right)^\top \sigma(t, X_t)\mathrm{d}W_t.$$

Following [11], (63) is called a family of martingale orthogonality conditions. When (63) holds for every $\xi \in L^2_{\mathcal{F}}([0, T]; \mathbb{R})$, the family of conditions is necessary and sufficient for M_t^v to be a martingale.

This theorem transforms the infinite-dimensional martingale condition into a family of equations parameterized by the test function ξ. When the condition holds for the full admissible test-function class, it is equivalent to the process M^v being a martingale. In numerical implementation, one selects finitely many test functions $\{\xi_t^{(1)}, \xi_t^{(2)}, \ldots, \xi_t^{(L)}\}$ and estimates the associated finite system of moment conditions. The choice of test functions determines the moment restrictions imposed by the approximation. In continuous environments, these constructions give counterparts of traditional TD algorithms in discrete environments. The corresponding pseudocode is given in Appendix B.4, Algorithm A4.

Let $\mathrm{d}M_t^\theta := \mathrm{d}v^\theta(t, X_t) + f(t, X_t)\mathrm{d}t$ denote the compensated-process increment induced by the Critic v^θ; it is not assumed to be a martingale away from the correct value function. In the TD algorithm based on the martingale orthogonality condition, the episode-

wise form corresponding to the traditional TD(0) algorithm [29] selects $\xi_t = \partial v^\theta(t, X_t)/\partial\theta$ and uses stochastic approximation [30]:

$$\theta \leftarrow \theta + \alpha \int_0^T \frac{\partial v^\theta}{\partial \theta}(t, X_t)\mathrm{d}M_t^\theta \approx \theta + \alpha \sum_{i=0}^{K-1} \frac{\partial v^\theta}{\partial \theta}(t_i, X_{t_i})\delta_i^\theta, \tag{64}$$

where

$$\delta_i^\theta = v^\theta(t_{i+1}, X_{t_{i+1}}) - v^\theta(t_i, X_{t_i}) + f(t_i, X_{t_i})\Delta t. \tag{65}$$

For a randomized policy, the corresponding entropy-regularized local residual is

$$\begin{aligned}\delta_i^{\theta,\pi} = {} & v^\theta(t_{i+1}, X_{t_{i+1}}^{\pi,a}) - v^\theta(t_i, X_{t_i}^{\pi,a}) \\ & + \Big[f(t_i, X_{t_i}^{\pi,a}, a_i^\pi) + \gamma p\big(t_i, X_{t_i}^{\pi,a}, a_i^\pi, \\ & \qquad \pi(\cdot \mid t_i, X_{t_i}^{\pi,a})\big) - \beta v^\theta(t_i, X_{t_i}^{\pi,a})\Big]\Delta t.\end{aligned} \tag{66}$$

An online update uses the same increment at each time step. Similarly, a continuous-time TD(λ) form uses

$$\xi_t = \int_0^t \lambda^{t-s} \frac{\partial v^\theta}{\partial \theta}(s, X_s)\mathrm{d}s, \quad 0 < \lambda \leq 1. \tag{67}$$

The parameter λ controls the weight placed on historical predictions: $\lambda = 1$ resembles a Monte Carlo method, smaller λ emphasizes recent predictions, and $\lambda = 0$ recovers TD(0). In the continuous-time TD(λ) form, after one episode the offline update can be written as

$$\begin{aligned}\theta \leftarrow \theta + \alpha \sum_{i=0}^{K-1} \sum_{j=0}^{i} \Delta t\, \lambda^{(i-j)\Delta t} \frac{\partial v^\theta}{\partial \theta}(t_j, X_{t_j}) \\ \times \big[v^\theta(t_{i+1}, X_{t_{i+1}}) - v^\theta(t_i, X_{t_i}) + f(t_i, X_{t_i})\Delta t\big].\end{aligned} \tag{68}$$

The corresponding online update uses the same weighted eligibility trace but only up to the current time step. Although these algorithms are implemented on a grid with size Δt, they are discrete approximations to continuous-time moment equations. As shown in [11], as $\Delta t \to 0$, the solutions of the discrete moment equations converge to those of their continuous counterparts at first order in Δt.

Unlike a pathwise squared TD objective, the martingale-orthogonality method targets a zero expected inner product between δ_i^θ and test functions. The martingale noise has zero expectation under Itô integration, so the population orthogonality condition addresses the quadratic-variation difficulty in expectation and imposes weighted moment restrictions on the Bellman residual [11].

The preceding TD constructions use prescribed members of a test-function family. The randomized-policy martingale-adversarial Critic of Section 3.2 retains this weak-moment principle but learns a parameterized test function from local residuals. Let $\mathcal{E}_i^\theta$ denote the residual used at time t_i: it is δ_i^θ in the fixed-policy case and $\mathcal{R}_{\Delta t}^\theta(t_i, X_{t_i}, \varepsilon_i; \pi)$ in the randomized-policy case. For two independent mini-batches $\mathcal{B}_1$ and $\mathcal{B}_2$ sampled uniformly from the available time points, define

$$\begin{aligned} \widehat{G}^{(k)}_{\Delta t}(\theta,\eta) &:= \frac{T-\Delta t}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k}\rho_\eta(t_i,X_{t_i})\mathcal{E}^\theta_i, \qquad k=1,2,\\ \widehat{\mathcal{L}}_{\text{weak}}(\theta,\eta) &:= \big(\widehat{G}^{(1)}_{\Delta t}\big)^\top \widehat{G}^{(2)}_{\Delta t},\\ &\min_\theta \max_\eta \ \widehat{\mathcal{L}}_{\text{weak}}(\theta,\eta). \end{aligned} \tag{69}$$

This is the empirical weak-moment component. For fixed network parameters, independent batches target the squared moment; the empirical-pool and episode-split sampling identities are given in Appendix A.8. The implemented Critic combines this component with the trajectory anchor in (45). The Critic alternates descent in θ with ascent in η using currently available transitions or a small replay batch. The Actor is updated from a fresh batch with the current-policy residual gradient in Section 3.3; alternating the two steps gives the online Actor–Critic implementation. The general and mean–variance-specific online procedures are reported in Appendix B.2 and Algorithm A7, respectively.

The effect of using a prescribed rather than a learned test function is illustrated next in a one-dimensional example.

4.3. A One-Dimensional Illustration of Fixed and Learnable Test Functions

Problem setting (Example 4 in [11]). Let X_t be a standard Brownian motion with $X_0 = 0$, and let the terminal reward be $g(X_1) = X_1$. The exact value function is

$$v(t,x) = \mathbb{E}[g(X_1) \mid X_t = x] = x, \qquad t \in [0,1].$$

Consider the parametric approximation satisfying the terminal condition $v^\theta(1,x) = g(x)$:

$$v^\theta(t,x) = x + (1-t)\exp\left(\theta x - \frac{1}{2}\theta^2 t + \theta\right).$$

For this example, the mean-squared value error admits the closed form [11]

$$\text{MSVE}(\theta) = -\frac{e^{2\theta}\left(2 - 2e^{\theta^2} + 2\theta^2 + \theta^4\right)}{\theta^6}.$$

By Theorem 9, each prescribed test function ξ induces a moment condition in (63), which we denote by $G(\theta) = 0$. To compare fixed-test TD constructions with the learned-Test Critic, we consider the test processes associated with continuous-time TD(0) and TD(λ) at $\lambda = 1$:

$$\xi^{(0)}_t = \partial_\theta v^\theta(t, X_t), \qquad \xi^{(1)}_t = \int_0^t \partial_\theta v^\theta(s, X_s)\mathrm{d}s.$$

The associated expected moment conditions reduce to the one-dimensional equations $G_0(\theta) = 0$ and $G_1(\theta) = 0$ (closed forms as in Example 4 of [11]):

$$G_0(\theta) = \frac{e^{2\theta}}{\theta^5}\left(2 - \theta + \theta^2 - \theta^3 + e^{\theta^2}(-2 + \theta + \theta^2)\right),$$

$$G_1(\theta) = \frac{e^{2\theta}}{\theta^7}\left(6 + 2e^{\theta^2}(-3 + \theta + \theta^2) - (-1+\theta)\theta(-2 + 2\theta + \theta^3)\right).$$

Numerically solving the two equations gives $\hat{\theta}_0 \approx -1.8392$ and $\hat{\theta}_1 \approx -2.1257$. Direct minimization of MSVE(θ) gives the global minimizer $\theta^* \approx -2.1257$, which coincides with $\hat{\theta}_1$. Thus, in this example, continuous-time TD(1), corresponding to $\xi^{(1)}$, converges to the MSVE optimum, whereas continuous-time TD(0), corresponding to $\xi^{(0)}$, converges to a suboptimal point. This demonstrates the sensitivity of martingale-orthogonality TD algorithms to the choice of test function.

The martingale-adversarial formulation instead treats the test function as a learnable object and considers the objective

$$\min_{\theta} \sup_{\rho \in \mathcal{T}_b} \left(\mathbb{E} \int_0^1 \rho_t \mathrm{d}M_t^{\theta} \right)^2.$$

Here, $\mathcal{T}_b$ is a bounded neural-network-parameterized class of test functions. If ρ is fixed to be $\xi^{(0)}$ or $\xi^{(1)}$, the formulation reduces, respectively, to the moment conditions induced by TD(0) and TD(1); these are fixed-test special cases of the same weak-moment construction. In the numerical experiment, the adversarial method obtains $\theta \approx -2.1259$, which is better than the TD(0) fixed point $\hat{\theta}_0 \approx -1.8392$ and close to the theoretical optimum $\theta^* \approx -2.1257$; see the left panel of Figure 1.

Because practical TD and adversarial training rely on trajectory sampling and stochastic gradients, convergence curves are affected by sampling variance and random seeds. In this example, the expected moment conditions $G_0(\theta)$ and $G_1(\theta)$ have closed forms. We therefore construct a deterministic, noise-free mean-field baseline as a benchmark:

$$\theta_{k+1} = \theta_k - \alpha G_i(\theta_k), \qquad i \in \{0, 1\}.$$

This iteration represents an ideal update trajectory in which the expected moment condition can be computed exactly at every step. It characterizes the intrinsic convergence difficulty of the moment equation itself and provides a reproducible efficiency baseline for comparing different methods.

To compare the parameter sequences $\{\theta_k\}$ generated by different algorithms on the same scale, we use the analytically computable TD(1) moment residual:

$$\mathrm{Res}(\theta_k) := |G_1(\theta_k)|.$$

Since the global minimizer θ^* of MSVE coincides with the root of the TD(1) moment equation $G_1(\theta) = 0$ in this example, $\mathrm{Res}(\theta_k)$ provides a unified error scale for measuring how quickly different algorithms approach the MSVE optimum. The right panel of Figure 1 shows that, under the same initial value $\theta_0 = -1$ and step size $\alpha = 1$, TD(0) remains on a nonzero residual plateau corresponding to its suboptimal fixed point, while the plotted adversarial updates reduce this analytic residual in fewer counted iterations than the deterministic TD(1) sequence and approach a neighborhood of θ^*. In the plotted runs, the adversarial method reaches $|G_1(\theta_k)| < 10^{-5}$ within about 150 updates, whereas the deterministic TD(1) iteration requires about 341 iterations.

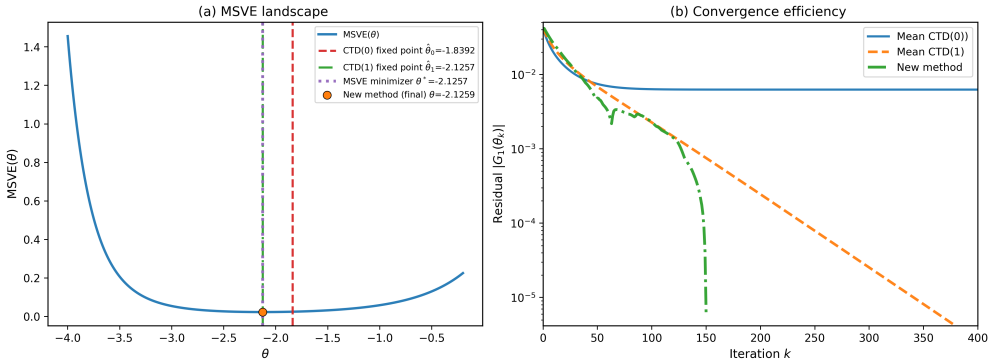


Figure 1. (**a**) MSVE(θ) and the fixed points induced by different moment conditions; (**b**) analytic residual $|G_1(\theta_k)|$ versus counted iterations for TD(0), TD(1), and the martingale-based adversarial method; the methods differ in computational work per iteration.

5. Financial Applications

We illustrate policy evaluation in high-dimensional basket-option pricing and policy optimization in mean–variance portfolio selection. The latter application also permits an examination of computational scalability and entropy regularization.

5.1. Basket-Option Pricing: Policy Evaluation

In this numerical experiment, we apply the fixed-policy evaluation algorithms under the continuous-time reinforcement learning framework to high-dimensional basket-option pricing. This problem corresponds to solving a high-dimensional parabolic partial differential equation [31,32]. We consider $d = 100$ underlying assets with constant pairwise correlation $c_{\text{corr}} = 0.5$. The option value function $v(t, x)$ satisfies

$$\frac{\partial v}{\partial t} + \sum_{i=1}^{d}(r - q_{\text{div}})x_i \frac{\partial v}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{d} c_{ij}\sigma^2 x_i x_j \frac{\partial^2 v}{\partial x_i \partial x_j} - rv = 0, \tag{70}$$

with terminal condition

$$v(T, x) = \left(\frac{1}{d}\sum_{i=1}^{d} x_i - K\right)^+.$$

Here, r is the risk-free rate, q_{div} is the dividend yield, σ is the volatility, c_{ij} is the correlation coefficient between assets i and j, and K is the strike price. We use the martingale-based numerical methods below to approximate the value function in this high-dimensional PDE setting.

Since a European basket call option pays off only at maturity T, the holder receives no cash flow during the life of the option $(t < T)$, and hence the running reward is zero. Under this fixed policy, the policy-evaluation task is essentially to compute

$$v(t, x) = \mathbb{E}\left[e^{-r(T-t)} g(X_T) \mid X_t = x\right], \qquad g(x) = \left(\frac{1}{d}\sum_{i=1}^{d} x_i - K\right)^+.$$

This is the discounted fixed-policy formulation with $\beta = r$, so the corresponding local residual contains the term $-rv$. To generate environment paths, we assume that the multi-asset price process $X_t \in \mathbb{R}^{100}$ follows a multidimensional geometric Brownian motion:

$$\mathrm{d}X_t = \mathrm{diag}(X_t)((r - q_{\text{div}})\mathbf{1}\,\mathrm{d}t + \sigma\,\mathrm{d}W_t), \tag{71}$$

where W_t is a correlated Brownian motion implemented by the Cholesky decomposition $W_t = CB_t$, B_t is a standard Brownian motion, and C is the Cholesky factor of the correlation matrix.

The experimental environment is specified as follows: initial state $X_0 = S_0\mathbf{1}_d$ with $S_0 = 1.0$, strike price $K = 0.9$, risk-free rate $r = 0.01$, dividend yield $q_{\text{div}} = 0$, volatility $\sigma = 0.4$, maturity $T = 2$ years, and time step $h = 0.04$ ($n_t = 50$). Exact geometric Brownian transitions generate the main training paths. A conditional Monte Carlo reference, obtained by integrating one Gaussian factor analytically and sampling the remaining factors with antithetic pairs, gives $v(0, S_0\mathbf{1}_d) \approx 0.216014$ with standard error 1.58×10^{-5}. The same reference construction supplies values and derivatives at the evaluation states; its numerical checks are reported in Appendix C.1.

Compared methods.

We compare the following three policy-evaluation methods:

- **ML (PE).** Offline policy evaluation based on the global martingale loss (56), using a single value network $v^\theta(t,x)$.
- **Adversarial TD (PE), denoted ADV.** Policy evaluation based on the local weak-martingale condition, with alternating value and bounded-test updates. The implemented value objective combines a two-batch cross-moment and a trajectory martingale-loss anchor; the latter uses discounted terminal payoffs from the sampled paths.
- **Deep BSDE (benchmark method [32]).** This method trains a learnable scalar initial value Y_0 and time-step-wise Delta networks, from which the diffusion increment is formed, so that the resulting Y_T matches the terminal payoff in the L^2 sense. We use $n_t = 50$ independent time-step-wise subnetworks.

Evaluation and training protocol.

Following the fairness convention in Section 5.3 of Jia and Zhou [11], all methods use 6.4×10^5 full training paths, obtained from 5000 outer iterations with the same batch size $B = 128$. All three methods use softplus activation. ML and ADV use the same $(128, 128, 64)$ value architecture and identical initial value-network weights within each seed. Deep BSDE uses $(128, 64)$ time-step-wise networks. Each outer iteration performs one ML or BSDE update, or four value and two Test updates for ADV; the additional local successors and actual computation times are recorded separately. The ADV value and Test learning rates are 10^{-3} and 5×10^{-4}; ML and BSDE use 10^{-3}. Appendix C.9 gives the optimizers, schedules, terminal-condition representation, sampling, and seed protocol.

The comparison includes all 20 fixed confirmation seeds. Training, monitoring, and final assessment use separate random streams. Parameters are also maintained by an exponential moving average (EMA) with decay 0.99, including the BSDE initial value and all its time-step networks. The comparison uses these EMA estimates throughout. Figure 2 reports seed medians and interquartile ranges, while Table 1 reports independent final-assessment means and sample standard deviations.

To simultaneously evaluate the approximation of the initial value, the global value function, and the hedging-gradient quantity, we use the following three error metrics, as in Jia and Zhou [11]:

$$\begin{aligned} &\text{initial price error:} && |v(0,X_0) - v^\theta(0,X_0)| / |v(0,X_0)|, \\ &\text{pathwise value RMSE:} && \sqrt{\mathbb{E}\int_0^T |v(t,X_t) - v^\theta(t,X_t)|^2 \,\mathrm{d}t}, \\ &\text{Delta-hedging RMSE:} && \sqrt{\mathbb{E}\int_0^T \|\nabla_x v(t,X_t) - \nabla_x v^\theta(t,X_t)\|^2 \,\mathrm{d}t}. \end{aligned}$$

The training curves in Figure 2 are computed using a unified evaluation protocol. For ML and ADV, we directly evaluate the network output $v^\theta(t_n, X_n)$ and use automatic differentiation to compute the hedging gradient $\nabla_x v^\theta(t_n, X_n)$. For Deep BSDE, on the same set of evaluation paths, we use the current EMA parameters to perform forward Euler propagation and obtain the sequence Y_n, which is then compared with the conditional reference values. The implied Delta is compared with the corresponding reference gradient. The path and Delta errors use the integrated definitions above.

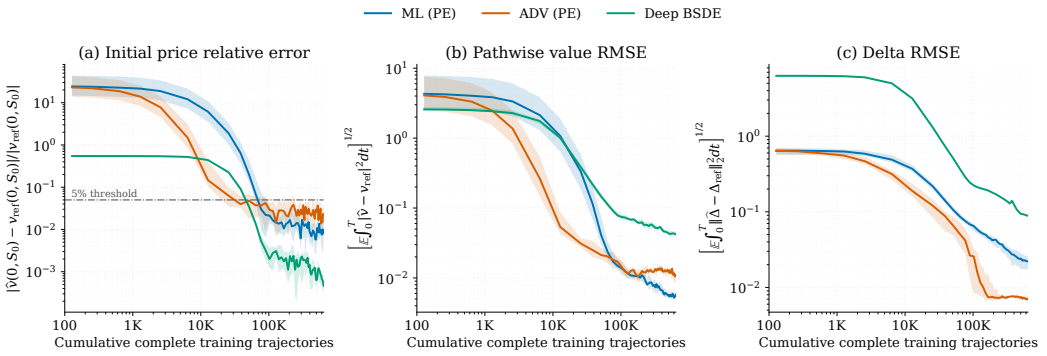


Figure 2. Policy evaluation for the $d = 100$ basket option. Solid curves are medians over 20 training seeds, and shading gives the 25th–75th percentiles. The three panels show initial relative price error, integrated path-value RMSE, and integrated Delta-vector RMSE. The horizontal axis counts full training paths.

Table 1. Basket-option errors after 5000 outer iterations: mean ± sample SD over 20 training seeds.

Method	Initial Price Error (%)	Path RMSE	Delta RMSE
ML (PE)	0.9273 ± 0.7721	$\mathbf{0.006315} \pm 0.001283$	0.020963 ± 0.006926
ADV (PE)	2.4728 ± 2.0170	0.010805 ± 0.003439	$\mathbf{0.008076} \pm 0.001625$
Deep BSDE	$\mathbf{0.0579} \pm 0.0439$	0.037649 ± 0.002903	0.094627 ± 0.000885

Bold values indicate the lowest mean error in each column.

Figure 2 and Table 1 jointly present the training dynamics and final accuracy of the three methods. The results show the following features.

- **Stronger global function approximation.** ML policy evaluation minimizes the global martingale loss and directly approximates the value function $v^\theta(t, x)$ as a joint function of time and state. Its mean path RMSE and Delta RMSE are, respectively, 83.2% and 77.8% lower than those of Deep BSDE, with ML attaining the lowest mean path RMSE. The path and Delta comparisons favor both martingale methods over Deep BSDE in all 20 paired seeds, with Holm-adjusted two-sided sign-test $p = 7.63 \times 10^{-6}$; Appendix C reports the statistical comparisons.
- **Good online extensibility.** ADV policy evaluation dynamically optimizes the test function ρ_η and thereby handles the martingale-orthogonality condition effectively. It attains the lowest mean Delta RMSE (0.008076 vs. 0.094627 for Deep BSDE), a reduction of 91.5%. Since its martingale-orthogonality term uses local one-step transitions, it naturally supports online updating and high-dimensional extension, providing a suitable algorithmic interface for deployment as a Critic module in the subsequent Actor–Critic framework. Its mean Delta error is also 61.5% below ML, with improvement in 19 of 20 paired seeds.
- **Better overall practicality.** Deep BSDE has an advantage in fitting the initial price Y_0 (mean relative error 0.0579%), consistent with its structural design, where Y_0 is parameterized as an independent trainable scalar. However, its time-step-wise Delta networks do not directly output a globally accessible value function $v^\theta(t, x)$. This initial-point advantage is accompanied by larger path and Delta errors than those of ML and ADV. For downstream tasks such as hedging, Greek sensitivity analysis, PDE post-processing, and Actor–Critic algorithms, martingale-driven methods output the global function $v^\theta(t, x)$ and its gradient $\nabla_x v^\theta(t, x)$, providing a common value representation for pricing, sensitivity analysis, and policy improvement.

We further evaluate the conventional single-asset second derivative $\Gamma_{11} = \partial^2 v / \partial x_1^2$ on two price slices at $t = 0.48, 1.00, 1.48$. The common-price slice sets $x = s\mathbf{1}_d$, and the individual-asset slice sets $x = (s, 1, \ldots, 1)$, for 25 values of s in $[0.7, 1.3]$. In both cases the

derivative remains with respect to x_1. Relative to Deep BSDE, ADV reduces the mean normalized L^2 Gamma error by 91.5% and 94.8% on the common-price and individual-asset slices, respectively. ADV has a lower error in all 20 paired seeds on both slices. Figure A1 displays the complete seed distribution; Appendix C.1 gives the price landscapes, reference checks, and absolute errors.

This experiment extends the comparison of Jia and Zhou [11] to the $d = 100$ basket-option setting. The martingale-driven PE methods improve path-value and Delta approximation, and the Gamma comparison further demonstrates the sensitivity information obtained from the learned representation. Appendix C reports the implementation, reference validation, bounded-test and Euler sensitivities, and measured computation times. Algorithm A5 gives the implemented ADV instance.

5.2. Mean–Variance Portfolio Selection: Policy Optimization

The mean–variance portfolio-selection problem is a cornerstone of modern financial theory. Since the seminal work of Markowitz [33], it has become a central paradigm for asset allocation. Let $u_t \in \mathcal{U} \subseteq \mathbb{R}^d$ denote a portfolio-weight action and let X_t denote portfolio wealth. In a high-dimensional MV problem, the objective is to minimize the variance of terminal wealth over a given investment horizon T, while achieving a prescribed expected target return:

$$\begin{aligned} \min_{u} \quad & \mathrm{Var}(X_T), \\ \text{s.t.} \quad & \mathbb{E}[X_T] = z. \end{aligned}$$

In practice, this formulation faces two fundamental challenges. First, its dynamic extension is affected by time inconsistency, so the classical Bellman optimality principle does not apply directly. Second, parameter-estimation error, especially the sensitivity of expected-return estimates, severely deteriorates out-of-sample performance.

In the continuous-time setting, Zhou and Li [34] resolved the time-inconsistency issue through an embedding technique with a Lagrange multiplier w, which transforms the original problem into the following pre-commitment problem:

$$\min_{u} \ \mathbb{E}\left[(X_T - w)^2\right] - (w - z)^2. \tag{72}$$

Here, w is the Lagrange multiplier used to adjust the mean constraint $\mathbb{E}[X_T] = z$. The cost-to-go $V_w^{\mathrm{c}}(t, x)$ of this problem satisfies the HJB equation

$$\partial_t V_w^{\mathrm{c}} + \inf_{u \in \mathcal{U}} \left\{ x u^\top \mu \, \partial_x V_w^{\mathrm{c}} + \frac{1}{2} x^2 u^\top \sigma \sigma^\top u \, \partial_{xx} V_w^{\mathrm{c}} \right\} = 0,$$

with terminal condition $V_w^{\mathrm{c}}(T, x) = (x - w)^2 - (w - z)^2$. This approach, however, relies heavily on precise knowledge of market-dynamics parameters. In high-dimensional applications, such parameters are extremely difficult to estimate: drift estimation requires very long historical samples [35], while covariance estimation suffers from the curse of dimensionality. Although shrinkage estimators [36] and Bayesian methods [37] have been proposed, estimation errors accumulate in dynamic environments and may lead to decisions that deviate substantially from optimality.

Reinforcement learning provides an alternative route for portfolio optimization. Rather than first estimating all model parameters and then optimizing a plug-in objective, an RL implementation can improve a policy from transitions generated by a specified environment. We use the entropy-regularized stochastic-control formulation of Wang et al. [21], embed martingale theory into the continuous-time RL framework, and apply the martingale-based Actor–Critic algorithm proposed above to the high-dimensional MV setting. In this

formulation, $\pi^{\phi}(\cdot \mid t, x)$ denotes a randomized policy over the portfolio actions $u \in \mathcal{U}$; it is distinct from the portfolio-weight action itself. In the sampled-action implementation, we write $a_t = u_t \sim \pi^{\phi}(\cdot \mid t, X_t)$.

We design a high-dimensional portfolio experiment with $d = 100$ risky assets. The asset prices follow a geometric Brownian motion with time-varying parameters:

$$\mathrm{d}S_t^i = S_t^i \left(\mu_t^i \mathrm{d}t + \sum_{j=1}^{d} \sigma_t^{ij} \mathrm{d}W_t^j \right), \quad i = 1, \ldots, d. \tag{73}$$

For a self-financing portfolio with weight u_t, the associated wealth process is

$$\mathrm{d}X_t = X_t u_t^{\top} \mu_t \, \mathrm{d}t + X_t u_t^{\top} \sigma_t \, \mathrm{d}W_t, \qquad X_0 = 1. \tag{74}$$

The drift vector μ_t and volatility matrix σ_t vary with the market state, thereby mimicking alternating bull and bear regimes. We generate 8000 prices (7999 daily return observations) with market seed 231 and pairwise asset correlation 0.1. The first 3999 returns form the training history and the last 4000 form the evaluation period, corresponding to 15.873 trading years at 252 days per year. A training episode contains 252 days, with an allocation held for 21 days between decisions. The entropy temperature is $\gamma = 0.01$, and the target terminal wealth for each training episode is $z = 1.15$. The schedule and all model parameters are given in Appendix C.9.

We compare the martingale-driven Actor–Critic algorithm with seven benchmark strategies: the market portfolio (Market), the equally weighted portfolio (EW), sample mean–variance (MV), distributionally robust mean–variance (DRMV), a constrained plug-in adaptation of the continuous-time mean–variance direction (CTMV), DDPG, and PPO. Market resets daily to equal weights, whereas EW resets every 21 days. MV and DRMV use a rolling 252-day estimation window with covariance shrinkage; CTMV normalizes the positive part of $\widehat{\Sigma}^{-1}\widehat{\mu}$ to respect the common allocation constraint. The three learning methods use a common shared-asset policy architecture and terminal MV objective.

All executed allocations satisfy $a_i \geq 0$ and $\sum_i a_i = 1$. The backtest charges a proportional rate $c = 0.001$ on the absolute traded notional at every rebalance, using the same self-financing accounting for every method. Specifically, if a^- is the current weight vector, the remaining wealth fraction k solves $k = 1 - c\|ka - a^-\|_1$. The initial allocation is free of charge. Training uses the terminal MV reward with entropy regularization; costs are then included in the portfolio execution and reported performance.

The observation contains episode progress, log wealth, three aggregate market features, and three lagged features per asset, giving dimension $3d + 5$. Each feature uses returns available strictly before the decision. The Actor shares one scoring network across assets, while the Critic and Test use aggregate summaries. This construction retains a d-component allocation while exploiting the common structure of the asset inputs.

To examine the two update regimes, the offline method is trained for 900 outer iterations on one-year blocks sampled from the training history, including 50 initial Critic/Test warmup iterations. The online method starts from the same pretraining protocol and adapts after each completed 21-day block. At that time, four updates draw from the most recent 1260 observed days and the fixed pretraining history in equal proportions, with learning rates scaled by 0.1. Both methods therefore use the same martingale-residual construction with the appropriate available history.

For the martingale-driven Actor–Critic algorithm, we retain the reward-maximization convention used in the preceding sections. Define the terminal reward by

$$g_w(x) := -(x-w)^2 + (w-z)^2. \tag{75}$$

For a randomized policy π, write X^π for the policy-level wealth process in (74) and set

$$v_w(t,x;\pi) = \mathbb{E}_t\left[g_w(X_T^\pi) + \gamma \int_t^T H(\pi_s)\,\mathrm{d}s\right], \qquad \beta = 0, \tag{76}$$

where, for the continuous portfolio-action distribution and a generic state x,

$$\begin{aligned} H(\pi(\cdot \mid t,x)) &:= -\int_{\mathcal{U}} \pi(u \mid t,x)\log \pi(u \mid t,x)\,\mathrm{d}u, \\ H(\pi_t) &:= H(\pi(\cdot \mid t, X_t^\pi)). \end{aligned} \tag{77}$$

The corresponding reward-form HJB equation is

$$\sup_\pi \{\mathcal{L}^\pi v_w + \gamma H(\pi(\cdot \mid t,x))\} = 0, \qquad v_w(T,x) = g_w(x), \tag{78}$$

where $\mathcal{L}^\pi$ is the infinitesimal generator of the wealth dynamics under π.

With the value function parameterized by v_w^θ, define the corresponding compensated process

$$M_t^{\theta,w,\pi} = v_w^\theta(t, X_t^\pi) + \int_0^t \gamma H(\pi_s)\,\mathrm{d}s. \tag{79}$$

For the numerical implementation, the differential entropy is centered at the uniform-simplex entropy and normalized by the number of assets,

$$\overline{H}(\pi) = \frac{H(\pi) - H(\mathrm{Dirichlet}(\mathbf{1}_d))}{d}. \tag{80}$$

The policy is Dirichlet, with concentrations $\alpha_i = 0.15 + 2d\ \mathrm{softmax}_i(\ell)$ and bounded shared-asset logits $\ell_i \in [-1,1]$. Randomized actions generate the training episodes; their conditional mean is used for the executed portfolio. In the formulas above, this numerical convention replaces γH by $\gamma\overline{H}$.

Let G denote the observed vector of 21-day asset gross returns. The held-allocation transition is

$$X' = Xa^\top G, \qquad a \sim \pi^\phi(\cdot \mid s), \tag{81}$$

where s includes the causal observation features. With duration $\Delta = 21/252$, the local residual is

$$R^{\theta,w} = v_w^\theta(s', X') - v_w^\theta(s, X) + \gamma\Delta\overline{H}(\pi^\phi(\cdot \mid s)). \tag{82}$$

This exact historical block mapping is differentiable with respect to the allocation. The Actor uses its pathwise residual derivative with the sampled starting observation and current Critic fixed, as in the transition formulation of the preceding theory.

The test combines nine fixed bounded features and five learned sine outputs. For independent episode groups, define $\widehat{G}^{(k)} = |\mathcal{B}_k|^{-1}\sum_{\mathcal{B}_k}\rho_\eta(s)R^{\theta,w}/\Delta$. The implemented value objective is

$$\widehat{\mathcal{L}}_{\mathrm{C}} = \frac{1}{14}(\widehat{G}^{(1)})^\top \widehat{G}^{(2)} + 0.25\widehat{A}_{\mathrm{mom}}, \tag{83}$$

with descent in θ and ascent in η for the cross-moment term. The anchor $\widehat{A}_{\mathrm{mom}}$ fits the first and second terminal gross-return moments and accumulated entropy, as specified in Appendix C.9. The multiplier is updated every five outer iterations by

$$w \leftarrow \mathrm{proj}_{[0,10]}\{w - 0.25(\widehat{\mathbb{E}}[X_T] - z)\}. \tag{84}$$

Algorithms A6 and A7 give the offline and causal online implementations.

We compare the wealth trajectories over 100 fixed training seeds, 10001–10100, for each learning method and each mode. All runs use the same market path, and all 600 fitted policies are included. Figures 3 and 4 show mean wealth and the pointwise 10th–90th percentile range across training seeds. The horizontal axis is elapsed trading years; the gray interval marks the simulated bear regime. The proposed AC strategy finishes with the highest mean terminal wealth in both modes.

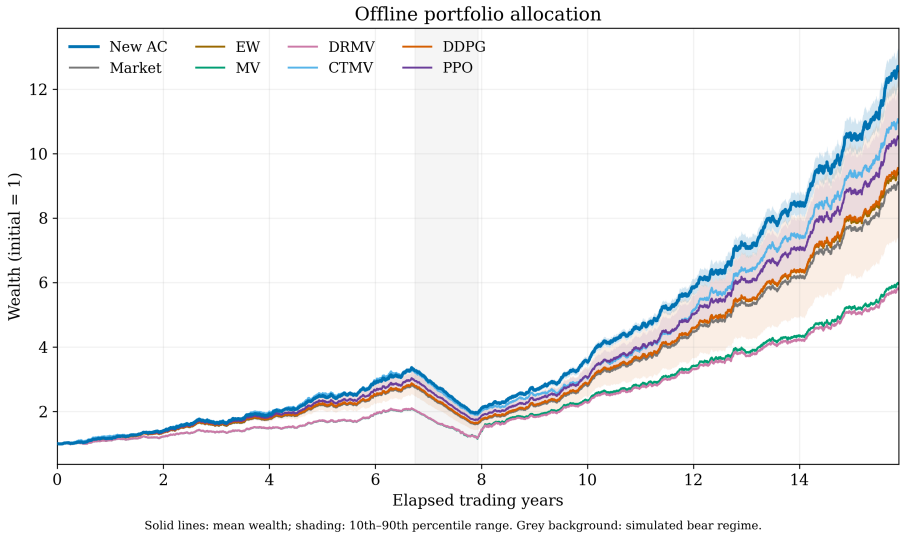


Figure 3. Offline portfolio allocation on the common evaluation market. Solid curves show mean wealth over all 100 training seeds, initialized at one; shading shows the pointwise 10th–90th percentiles. The gray background denotes the simulated bear regime. The bands describe training variability conditional on this market path.

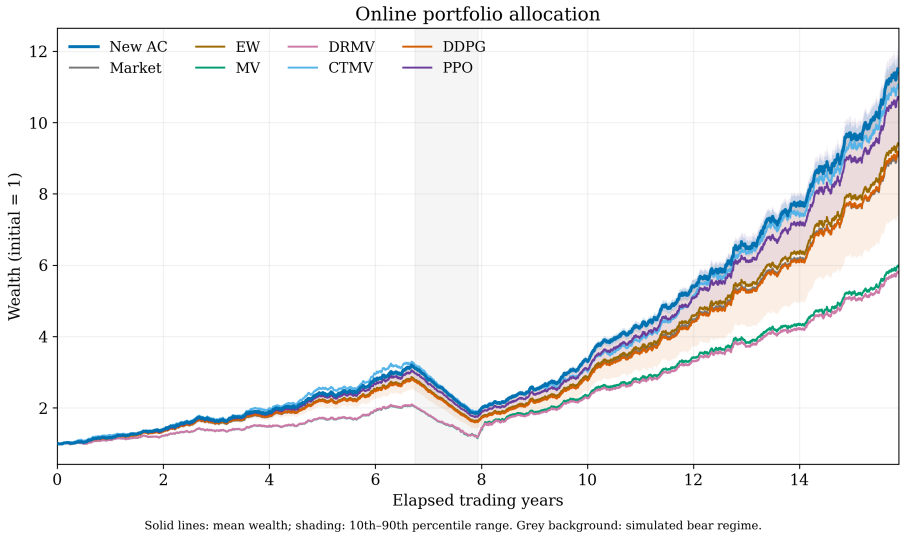


Figure 4. Online portfolio allocation with causal updates after completed holding periods. Curves, bands, and background shading use the same definitions and the same fixed market as Figure 3, including transaction costs.

Although these figures provide a high-level comparison, a more detailed assessment uses the financial criteria in Tables 2 and 3. New AC attains the highest mean annualized return, and Sharpe, Sortino, and Calmar ratios, together with the lowest mean maximum drawdown, in both modes. Offline annualized return is $16.223 \pm 0.257\%$, compared with $14.998 \pm 0.596\%$ for PPO and $14.298 \pm 1.190\%$ for DDPG. The corresponding online values are $15.593 \pm 0.178\%$, $15.108 \pm 0.581\%$, and $14.072 \pm 1.073\%$. The smaller across-seed dispersion of New AC accompanies its higher mean performance.

Table 2. Performance in the offline setting: means and sample standard deviations across 100 training seeds. Boldface denotes the best mean in each column.

Method	Ann. Ret. (%)	Sharpe	Sortino	Calmar	MDD (%)	RT (Years)
New AC	**16.223** ±0.257	**2.377** ±0.017	**3.869** ±0.032	**0.384** ±0.007	**42.248** ±0.110	1.965 ±0.017
Market	14.105 ±0.000	2.195 ±0.000	3.549 ±0.000	0.324 ±0.000	43.512 ±0.000	2.079 ±0.000
EW	14.315 ±0.000	2.227 ±0.000	3.605 ±0.000	0.330 ±0.000	43.438 ±0.000	2.060 ±0.000
MV	11.508 ±0.000	1.573 ±0.000	2.490 ±0.000	0.255 ±0.000	45.119 ±0.000	**1.583** ±0.000
DRMV	11.312 ±0.000	1.590 ±0.000	2.498 ±0.000	0.254 ±0.000	44.478 ±0.000	1.623 ±0.000
CTMV	15.329 ±0.000	2.206 ±0.000	3.537 ±0.000	0.358 ±0.000	42.838 ±0.000	2.167 ±0.000
DDPG	14.298 ±1.190	2.179 ±0.117	3.521 ±0.206	0.329 ±0.033	43.533 ±0.770	2.100 ±0.085
PPO	14.998 ±0.596	2.286 ±0.049	3.713 ±0.089	0.348 ±0.016	43.127 ±0.263	2.039 ±0.019

Each cell displays the mean above its ±SD. Classical strategies are deterministic on the common market path, so their between-training-seed SD is zero. Ann. ret. is the annualized arithmetic mean return; RT is the time from the maximum-drawdown trough to recovery of the preceding peak. Costs are included in all reported paths.

Table 3. Performance in the online setting: means and sample standard deviations across 100 training seeds. Boldface denotes the best mean in each column.

Method	Ann. Ret. (%)	Sharpe	Sortino	Calmar	MDD (%)	RT (Years)
New AC	**15.593** ±0.178	**2.337** ±0.014	**3.795** ±0.026	**0.366** ±0.005	**42.605** ±0.145	1.992 ±0.016
Market	14.105 ±0.000	2.195 ±0.000	3.549 ±0.000	0.324 ±0.000	43.512 ±0.000	2.079 ±0.000
EW	14.315 ±0.000	2.227 ±0.000	3.605 ±0.000	0.330 ±0.000	43.438 ±0.000	2.060 ±0.000
MV	11.508 ±0.000	1.573 ±0.000	2.490 ±0.000	0.255 ±0.000	45.119 ±0.000	**1.583** ±0.000
DRMV	11.312 ±0.000	1.590 ±0.000	2.498 ±0.000	0.254 ±0.000	44.478 ±0.000	1.623 ±0.000
CTMV	15.329 ±0.000	2.206 ±0.000	3.537 ±0.000	0.358 ±0.000	42.838 ±0.000	2.167 ±0.000
DDPG	14.072 ±1.073	2.151 ±0.109	3.470 ±0.191	0.323 ±0.029	43.687 ±0.694	2.117 ±0.085
PPO	15.108 ±0.581	2.295 ±0.047	3.730 ±0.085	0.351 ±0.016	43.074 ±0.261	2.038 ±0.019

Each cell displays the mean above its ±SD. Classical strategies are deterministic on the common market path, so their between-training-seed SD is zero. Ann. ret. is the annualized arithmetic mean return; RT is the time from the maximum-drawdown trough to recovery of the preceding peak. Costs are included in all reported paths.

Recovery time provides a complementary comparison. MV and DRMV recover sooner from their drawdown troughs, while New AC combines a smaller maximum drawdown

with substantially greater terminal wealth. Between the two AC modes, the offline implementation has the higher mean return and risk-adjusted ratios on this market. The online implementation maintains competitive performance while incorporating newly observed history through sequential adaptation.

In this high-dimensional mean–variance experiment, the martingale-driven AC method learns directly from historical transition blocks. Its residual gradient uses the known allocation-to-wealth map (81); a separate plug-in estimate of the full drift vector and covariance matrix is not required by this update. Appendix C.5 reports the actual Critic/Test trajectories, and Appendix C.6 examines the three prescribed training factors.

5.3. Computational Scalability and Entropy-Regularization Sensitivity

We next examine computational scalability and sensitivity to entropy regularization in the high-dimensional mean–variance portfolio problem.

5.3.1. Computational Complexity in the MV Problem

We report operation counts for specified components and measured training times separately.

Proposition 2 (Computational complexity in the MV problem)**.** *Consider the shared-asset implementation with d risky assets, fixed hidden widths, fixed batch size, fixed inner-update counts and a fixed number of transitions per update. Suppose that the asset-return blocks have been prepared, the policy applies the same fixed-width scoring network to each asset, and the value and test networks use fixed-size aggregated features. If allocation sampling and its first-order pathwise differentiation have expected cost $\mathcal{O}(d)$ and its log-density and entropy have cost $\mathcal{O}(d)$, one learning iteration has expected cost $\mathcal{O}(d)$. The shared-network parameter counts are independent of d, while the asset observations and allocations require $\mathcal{O}(d)$ storage per transition. For comparison, dense sample-covariance estimation uses $\mathcal{O}(nd^2)$ arithmetic and $\mathcal{O}(d^2)$ storage for n observations, and a dense covariance factorization uses $\mathcal{O}(d^3)$ arithmetic.*

For a measured time-to-performance comparison, we use common Sharpe thresholds of 2.25, 2.30, and 2.35. We run New AC, PPO, and DDPG serially on one Tesla T4, using 20 fixed seeds per method and the common $d = 100$ offline market. Each run has a 180 s budget. A target is confirmed after three consecutive nominal five-second evaluations satisfy the Sharpe threshold; the actual evaluation completion time is used. Table 4 reports the first observed time by which at least 10 of the 20 runs confirm the target, retaining every run in the calculation.

Table 4. Median time in seconds to common portfolio-performance targets, using all 20 runs per method ($d = 100$, offline, one Tesla T4).

Method	Sharpe $\geq$ 2.25	Sharpe $\geq$ 2.30	Sharpe $\geq$ 2.35
New AC	**30.27**	**35.28**	**40.27**
PPO	55.28	80.27	100.24
DDPG	105.28	$>$ 180	$>$ 180

Non-hits are right-censored at 180 s; $>$180 means that the median target time is beyond the budget. Time includes learner/optimizer initialization, Critic/Test warmup, all training, scheduled evaluations, and in-loop logging. Bold values indicate the shortest median time in each column.

At the 2.25 target, all three methods have observable median target times, with New AC retaining the shortest time. At the 2.30 and 2.35 targets, New AC requires 56.0% and 59.8% less median time than PPO, respectively. Separate fixed-workload runs at $d = 10, 100, 1000$ give mean training times of 102.738, 102.196, and 122.610 s for 900 outer

iterations (Table A8). The shared-asset implementation therefore exhibits a 19.34% training-time increase when the number of assets grows by a factor of 100.

These measurements complement the operation counts by distinguishing training-workload scaling from time to a common performance target.

The covariance-based MV, DRMV, and CTMV implementations store a dense $d \times d$ matrix. Their covariance estimation and dense factorization require quadratic storage and the polynomial operation counts stated above. Parameter sharing and direct differentiation of the observed wealth map allow the neural implementations to exploit the asset structure without this covariance factorization in the policy update.

5.3.2. Sensitivity to Entropy Regularization

We examine the sensitivity of the offline mean–variance algorithm to the entropy regularization parameter γ. The parameter controls the exploration–exploitation tradeoff in the randomized-control framework [21]; the implemented normalization is given by (80).

For the offline variant, we test $\gamma \in \{0.005, 0.01, 0.02, 0.05, 0.1\}$ using the same 100 training seeds at every temperature. All hyperparameters, the market path, and the execution rule are kept fixed. The 100 default records are reused, and each of the four alternative temperatures is trained for all 100 seeds. The resulting 500 records enter the descriptive comparison. For each alternative, we compute six paired differences from the default $\gamma = 0.01$ and apply a two-sided Wilcoxon signed-rank test. The 24 comparisons form one Holm correction family. The repeated unit is a training seed; the statistical procedure and adjusted p-values are given in Appendix C.7.

Figure 5 and Table 5 show a consistent range of strong performance across the tested temperatures. Mean annualized returns range from 16.125% to 16.345%. At $\gamma = 0.05$, the mean Sharpe ratio increases from 2.377 to 2.401, and its across-seed SD decreases from 0.017 to 0.011. The default $\gamma = 0.01$ gives the lowest mean maximum drawdown (42.248%). The $\gamma = 0.05$ return and Sharpe improvements have Holm-adjusted $p = 4.27 \times 10^{-7}$ and 4.19×10^{-16}, respectively; its MDD increases to 42.397%, while its Calmar difference has adjusted $p = 0.0854$. The analysis thus quantifies the tradeoff between mean reward, sensitivity to training randomness, and drawdown when selecting a fixed temperature.

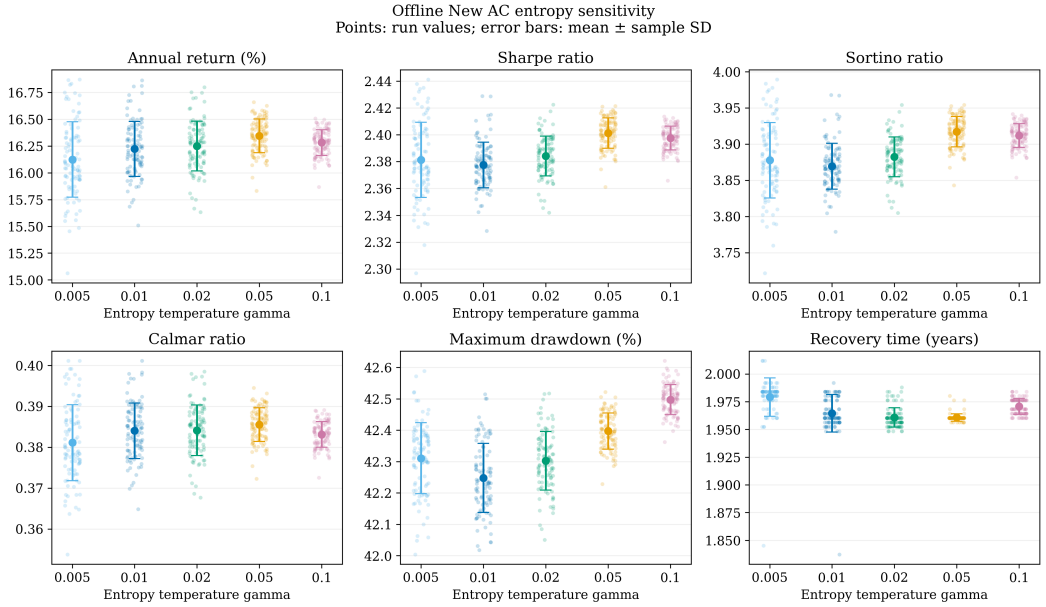


Figure 5. Offline New AC entropy-temperature sensitivity. Small points show all 100 run values per temperature; large points and error bars show means and sample standard deviations. Each temperature uses the same fixed market and the same training seeds.

Table 5. Offline entropy-temperature sensitivity: mean and sample SD over the same 100 training seeds for each temperature. An asterisk marks a difference from $\gamma = 0.01$ at Holm-adjusted $p \leq 0.05$; the sign and size of each difference are reported in Table A7.

γ	Ann. Ret. (%)	Sharpe	Sortino	Calmar	MDD (%)	RT (Years)
0.005	16.125 * ±0.353	2.381 ±0.028	3.878 ±0.052	0.381 * ±0.009	42.310 * ±0.113	1.979 * ±0.017
0.01	16.223 ±0.257	2.377 ±0.017	3.869 ±0.032	0.384 ±0.007	42.248 ±0.110	1.965 ±0.017
0.02	16.249 * ±0.232	2.384 * ±0.015	3.882 * ±0.027	0.384 ±0.006	42.302 * ±0.093	1.961 * ±0.009
0.05	16.345 * ±0.156	2.401 * ±0.011	3.917 * ±0.021	0.386 ±0.004	42.397 * ±0.058	1.961 * ±0.003
0.1	16.282 ±0.121	2.398 * ±0.009	3.912 * ±0.016	0.383 ±0.003	42.497 * ±0.048	1.971 * ±0.007

Each cell displays the mean above its ±SD. All configurations share one market path, execution costs, and seeds. The default $\gamma = 0.01$ reuses the 100 main-experiment records; the other four temperatures contribute 400 new runs. The preferred direction of change depends on the metric.

6. Conclusions

This paper develops a martingale-based continuous-time reinforcement learning framework for high-dimensional financial problems. Policy evaluation, policy gradient, and entropy-regularized Actor–Critic learning are organized around a common martingale-residual principle. For the policy-gradient component, we show that, as the local time step vanishes, the rescaled derivative of the martingale-residual objective is asymptotically equivalent to the classical continuous-time policy gradient.

The framework also yields a weak-martingale adversarial Critic in which the test function is learned jointly with the value approximation, rather than prescribed in advance. This construction removes the need to select a fixed test function in martingale-orthogonality TD methods. The resulting implementations work with sampled martingale residuals and first-order automatic differentiation, without explicit evaluation of Hessians or other higher-order differential operators, and support both batch learning from complete trajectories and sequential updating from simulated transitions.

The numerical studies illustrate these constructions in two financial settings. In high-dimensional basket-option pricing, the martingale-based policy-evaluation methods provide accurate value estimates. The 20-seed comparison shows lower pathwise value and Delta errors for the martingale-based methods than for the evaluated Deep BSDE implementation, with complementary strengths of ML and the adversarial Critic. In the mean–variance portfolio experiment, the offline Actor–Critic implementation attains a mean annualized return of 16.223%, while the online implementation attains a mean Sharpe ratio of 2.337. The proposed method also reaches the common portfolio-performance targets in shorter median time than PPO in the controlled comparison. When the asset dimension expands from 10 to 1000, the fixed-budget training time rises from 102.7 to 122.6 s under the shared-network implementation. These results illustrate the use of the framework for policy evaluation and policy gradient in the stated high-dimensional simulation settings.

The theoretical guarantees connect compatible conditional moments and trajectory regression to Critic identification, local anchor curvature to adversarial stability, and Critic derivative accuracy to policy improvement. The policy-improvement step uses a differentiable transition interface, supplied by the model where required and given directly by the observed-return wealth map in the portfolio implementation. These conditions provide the setting for the error and local convergence results. Several directions merit further study. The analysis motivates extensions to jump dynamics, adaptive entropy selection, and evolving market environments. It would also be useful to extend the framework to more general diffusion structures and to non-Markovian or multi-agent financial-control problems. Finally,

combining martingale-based adversarial learning with value- or Q-function formulations may lead to new continuous-time Q-learning methods for high-dimensional stochastic control.

Author Contributions: Conceptualization, H.Q., D.X., M.D., and X.G.; methodology, H.Q., D.X., M.D., and X.G.; validation, H.Q. and D.X.; formal analysis, H.Q. and D.X.; investigation, D.X.; writing—original draft, H.Q., D.X., M.D., and X.G.; writing—review and editing, H.Q., D.X., M.D., and X.G.; supervision, H.Q., M.D., and X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AC, Actor–Critic; CTD, continuous-time temporal difference; CTMV, continuous-time mean–variance; HJB, Hamilton–Jacobi–Bellman; ML, martingale loss; MSBE, mean-squared Bellman error; MSVE, mean-squared value error; PE, policy evaluation; PG, policy gradient; RL, reinforcement learning; SDE, stochastic differential equation; TD, temporal difference.

Appendix A. Proofs of Theoretical Results

Appendix A.1. Proof of Theorem 1

Applying Itô's formula to the value function in (12), with all derivatives below evaluated at (t, X_t), we obtain

$$\begin{aligned} \mathrm{d}v(t,X_t) = {} & \partial_t v(t,X_t)\mathrm{d}t \\ & + \sum_{i=1}^{d} \partial_{x_i} v(t,X_t)\mathrm{d}X_t^i \\ & + \frac{1}{2}\sum_{i,j=1}^{d} v_{x_i x_j}(t,X_t)\mathrm{d}X_t^i \mathrm{d}X_t^j. \end{aligned} \tag{A1}$$

Substituting the SDE (8) into (A1) and using

$$\begin{aligned} \mathrm{d}X_t^i \mathrm{d}X_t^j &= \left(\mu^i \mathrm{d}t + \sum_{k=1}^{m} \sigma^{ik}\mathrm{d}W_t^k\right)\left(\mu^j \mathrm{d}t + \sum_{l=1}^{m} \sigma^{jl}\mathrm{d}W_t^l\right) \\ &= \sum_{k=1}^{m} \sigma^{ik}\sigma^{jk}\mathrm{d}t = (\sigma\sigma^\top)^{ij}\mathrm{d}t, \end{aligned}$$

we get

$$\mathrm{d}v(t,X_t) = \mathcal{L}v(t,X_t)\mathrm{d}t + (\nabla_x v(t,X_t))^\top \sigma(t,X_t)\mathrm{d}W_t. \tag{A2}$$

Combining this identity with the PDE (10) gives

$$\mathrm{d}v(t,X_t) = -f(t,X_t)\mathrm{d}t + (\nabla_x v(t,X_t))^\top \sigma(t,X_t)\mathrm{d}W_t. \tag{A3}$$

Integrating from s to t yields

$$\begin{aligned} v(t,X_t) - v(s,X_s) = {} & -\int_s^t f(r,X_r)\mathrm{d}r \\ & + \int_s^t \left(\nabla_x v(r,X_r)\right)^\top \sigma(r,X_r)\mathrm{d}W_r. \end{aligned} \tag{A4}$$

Therefore,

$$\begin{aligned}\Delta M^v_{s,t} &:= M^v_t - M^v_s \\ &= v(t, X_t) - v(s, X_s) + \int_s^t f(r, X_r)\mathrm{d}r \\ &= \int_s^t \left(\nabla_x v(r, X_r)\right)^\top \sigma(r, X_r)\mathrm{d}W_r.\end{aligned}$$

Under the stated integrability conditions, the last term is a square-integrable martingale increment [25]. The value function also satisfies $v(T, x) = g(x)$. Conversely, suppose that a candidate u satisfies this terminal condition and that M^u is a square-integrable martingale for every admissible starting point. Conditioning its terminal value at time t gives $u(t, x) = \mathbb{E}_{t,x}[g(X_T) + \int_t^T f(s, X_s)\mathrm{d}s]$, the fixed-policy value in (9). Uniqueness identifies $u = v$. Without the terminal condition, $u = v + c$ is a counterexample for every constant c.

Appendix A.2. Proof of Theorem 2

Assume that the conditions stated in the theorem hold. Since $X^s_s = \hat{X}_s$, Itô's formula and the definition of R give

$$\mathbb{E}[\Delta M^v_{s,s+h} \mid \hat{X}_s] = \int_s^{s+h} \mathbb{E}[R(r, X^s_r; v) \mid \hat{X}_s]\mathrm{d}r. \tag{A5}$$

Applying Dynkin's formula to $R(\cdot, \cdot; v)$ along the process $\{X^s_u : u \in [s, T]\}$ yields, for $r \in [s, s + h]$,

$$\mathbb{E}[R(r, X^s_r; v) \mid \hat{X}_s] = R(s, \hat{X}_s; v) + \mathbb{E}\left[\int_s^r \mathcal{L}R(u, X^s_u; v)\mathrm{d}u \,\middle|\, \hat{X}_s\right]. \tag{A6}$$

The regularity and growth assumptions imply that the second term is $O(r - s)$ uniformly over the short interval. Substitution into (A5) therefore gives

$$\begin{aligned}\mathbb{E}[\Delta M^v_{s,s+h} \mid \hat{X}_s] &= \int_s^{s+h} \big(R(s, \hat{X}_s; v) + O(r - s)\big)\mathrm{d}r \\ &= hR(s, \hat{X}_s; v) + O(h^2),\end{aligned} \tag{A7}$$

which proves (17).

Appendix A.3. Proof of Theorem 4

Differentiating the PDE in Lemma 1 with respect to the policy parameter ϕ gives a new PDE system for $\mathfrak{g}(t, x; \phi) = \partial_\phi v(t, x; \pi^\phi)$:

$$\begin{aligned}0 = \int_{\mathcal{A}} \Big\{ &\big[\mathcal{L}^a \mathfrak{g}(t, x; \phi) - \beta \mathfrak{g}(t, x; \phi) \\ &\quad + \gamma q_\phi(t, x, a; \phi)\big]\pi^\phi(a|t, x) \\ &+ \big[\mathcal{L}^a v(t, x; \pi^\phi) + f(t, x, a) \\ &\quad + \gamma p(t, x, a, \pi^\phi(\cdot|t, x)) - \beta v(t, x; \pi^\phi)\big] \\ &\quad \times \frac{\partial \pi^\phi}{\partial \phi}(a|t, x)\Big\}\mathrm{d}a,\end{aligned} \tag{A8}$$

with terminal condition $\mathfrak{g}(T, x; \phi) = 0$. Here, $q_\phi(t, x, a; \phi) = \partial_\phi p(t, x, a, \pi^\phi(\cdot|t, x))$, mapping $[0, T] \times \mathbb{R}^d \times \mathcal{A} \times \Phi$ into $\mathbb{R}^{L_\phi}$. Define the new reward function

$$\begin{aligned}\hat{f}(t, x, a, \phi) = &\left[\mathcal{L}^a v(t, x; \pi^\phi) + f(t, x, a)\right. \\ &\left. + \gamma p(t, x, a, \pi^\phi(\cdot|t, x)) - \beta v(t, x; \pi^\phi)\right] \\ &\times \frac{\partial_\phi \pi^\phi(a|t, x)}{\pi^\phi(a|t, x)} + \gamma q_\phi(t, x, a; \phi) \\ = &\left[\mathcal{L}^a v(t, x; \pi^\phi) + f(t, x, a)\right. \\ &\left. + \gamma p(t, x, a, \pi^\phi(\cdot|t, x)) - \beta v(t, x; \pi^\phi)\right] \\ &\times \partial_\phi \log \pi^\phi(a|t, x) + \gamma q_\phi(t, x, a; \phi).\end{aligned}$$

This is also a function from $[0, T] \times \mathbb{R}^d \times \mathcal{A} \times \Phi$ to $\mathbb{R}^{L_\phi}$. Therefore, (A8) can be rewritten as

$$\begin{aligned}&\int_{\mathcal{A}} \left[\mathcal{L}^a \mathfrak{g}(t, x; \phi) - \beta \mathfrak{g}(t, x; \phi) + \hat{f}(t, x, a, \phi)\right] \pi^\phi(a|t, x) \mathrm{d}a = 0, \\ &\mathfrak{g}(T, x; \phi) = 0.\end{aligned} \tag{A9}$$

Using the Feynman–Kac formula,

$$\begin{aligned}\mathfrak{g}(t, x; \phi) =&\mathbb{E}^{\mathbb{P}}\left[\int_t^T e^{-\beta(s-t)} \hat{f}(s, X_s^{\pi^\phi}, a_s^{\pi^\phi}; \phi) \mathrm{d}s \,\middle|\, X_t^{\pi^\phi} = x\right] \\ =&\mathbb{E}^{\mathbb{P}}\left[\int_t^T e^{-\beta(s-t)} \int_{\mathcal{A}} \hat{f}(s, X_s^{\pi^\phi}, a; \phi) \pi^\phi(a|s, X_s^{\pi^\phi}) \mathrm{d}a \mathrm{d}s \,\middle|\, X_t^{\pi^\phi} = x\right].\end{aligned} \tag{A10}$$

Because the reward $\hat{f}$ contains the operator $\mathcal{L}^a$, it depends on unknown environment dynamics and cannot be directly observed or computed by the agent. Applying Itô's formula to $v(t + \Delta t, X_{t+\Delta t}^a; \pi^\phi) - v(t, X_t; \pi^\phi)$ and using the martingale property replaces the generator term by observable value increments plus a martingale term. Substituting this representation into (A10) gives (33).

Appendix A.4. Proof of Theorem 5

By the first-order condition in the theorem, $\mathfrak{g}(t, x; \phi^*) = 0$ for every $(t, x) \in [0, T] \times \mathbb{R}^d$. From (A8),

$$\begin{aligned}0 = &\int_{\mathcal{A}} \left[\left(\mathcal{L}^a v(t, x; \pi^{\phi^*}) + f(t, x, a)\right.\right. \\ &\left. + \gamma p(t, x, a, \pi^{\phi^*}(\cdot|t, x)) - \beta v(t, x; \pi^{\phi^*})\right) \frac{\partial \pi^{\phi^*}}{\partial \phi}(a|t, x) \\ &\left.\left. + \gamma q_\phi(t, x, a; \phi^*) \pi^{\phi^*}(a|t, x)\right] \mathrm{d}a \right. \\ = &\int_{\mathcal{A}} \hat{f}(t, x, a, \phi^*) \pi^{\phi^*}(a|t, x) \mathrm{d}a.\end{aligned} \tag{A11}$$

Equation (A11) is the same type of PDE as (25). By the same argument as in Theorem 3, we obtain

$$0 = \mathbb{E}^{\mathbb{P}}\Bigg[\int_0^T \kappa_s \Bigg\{\Big[\partial_\phi \log \pi^{\phi^*}(a_s^{\pi^{\phi^*}}|s, X_s^{\pi^{\phi^*}}) + \zeta_s\Big] \times \Big[\mathrm{d}v(s, X_s^{\pi^{\phi^*}}; \pi^{\phi^*}) + \big(f(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}) + \gamma p(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}, \pi^{\phi^*}(\cdot|s, X_s^{\pi^{\phi^*}})) - \beta v(s, X_s^{\pi^{\phi^*}}; \pi^{\phi^*})\big)\mathrm{d}s\Big] + \gamma q_\phi(s, X_s^{\pi^{\phi^*}}, a_s^{\pi^{\phi^*}}; \phi^*)\mathrm{d}s\Bigg\} \;\Big|\; X_0^{\pi^{\phi^*}} = x\Bigg]. \tag{A12}$$

for any $\kappa \in L^2_{\mathcal{F}^{X^{\pi^{\phi^*}}}}([0,T];\mathbb{R})$ and $\zeta \in L^2_{\mathcal{F}^{X^{\pi^{\phi^*}}}}([0,T];\mathbb{R}^{L_\phi})$.

Appendix A.5. Proof of Theorem 6

To establish the asserted asymptotic equivalence, define the current-policy derivative

$$D_\phi F(\pi^\phi) := \nabla_\psi F(\pi^\psi)\big|_{\psi=\phi}.$$

The exact Critic $v^\theta = v(\cdot,\cdot;\pi^\phi)$ and the law of the pilot process $\hat{X}^{\pi^\phi}$ remain fixed under D_ϕ.

By interchangeability of expectation and differentiation,

$$D_\phi L_h = \int_0^{T-h} e^{-\beta t}\mathbb{E}\big[D_\phi \mathcal{R}_h^\theta(t, \hat{X}_t^{\pi^\phi}, \varepsilon; \pi^\phi)\big]\mathrm{d}t. \tag{A13}$$

For notational convenience, write $Y_{t+h} = \hat{X}_t^{\pi^\phi} + h\bar{\mu}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi) + \bar{\sigma}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi)\sqrt{h}\,\varepsilon$ for the local Euler propagation in (30). Differentiating the residual in its policy argument gives

$$D_\phi \mathcal{R}_h^\theta = \nabla_x v^\theta(t+h, Y_{t+h}) \cdot D_\phi Y_{t+h} + hD_\phi \tilde{q}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi), \tag{A14}$$

where

$$D_\phi Y_{t+h} = hD_\phi \bar{\mu}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi) + \sqrt{h}\, D_\phi \bar{\sigma}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi)\varepsilon. \tag{A15}$$

For each policy coordinate j, conditional Gaussian integration by parts transforms $h^{-1/2}\mathbb{E}[\nabla_x v^\theta(t+h, Y_{t+h})^\top(\partial_{\phi_j}\bar{\sigma})\varepsilon \mid \mathcal{F}_t]$ into $\frac{1}{2}(\partial_{\phi_j}\bar{a}) : \mathbb{E}[\nabla_x^2 v^\theta(t+h, Y_{t+h}) \mid \mathcal{F}_t]$. The derivative-level estimates (A34), with their integrable envelopes, then give

$$\mathbb{E}[D_\phi \mathcal{R}_h^\theta|\mathcal{F}_t] = h\nabla_x v^\theta(t, \hat{X}_t^{\pi^\phi}) \cdot D_\phi \bar{\mu}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi) \tag{A16}$$

$$+ \frac{h}{2}\mathrm{Tr}\Big(D_\phi \bar{a}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi)\nabla_x^2 v^\theta(t, \hat{X}_t^{\pi^\phi})\Big) \tag{A17}$$

$$+ hD_\phi \tilde{q}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi) + O(h^2). \tag{A18}$$

Integrating over $t \in [0, T-h]$ and taking full expectation, Fubini's theorem yields

$$D_\phi L_h = h\mathbb{E}\int_0^{T-h} e^{-\beta t}\Bigg[\nabla_x v^\theta(t, \hat{X}_t^{\pi^\phi}) \cdot D_\phi \bar{\mu}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi) \tag{A19}$$

$$+ \frac{1}{2}\mathrm{Tr}\Big(D_\phi \bar{a}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi)\nabla_x^2 v^\theta(t, \hat{X}_t^{\pi^\phi})\Big) + D_\phi \tilde{q}(t, \hat{X}_t^{\pi^\phi}, \pi^\phi)\Bigg]\mathrm{d}t + O(h^2). \tag{A20}$$

The first term recovers, by the chain rule and Itô's formula, the policy-gradient integral kernel $\hat{f}$ in [22]:

$$\nabla_x v^\theta \cdot D_\phi \bar{\mu} + \tfrac{1}{2}\,\mathrm{Tr}\Big(D_\phi \bar{a}\,\nabla_x^2 v^\theta\Big) = \mathbb{E}_{a\sim\pi^\phi}\Big[\big(\mathcal{L}^a v^\theta - \partial_t v^\theta\big) \tag{A21}$$

$$\times\,\nabla_\phi \log \pi^\phi\Big]. \tag{A22}$$

The second term is the contribution from the entropy-regularized reward:

$$\begin{aligned} D_\phi \widetilde{q} =& D_\phi \int_{\mathcal{A}} [f(t, \hat{X}_t^{\pi^\phi}, a) + \gamma p(\cdot)]\pi^\phi(a|t, \hat{X}_t^{\pi^\phi})\mathrm{d}a \\ =& \mathbb{E}_{a\sim\pi^\phi}\Big[(f(t, \hat{X}_t^{\pi^\phi}, a) + \gamma p(\cdot))\nabla_\phi \log \pi^\phi + \gamma q_\phi(t, \hat{X}_t^{\pi^\phi}, a; \phi)\Big]. \end{aligned}$$

The terminal-kernel condition in Appendix A.7 bounds the integral over $[T-h, T]$ by $C_T h$. The integrated derivative-level Euler remainder is also $O(h)$ after division by h. After rearranging, we therefore obtain

$$\frac{1}{h} D_\phi L_h = \mathfrak{g}(0, x_0; \phi) + O(h),$$

which proves the theorem.

Appendix A.6. Identification, Anchoring, and Local Adversarial Stability

Appendix A.6.1. Bounded-Test Identification

Let ν be the time–pilot-state law, and $\bar{r}_h(Z) = \mathbb{E}[r_h[w] \mid Z] \in L^1(\nu)$. For the full measurable scalar test class $|\rho| \le C_\rho$, conditional expectation gives

$$\sup_{|\rho|\le C_\rho} |\mathbb{E}[\rho r_h[w]]| = C_\rho \|\bar{r}_h\|_{L^1(\nu)}. \tag{A23}$$

Indeed, the upper bound follows from $|\rho \bar{r}_h| \le C_\rho |\bar{r}_h|$, and equality is attained by $\rho = C_\rho\,\mathrm{sgn}(\bar{r}_h)$. A vector test gives the same result by using one fixed unit output direction. Smooth bounded classes inherit the identification whenever they approximate the sign test in the corresponding weighted integral. A prescribed neural class enforces its indexed moments on the pilot support. Every pointwise Euclidean-bounded class also satisfies

$$0 \le D_h(w) \le C_\rho^2 \mathbb{E}|r_h[w]|^2. \tag{A24}$$

This follows from Cauchy–Schwarz. Exact compensated transitions identify the continuous-time value. For a grid transition kernel, the compatible discrete value has zero discrete conditional residual; evaluating a continuous-time value through approximate propagation records its propagation defect.

Appendix A.6.2. Anchoring and a Finite-Class Approximation Bound

Fix the current and successor state-time laws, the terminal values and admissible function versions on their supports. For the function-space statement, assume the residual is a well-defined operator on the stated $L^2_W(\nu)$ domain, with successor evaluations determined by those functions. If $q = \mathbb{E}[Y \mid Z]$ and Y is square-integrable, expansion of $w - Y = (w - q) + (q - Y)$ proves (46); the conditional cross-term is zero. For fixed η, the moment is affine in w, so its squared norm is convex. The supremum D_h is therefore convex in the

represented functions. On a convex admissible function domain, the positive anchor gives strong convexity in the $L^2_W(\nu)$ norm. If q belongs to that domain and $D_h(q) = 0$, then

$$\Psi_h(w) - \Psi_h(q) = aD_h(w) + \lambda \|w - q\|^2_{L^2_W(\nu)}. \tag{A25}$$

In particular, q is the unique compatible minimizer up to ν-almost-everywhere equality.

For any neural function class $\mathcal{W}_\Theta$, suppose the empirical objective and its output $\widehat{w}$ satisfy

$$\sup_{w \in \mathcal{W}_\Theta} |\widehat{\Psi}_h(w) - \Psi_h(w)| \le \delta_{\text{emp}}, \qquad \widehat{\Psi}_h(\widehat{w}) \le \inf_{w \in \mathcal{W}_\Theta} \widehat{\Psi}_h(w) + \epsilon_{\text{opt}}.$$

For every comparator $\bar{w} \in \mathcal{W}_\Theta$, uniform comparison yields $\Psi_h(\widehat{w}) \le \Psi_h(\bar{w}) + \epsilon_{\text{opt}} + 2\delta_{\text{emp}}$. Inserting the anchor decomposition and dropping $aD_h(\widehat{w}) \ge 0$ gives

$$\|\widehat{w} - q\|^2_{L^2_W(\nu)} \le \|\bar{w} - q\|^2_{L^2_W(\nu)} + \frac{a}{\lambda} D_h(\bar{w}) + \frac{\epsilon_{\text{opt}} + 2\delta_{\text{emp}}}{\lambda}. \tag{A26}$$

This separates function approximation, weak-moment compatibility, empirical error, and optimization accuracy, and applies to a nonconvex neural class. In a summarized representation, Z specifies the conditioning information used by its regression target; the comparator term retains the corresponding approximation and compatibility contributions.

Proof of Theorem 7. Write $m(\theta, \eta) = m_h(w_\theta, \eta)$. At the common zero,

$$\nabla^2_{\theta\theta} F(\theta_*, \eta) = 2a(D_\theta m)^\top D_\theta m + \lambda \nabla^2 A(\theta_*) \succeq \lambda a_0 I. \tag{A27}$$

Continuity and compactness of K give $r > 0$ and uniform constants $0 < \mu \le L$ such that $\mu I \preceq \nabla^2_{\theta\theta} F(\theta, \eta) \preceq LI$ for $\|\theta - \theta_*\| \le r$ and $\eta \in K$. Since $\nabla_\theta F(\theta_*, \eta) = 0$, integration of the Hessian along the parameter segment gives

$$\theta - \alpha \nabla_\theta F(\theta, \eta) - \theta_* = \left[I - \alpha \int_0^1 \nabla^2_{\theta\theta} F(\theta_* + u(\theta - \theta_*), \eta) \mathrm{d}u \right] (\theta - \theta_*).$$

For $0 < \alpha \le L^{-1}$, the bracket has operator norm at most $\kappa = 1 - \alpha\mu < 1$, uniformly in $\eta \in K$.

The identity $m(\theta_*, \eta) = 0$ on K implies $D_\eta m(\theta_*, \eta) = 0$. Bounded first and mixed derivatives yield $\|m(\theta, \eta)\| \le C_1 \|\theta - \theta_*\|$ and $\|D_\eta m(\theta, \eta)\| \le C_2 \|\theta - \theta_*\|$. Hence,

$$\|\nabla_\eta F(\theta, \eta)\| \le 2aC_1C_2 \|\theta - \theta_*\|^2 =: C\|\theta - \theta_*\|^2. \tag{A28}$$

For $0 < \kappa < 1$, the total Test movement is bounded by

$$\sum_{k \ge 0} \|\eta_{k+1} - \eta_k\| \le \frac{\bar{\tau} C \kappa^2}{1 - \kappa^2} \|\theta_0 - \theta_*\|^2. \tag{A29}$$

Choose the initial Critic radius within r so the right-hand side is smaller than $\operatorname{dist}(\eta_0, \partial K)$. Induction then keeps every Test iterate inside K and every Critic iterate inside its contraction ball. The geometric estimate proves Critic convergence, and summability proves Test convergence. At θ_*, $F(\theta_*, \eta) = \lambda A(\theta_*)$ is constant in η; local Critic curvature gives $F(\theta_*, \eta) \le F(\theta_*, \eta_\infty) \le F(\theta, \eta_\infty)$. These are the local saddle inequalities. If $\kappa = 0$, one Critic step reaches θ_* and subsequent Test gradients are zero.

For a representable regression target $w_{\theta_*} = q$, differentiating (46) gives $\nabla^2 A(\theta_*) = 2\mathbb{E}[J_*^\top W J_*]$. Thus, local curvature is the positive Gram condition for the locally identifiable represented functions. This proof allows a globally nonlinear neural parameterization. □

Appendix A.6.3. Tracking a Changing Policy

Let F_k have target θ_k^* and satisfy the common-zero, stationary-anchor, and local derivative conditions uniformly. Denote $r_k = \|\theta_k - \theta_k^*\|$ and $\nu_k = \|\theta_{k+1}^* - \theta_k^*\|$. If a Critic step uses gradient perturbation ζ_k with $\|\zeta_k\| \le \epsilon_k$, contraction and the triangle inequality give

$$\begin{aligned} r_{k+1} &\le \kappa r_k + \alpha\epsilon_k + \nu_k, \\ r_k &\le \kappa^k r_0 + \sum_{j<k} \kappa^{k-1-j}(\alpha\epsilon_j + \nu_j). \end{aligned} \tag{A30}$$

For a finite horizon of K_0 rounds, let R_k denote the displayed upper bound and $u_k = \kappa R_k + \alpha\epsilon_k$, the distance after the Critic step from the round-k target. A sufficient neighborhood condition is $R_k \le r$ and $u_k \le r$ for every round, together with

$$\sum_{k<K_0} \tau_k(Cu_k^2 + \epsilon_{\eta,k}) < \operatorname{dist}(\eta_0, \partial K), \tag{A31}$$

where $\epsilon_{\eta,k}$ bounds the Test-gradient perturbation. The same induction proves (A30). Uniform bounds on ϵ_k and ν_k give the Critic tracking radius $(\alpha\bar{\epsilon} + \bar{\nu})/(1-\kappa)$; (A31) specifies the associated Test margin.

Appendix A.7. Critic Derivatives, Euler Consistency, and Policy Improvement

Proof of Proposition 1. The reward and current-value terms cancel when subtracting the two residual derivatives. Conditional on the pilot state x, the coordinate-j difference divided by h is

$$\mathbb{E}_\varepsilon\left[\nabla_x e(t+h, Y_h^\phi)^\top \left(\partial_{\phi_j} b_\phi + h^{-1/2}\partial_{\phi_j} S_\phi \varepsilon\right)\right]. \tag{A32}$$

For a Gaussian coordinate ε_l, integration by parts gives

$$\mathbb{E}_\varepsilon[\partial_{x_i} e(t+h, Y_h^\phi)\varepsilon_l] = \sqrt{h}\,\mathbb{E}_\varepsilon\left[\sum_p \partial_{x_i x_p} e(t+h, Y_h^\phi)(S_\phi)_{pl}\right].$$

Consequently, the diffusion part of (A32) equals $\frac{1}{2}(\partial_{\phi_j} C_\phi) : \mathbb{E}_\varepsilon[\nabla_x^2 e(t+h, Y_h^\phi)]$, using symmetry of the Hessian and $\partial_{\phi_j} C_\phi = (\partial_{\phi_j} S_\phi) S_\phi^\top + S_\phi(\partial_{\phi_j} S_\phi)^\top$. Integrating against the discounted pilot law proves (51). Operator duality and Cauchy–Schwarz give

$$\|g_h[\widehat{v}] - g_h[v]\| \le B_{1,h} E_{1,h} + \tfrac{1}{2} B_{2,h} E_{2,h}. \tag{A33}$$

Combining this with Theorem 6 gives (52). □

Appendix A.7.1. A Derivative-Level Euler Estimate and the Terminal Interval

The consistency assumptions used in Theorem 6 can be stated directly. For $U = \nabla_x v$ and $U = \nabla_x^2 v$, require

$$\|\mathbb{E}_\varepsilon U(t+h, Y_h^\phi) - U(t,x)\| \le h C_U(t,x), \tag{A34}$$

with coefficient-weighted integrable envelopes uniform in h. One sufficient set of conditions consists of spatial derivatives of v through order four, $\partial_t \nabla_x v$ and $\partial_t \nabla_x^2 v$, together with growth and moment bounds controlling their Taylor remainders along the propagated states. Indeed, the mean displacement is $h b_\phi$ and its second moment is $h\,\mathrm{Tr}(C_\phi) + h^2\|b_\phi\|^2$; time increments and second-order spatial Taylor formulas give (A34) with these envelopes.

Define the exact gradient kernel

$$K_\phi[v] = (D_\phi b_\phi)^\top \nabla_x v + \tfrac{1}{2}(D_\phi C_\phi)^* \nabla_x^2 v + D_\phi \widetilde{q}_\phi. \tag{A35}$$

Assume $\int_{T-h}^{T} e^{-\beta t}\mathbb{E}\|K_\phi[v](t, X_t^{\pi^\phi})\|\mathrm{d}t \le C_T h$. The differentiated-value PDE, or the equivalent policy-gradient representation, gives $\nabla J(\phi) = \int_0^T e^{-\beta t}\mathbb{E}K_\phi[v](t, X_t^{\pi^\phi})\mathrm{d}t$. Applying the Gaussian identity to the exact-Critic residual first, and then using (A34), bounds its kernel discrepancy on $[0, T-h]$ by an integrable multiple of h. The terminal estimate gives the remaining interval. Therefore,

$$\|g_h[v] - \nabla J(\phi)\| \le C_E h. \tag{A36}$$

For a payoff with nonsmooth terminal data, the displayed interior-derivative and terminal-integral conditions specify the regularity needed for this rate. They can be checked at the expected-kernel level rather than by requiring uniformly bounded pointwise terminal Greeks.

Appendix A.7.2. From Critic Contraction to Gradient Accuracy

Fix the policy and h. Suppose the parameter derivatives of $D_x^j v_\theta$, $j = 1, 2$, as maps to $L^2(\mu_h)$ at propagated states, have operator norm bounded by M_j on the convex local Critic ball. For a reference θ_* put $E_{j,*} = \|D_x^j(v_{\theta_*} - v)\|_{L^2(\mu_h)}$. Parameter-line integration gives $E_{j,h}(\theta_k) \le E_{j,*} + M_j\|\theta_k - \theta_*\|$. Thus, Theorem 7 implies

$$\begin{aligned}\|g_h[v_{\theta_k}] - \nabla J(\phi)\| \le{}& C_E h + B_{1,h}E_{1,*} + \tfrac{1}{2}B_{2,h}E_{2,*} \\ &+ (B_{1,h}M_1 + \tfrac{1}{2}B_{2,h}M_2)\kappa^k\|\theta_0 - \theta_*\|.\end{aligned} \tag{A37}$$

The reference terms distinguish the compatible represented Critic from the continuous-time value. A moving policy uses the tracking bound (A30) in place of the geometric term.

Appendix A.7.3. Pilot-Law Comparison and Monte Carlo Precision

Let $\widehat{\nu}$ denote the occupation law used by a numerical or empirical pilot. Define $\epsilon_{P,h}$ as the norm of the difference between the integrated finite-step gradient kernels under $\widehat{\nu}$ and the on-policy law, using the same fitted Critic in both kernels. The triangle inequality adds $\epsilon_{P,h}$ to (52). Conditional on the fitted Critic and pilot specification, an independent N-sample gradient estimate has variance σ_h^2/N. The pathwise diffusion derivative in (A32) carries $h^{-1/2}$; uniform second-moment envelopes give $\sigma_h^2 \le C_0 + C_1/h$. Its root-mean-square total error is consequently at most

$$C_E h + B_{1,h}E_{1,h} + \tfrac{1}{2}B_{2,h}E_{2,h} + \epsilon_{P,h} + \sqrt{(C_0 + C_1/h)/N}. \tag{A38}$$

This expression states how step size, derivative accuracy, and sample size jointly determine the residual gradient.

Proof of Corollary 1. Write $g_k = \nabla J(\phi_k)$. The smoothness inequality and conditional centering of the noise imply

$$\mathbb{E}[J(\phi_{k+1}) \mid \mathcal{F}_k] \ge J(\phi_k) + \alpha_\phi g_k^\top (g_k + b_k) - \tfrac{1}{2}L_J\alpha_\phi^2(\|g_k + b_k\|^2 + s_k^2).$$

Use $L_J \alpha_\phi \le 1$ and $g^\top(g+b) - \frac{1}{2}\|g+b\|^2 = \frac{1}{2}\|g\|^2 - \frac{1}{2}\|b\|^2$, sum over k, and apply $J(\phi_K) \le J_{\max}$. Rearranging gives (53). In addition, a deterministic direction d is an ascent direction whenever $\|d - \nabla J\| < \|\nabla J\|$, since $\langle \nabla J, d \rangle \ge \|\nabla J\|(\|\nabla J\| - \|d - \nabla J\|) > 0$. □

Appendix A.7.4. Differentiable Observed Transitions

Let $F_\psi(z, \xi)$ be a differentiable transition with a policy-independent base random variable, and fix the pilot law and Critic. For a block duration Δ, define

$$\mathcal{G}_\Delta[w] = \Delta^{-1} \nabla_\psi \mathbb{E}[w(F_\psi(z,\xi)) - w(z) + r_\psi(z,\xi)]\Big|_{\psi=\phi}.$$

Dominated differentiation gives the exact counterpart

$$\begin{aligned} \mathcal{G}_\Delta[\widehat{v}] - \mathcal{G}_\Delta[v] &= \Delta^{-1}\mathbb{E}[(D_\phi F_\phi)^\top \nabla e(F_\phi)], \\ \|\mathcal{G}_\Delta[\widehat{v}] - \mathcal{G}_\Delta[v]\| &\le \Delta^{-1}\|D_\phi F_\phi\|_{L^2(\mathrm{op})}\|\nabla e(F_\phi)\|_{L^2}. \end{aligned} \tag{A39}$$

The portfolio implementation uses $W' = W a_\psi(z,\xi)^\top G$, where G contains observed 21-day asset gross factors and a_ψ is a differentiable Dirichlet sample. The derivative includes both wealth and the next log-wealth feature; historical return features remain fixed. The block reward includes

$$\Delta \frac{\gamma}{d}\{H(\mathrm{Dir}(\alpha_\psi)) - H(\mathrm{Dir}(\mathbf{1}))\} = -\Delta \frac{\gamma}{d}\,\mathrm{KL}(\mathrm{Dir}(\alpha_\psi)\|\,\mathrm{Unif}). \tag{A40}$$

Its full policy derivative enters r_ψ. Summing exact-value block derivatives against on-policy occupancies gives the finite-horizon Markov policy gradient. Under historical replay, (A39) quantifies the derivative of the specified replay residual objective. Thus, trajectory PE uses sampled states and realized rewards, while the pathwise Actor uses the differentiable allocation-to-wealth map supplied by observed return blocks.

Appendix A.8. Cross-Batch Estimation of Squared Weak Moments

For parameters fixed before sampling, let $U = \rho_\eta(Z) r_h[w](Z, Z')$ have mean m and covariance Σ. Let $\widehat{m}_1, \widehat{m}_2$ be means of independent, identically distributed batches of size B. Writing $\widehat{m}_i = m + e_i$ and using independence gives

$$\begin{aligned} \mathbb{E}[\widehat{m}_1^\top \widehat{m}_2] &= \|m\|^2, & \mathbb{E}\|\widehat{m}_1\|^2 &= \|m\|^2 + \mathrm{Tr}(\Sigma)/B, \\ \mathrm{Var}(\widehat{m}_1^\top \widehat{m}_2) &= 2m^\top \Sigma m/B + \mathrm{Tr}(\Sigma^2)/B^2. \end{aligned} \tag{A41}$$

The cross-product removes the single-batch self-product term. Its realized value and the negative Test minimizer loss can have either sign. Dominated differentiation gives the same fresh-batch identity for its gradient.

In the option implementation, the local batches use independent index and noise draws conditional on their common pilot library D; for fixed pre-sampling parameters, their cross-product targets $\|m_D\|^2$. In the portfolio implementation, episodes are split into two groups before within-group resampling. Conditional on the two rollout buffers, the corresponding target is $m_{D_1}^\top m_{D_2}/14$. The Test update reuses the sampled data after a Critic step and is an adaptive empirical update. Pool refresh, episode construction, and inner-loop reuse specify this conditional empirical objective in the numerical protocol. The empirical-error term in (A26) accommodates the comparison with its population counterpart.

Appendix A.9. Proof of Proposition 2

The sample covariance is

$$\widehat{\Sigma} = \frac{1}{n-1}\sum_{j=1}^{n}(\mathbf{r}_j - \bar{\mathbf{r}})(\mathbf{r}_j - \bar{\mathbf{r}})^\top.$$

Its formation costs $\mathcal{O}(nd^2)$ and stores $\mathcal{O}(d^2)$ entries. A dense Cholesky factorization, where used to solve the covariance system, costs $\mathcal{O}(d^3)$ [38]. Further constrained-optimization work depends on the specified solver and stopping criterion.

In the reported neural implementation, the policy scores each asset using a shared fixed-width network. Its input contains six common context features and three features for that asset, giving constant work per score and $\mathcal{O}(d)$ work for all scores. Softmax normalization and construction of the Dirichlet parameters also take $\mathcal{O}(d)$ work. Dirichlet sampling has the stipulated linear expected cost, while log-density and entropy require linear sums of scalar terms. The weighted-return wealth map costs $\mathcal{O}(d)$ per transition.

The value and test networks first aggregate the asset features by means and standard deviations, using $\mathcal{O}(d)$ work, and then evaluate fixed-width networks on their fixed-size inputs. Reverse-mode differentiation has the same order as these forward operations. With batch size, episode length, and the numbers of inner updates fixed, the full learning iteration therefore has expected cost $\mathcal{O}(d)$. Parameter sharing makes the network parameter counts independent of the asset count; asset inputs, samples, and allocations grow linearly with d.

These operation counts describe learning with the prepared return blocks. Market simulation, covariance construction for the simulator, history preparation, and evaluation have their own costs, recorded separately in the timing protocols. The scale study in Table A8 measures fixed-workload training; Table 4 measures time to common performance targets. The shared structure is also used by the implemented neural baselines, whose relative learning efficiency is evaluated by the latter experiment.

Appendix B. Algorithmic Details

We use the grid $t_i = i\Delta t$, $i = 0, \ldots, K$, with $K\Delta t = T$. The algorithms below are ordered by their first use in the main text. For the generic weak-martingale algorithms, two independent mini-batches form the cross-moment, and trajectory targets supply the positive anchor in (45). The fresh-batch identities and application-specific buffer conditioning are given in Appendix A.8. This implements the weak-moment objectives in (24) and (37).

For fixed-policy evaluation, the terminal condition can be imposed by writing

$$v^\theta(t,x) = g(x) + (T-t)u^\theta(t,x), \tag{A42}$$

where u^θ is an unconstrained neural network. A bounded test network for the generic fixed-policy construction is

$$\rho_\eta(t,x) = \sin\big(\Lambda(W_1 t + W_2 x) + b\big) \in \mathbb{R}^r, \tag{A43}$$

where $W_1 \in \mathbb{R}^r$, $W_2 \in \mathbb{R}^{r\times d}$, $b \in \mathbb{R}^r$, and $\eta = (W_1, W_2, b)$. Moreover, $\Lambda(y_1, \ldots, y_r) = (c_{\text{test}}y_1, 2c_{\text{test}}y_2, \ldots, rc_{\text{test}}y_r)^\top$ for a fixed $c_{\text{test}} > 0$. Other bounded parameterizations of $\mathcal{T}_b$ can be used in the same way.

Appendix B.1. Martingale-Adversarial Policy Evaluation

Algorithm A1 Weak-martingale adversarial policy evaluation

Input: step size h, batch size B, numbers N_θ, N_η of Critic and test-function updates, and learning rates $\alpha_\theta, \alpha_\eta$; trajectory targets and positive weights a, λ for (45).
Initialize: v^θ using (A42) and the bounded test network ρ_η in (A43).
for each outer iteration **do**
  for $j = 1$ to N_θ **do**
    Draw independent mini-batches $\mathcal{B}_1, \mathcal{B}_2$ of B tuples $(t, \hat{X}_t, \varepsilon)$, where t is uniform on $[0, T-h]$, $\hat{X}_t$ is sampled from the pilot process, and $\varepsilon \sim N(0, I_m)$.
    for $k = 1, 2$ **do**
      Compute
$$\widehat{G}_h^{(k)}(\theta, \eta) = \frac{T-h}{B} \sum_{(t,x,\varepsilon)\in\mathcal{B}_k} \rho_\eta(t,x)\, \widehat{\Delta M}_{t,h}^{\theta}(x,\varepsilon),$$
    end for
    Form $\widehat{A}$ from sampled trajectory targets, set $\widehat{\mathcal{L}}_{\text{weak}} = (\widehat{G}_h^{(1)})^\top \widehat{G}_h^{(2)}$ and $\widehat{\mathcal{L}}_{\text{C}} = a\widehat{\mathcal{L}}_{\text{weak}} + \lambda \widehat{A}$. Update $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \widehat{\mathcal{L}}_{\text{C}}$.
  end for
  for $j = 1$ to N_η **do**
    Draw fresh independent mini-batches $\mathcal{B}_1, \mathcal{B}_2$ and form $\widehat{\mathcal{L}}_{\text{C}}$ as above.
    Update $\eta \leftarrow \eta + \alpha_\eta \nabla_\eta \widehat{\mathcal{L}}_{\text{C}}$, holding θ fixed.
  end for
end for
Output: the value approximation v^θ.

Appendix B.2. Martingale-Residual Actor–Critic

Algorithm A2 Alternating martingale-residual Actor–Critic

Input: value network v^θ, randomized policy distribution $\pi^\phi(\cdot \mid t, x)$, test network ρ_η, local step h, batch size B, Critic-update count N_{C}, and learning rates $\alpha_\theta, \alpha_\eta, \alpha_\phi$; trajectory targets and positive a, λ.
for outer iteration $n = 1, 2, \ldots$ **do**
  for $j = 1$ to N_{C} **do**
    Draw independent relaxed-policy mini-batches $\mathcal{B}_1, \mathcal{B}_2$ of $(t, \hat{X}_t^{\pi^\phi}, \varepsilon)$ under the current policy, with t uniform on $[0, T-h]$.
    for $k = 1, 2$ **do**
      Compute
$$\widehat{G}_h^{(k)}(\theta, \phi, \eta) = \frac{T-h}{B} \sum_{(t,x,\varepsilon)\in\mathcal{B}_k} \rho_\eta(t,x)\, \mathcal{R}_h^\theta(t,x,\varepsilon;\pi^\phi),$$
      where $\mathcal{R}_h^\theta$ is defined in (30).
    end for
    Set $\widehat{\mathcal{L}}_{\text{weak}} = (\widehat{G}_h^{(1)})^\top \widehat{G}_h^{(2)}$ and $\widehat{\mathcal{L}}_{\text{C}} = a\widehat{\mathcal{L}}_{\text{weak}} + \lambda \widehat{A}$, with sampled trajectory anchor $\widehat{A}$. Update $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \widehat{\mathcal{L}}_{\text{C}}$, holding ϕ, η fixed.
    Recompute the weak moment at the updated Critic and update $\eta \leftarrow \eta + \alpha_\eta a \nabla_\eta \widehat{\mathcal{L}}_{\text{weak}}$, holding θ, ϕ fixed. The application protocol specifies reuse or refresh of the sampled data.
  end for
  Draw a fresh relaxed-policy mini-batch $\mathcal{B}_{\text{A}}$ of $(t, \hat{X}_t^{\pi^\phi}, \varepsilon)$, with t uniform on $[0, T-h]$.
  Introduce a copy ψ of the policy parameter. With θ and the sampled pilot points fixed, compute
$$\widehat{\mathfrak{g}}_\phi = \frac{T-h}{h|\mathcal{B}_{\text{A}}|} \sum_{(t,x,\varepsilon)\in\mathcal{B}_{\text{A}}} e^{-\beta t} \nabla_\psi \mathcal{R}_h^\theta(t,x,\varepsilon;\pi^\psi)\Big|_{\psi=\phi}.$$
  Update $\phi \leftarrow \phi + \alpha_\phi \widehat{\mathfrak{g}}_\phi$.
end for
Output: (v^θ, π^ϕ).

Appendix B.3. Offline Martingale-Loss Policy Evaluation

Algorithm A3 Offline policy evaluation by martingale loss

Input: value network v^θ, complete trajectories $\tau^b = \{X^b_{t_i}\}_{i=0}^K$, running reward f, terminal reward g, batch size B, and learning rate α_θ.
for each optimization step **do**
  Sample B complete trajectories under the fixed policy.
  for each trajectory b and each $i = 0, \dots, K-1$ **do**
    Form the realized return from t_i to T,
    $$Y_i^b = g(X^b_{t_K}) + \sum_{j=i}^{K-1} f(t_j, X^b_{t_j})\Delta t,$$
    and the pathwise martingale-loss contribution $\ell_i^b = |Y_i^b - v^\theta(t_i, X^b_{t_i})|^2 \Delta t$.
  end for
  Set $\mathcal{L}_{\mathrm{ML}} = (2B)^{-1} \sum_{b=1}^B \sum_{i=0}^{K-1} \ell_i^b$ and update $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \mathcal{L}_{\mathrm{ML}}$.
end for
Output: the martingale-loss value approximation v^θ.

Appendix B.4. Continuous-Time TD(0) by Martingale Orthogonality

Algorithm A4 CTD(0) with a prescribed test function

Input: value network v^θ, learning rate α_θ, fixed policy, and grid $t_0, \dots, t_K$.
for each episode **do**
  Generate a trajectory $\{X_{t_i}\}_{i=0}^K$ and record the running rewards $f(t_i, X_{t_i})$.
  for $i = 0$ to $K-2$ **do**
    Set $\xi_i = \nabla_\theta v^\theta(t_i, X_{t_i})$ and
    $\delta_i^\theta = v^\theta(t_{i+1}, X_{t_{i+1}}) - v^\theta(t_i, X_{t_i}) + f(t_i, X_{t_i})\Delta t$.
    Update $\theta \leftarrow \theta + \alpha_\theta\, \xi_i \delta_i^\theta$.
  end for
  Set $\xi_{K-1} = \nabla_\theta v^\theta(t_{K-1}, X_{t_{K-1}})$ and
  $\delta_{K-1}^\theta = g(X_{t_K}) - v^\theta(t_{K-1}, X_{t_{K-1}}) + f(t_{K-1}, X_{t_{K-1}})\Delta t$.
  Update $\theta \leftarrow \theta + \alpha_\theta\, \xi_{K-1} \delta_{K-1}^\theta$.
end for
Output: the CTD(0) value approximation v^θ.

Appendix B.5. Basket-Option Instance

Algorithm A5 Implemented weak-martingale ADV evaluation for the basket option

Input: parameters in Section 5.1, 5000 outer iterations, full-path and local batch sizes 128.
Initialize the value representation (A44), bounded Test, optimizers, and EMA copies as in Table A9.
for each outer iteration **do**
  Generate 128 exact-GBM full paths on the 50-step grid and draw independent local successors conditional on sampled pilot states.
  for four value updates **do**
    Draw two local batches and form $\widehat{G}_k = |\mathcal{B}_k|^{-1} \sum_{\mathcal{B}_k} \rho_\eta(t, X)[e^{-rh} v^\theta(t+h, X') - v^\theta(t, X)]/h$.
    Form the discounted terminal-payoff anchor $\widehat{A}_{\mathrm{path}}$ on 16 selected full paths and all nonterminal times.
    With the Test fixed, descend in θ on $0.1\widehat{G}_1^\top \widehat{G}_2 + 0.3\widehat{A}_{\mathrm{path}}$, clip the gradient at five, and update the EMA copy.
  end for
  for two Test updates **do**
    Draw fresh local batches; with value parameters fixed, descend in η on $-\widehat{G}_1^\top \widehat{G}_2$ and clip the gradient at five.
  end for
  Apply the stated learning-rate schedule and record held-out raw and EMA diagnostics at every 100 outer iterations.
end for
Output: fitted value network, EMA parameters, and complete training and evaluation records.

Appendix B.6. Mean–Variance Portfolio Selection: Offline Version

Algorithm A6 Implemented offline martingale-residual AC for the MV problem

Input: observed training returns, causal features, $z = 1.15$, $\gamma = 0.01$, duration $\Delta = 21/252$, and Table A10.
Materialize all admissible 252-day historical episodes before the training-history endpoint; store 21-day asset gross factors and causal observations.
Initialize the shared Dirichlet Actor, physical-moment value (A45), bounded Test, and $w = z$.
for 900 outer iterations **do**
    Sample 32 episodes under the current randomized policy; propagate wealth by $X' = Xa^\top G$ and record rewards and terminal outcomes.
    for four Critic/Test updates **do**
        Split the sampled episodes into independent groups and draw transition mini-batches within each group.
        Form residual rates from (82) and their Test-weighted empirical means $\widehat{G}_1, \widehat{G}_2$.
        With the Test fixed, descend on (83), using the moment anchor (A46).
        With the Critic fixed, descend on $-\widehat{G}_1^\top \widehat{G}_2/14$ in the Test parameters.
    end for
    if the 50-iteration Actor warmup is complete **then**
        Fix sampled starting observations and the Critic; draw reparameterized Dirichlet actions and form $X' = Xa^\top G$ using observed block factors.
        Ascend in the Actor parameters on the mini-batch mean of $[v_w^\theta(s', X') - v_w^\theta(s, X) + \gamma\Delta\overline{H}(\pi)]/\Delta$.
    end if
    Every five outer iterations, update w by (84); clip all network gradients at five.
end for
Output: pretrained $(\pi^\phi, v_w^\theta, w)$; execute the conditional-mean allocation with the common self-financing fee rule.

Appendix B.7. Mean–Variance Portfolio Selection: Online Version

Algorithm A7 Causal online adaptation of the MV Actor–Critic

Input: the common pretraining history, evaluation returns revealed sequentially, and the offline configuration.
Pretrain using Algorithm A6; retain the pretraining episode distribution as an anchor.
Scale network learning rates and multiplier rate by 0.1.
for each evaluation holding period **do**
    Construct the observation using returns strictly before the decision and execute the Actor's conditional-mean allocation.
    Charge the common proportional transaction cost and hold the allocation as shares for up to 21 trading days.
    Observe the completed block and extend the available-history endpoint.
    Materialize one-year episodes within the most recent 1260 observed days; form a 50:50 mixture with the fixed pretraining history.
    for four adaptation outer iterations **do**
        Draw each episode group independently from the same causal mixture.
        Apply the Critic, Test, Actor, and scheduled multiplier updates from Algorithm A6, using only the observed history.
    end for
end for
Output: executed wealth path, final policy, and recorded adaptation objectives.

Appendix C. Supplementary Numerical Evidence and Implementation

This appendix provides the implementation details, paired comparisons, sensitivity analyses, and timing measurements for Section 5. All summaries include the complete fixed seed sets for their respective protocols.

Appendix C.1. Basket-Option References and Gamma Landscapes

For equicorrelated geometric Brownian motion, conditioning on the idiosyncratic Gaussian factors leaves a one-factor lognormal call whose conditional price, Delta, and Γ_{11} can be integrated analytically. Antithetic sampling of the idiosyncratic factors gives the numerical references. The initial reference uses 262,144 pairs, each held-out path-state reference uses 4096 pairs, and the Greek grid uses 32,768 pairs. The final evaluation contains 64 independent paths on a 100-step fine grid, subsampled to the learner grid; the separate monitoring stream contains 16 paths with 1024 reference pairs per state.

The reference routine reproduces the Black–Scholes limit to a maximum absolute error of 2.22×10^{-16}. Common-random-number price finite differences at three price levels and bumps $0.005, 0.01, 0.02$ agree with the analytic conditional Gamma within 1.74×10^{-11}. Increasing the antithetic sample from 4096 to 65,536 pairs reduces the reported Gamma standard errors, consistent with the pair-based Monte Carlo standard-error calculation. The root-mean-square Gamma reference standard error on the assessment grid is approximately 1.68×10^{-7}.

Figures A2 and A3 show the signed error in the conventional Γ_{11} across both price slices and all three evaluation times. Each panel uses all 20 fitted EMA models; the central curve and band give their median and interquartile range. Price levels label evaluation states rather than directions of differentiation. The full seed distribution in Figure A1 complements these pointwise landscapes. ADV has the lowest mean Gamma error on both price slices.

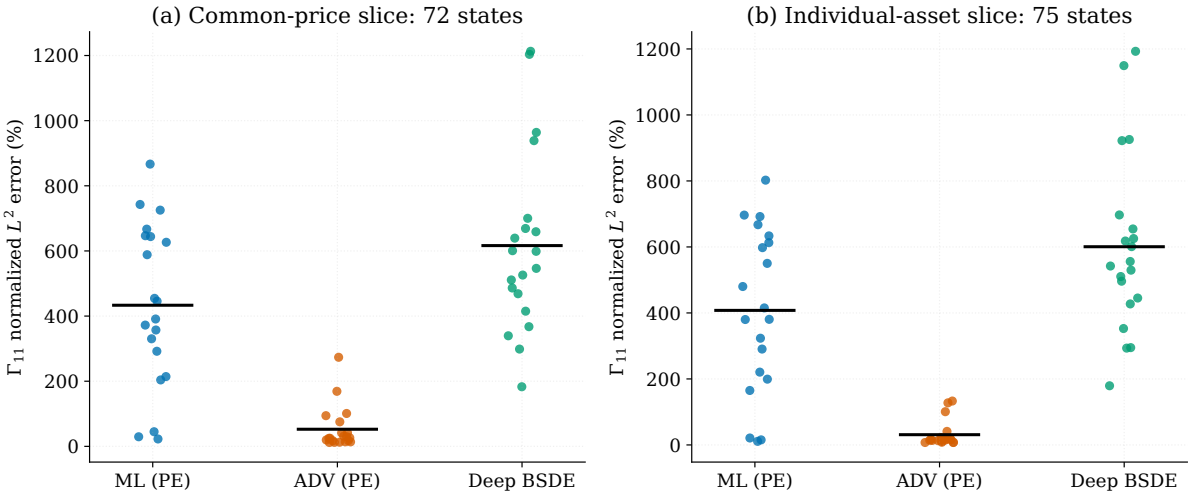


Figure A1. Distribution of normalized Γ_{11} errors over all 20 training seeds on the common-price and individual-asset slices. Every point represents one fitted model, and horizontal bars denote means. The full seed distribution is shown for each method.

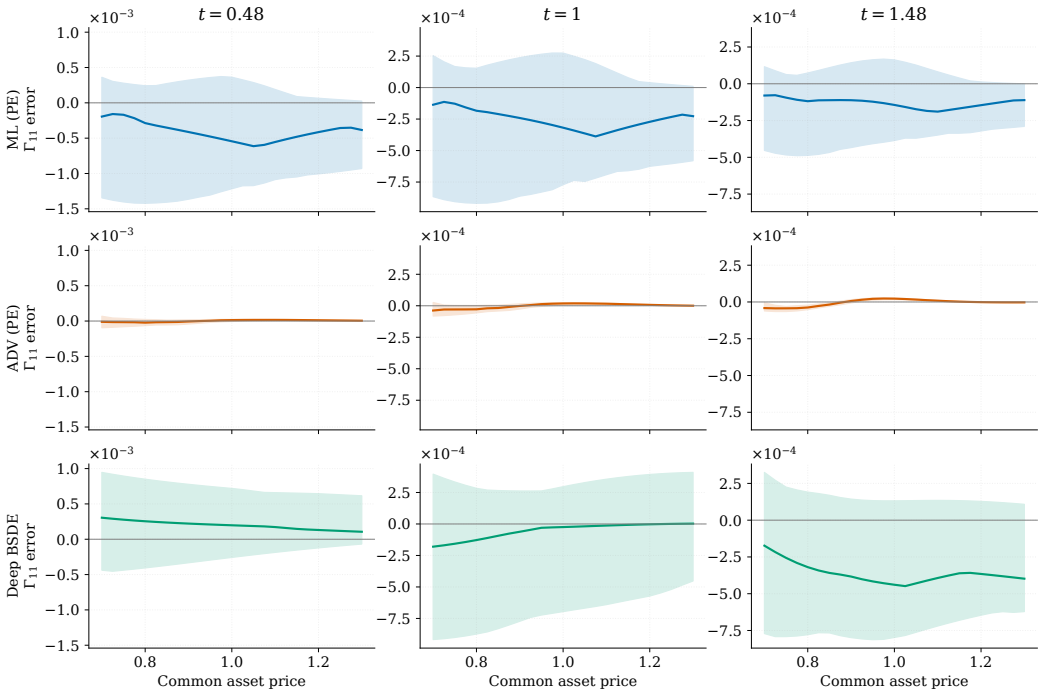


Figure A2. Signed Γ_{11} error on the common-price slice $x = s\mathbf{1}_d$ at three times. Rows correspond to the three methods; curves and bands are seed medians and interquartile ranges. Derivatives are with respect to x_1.

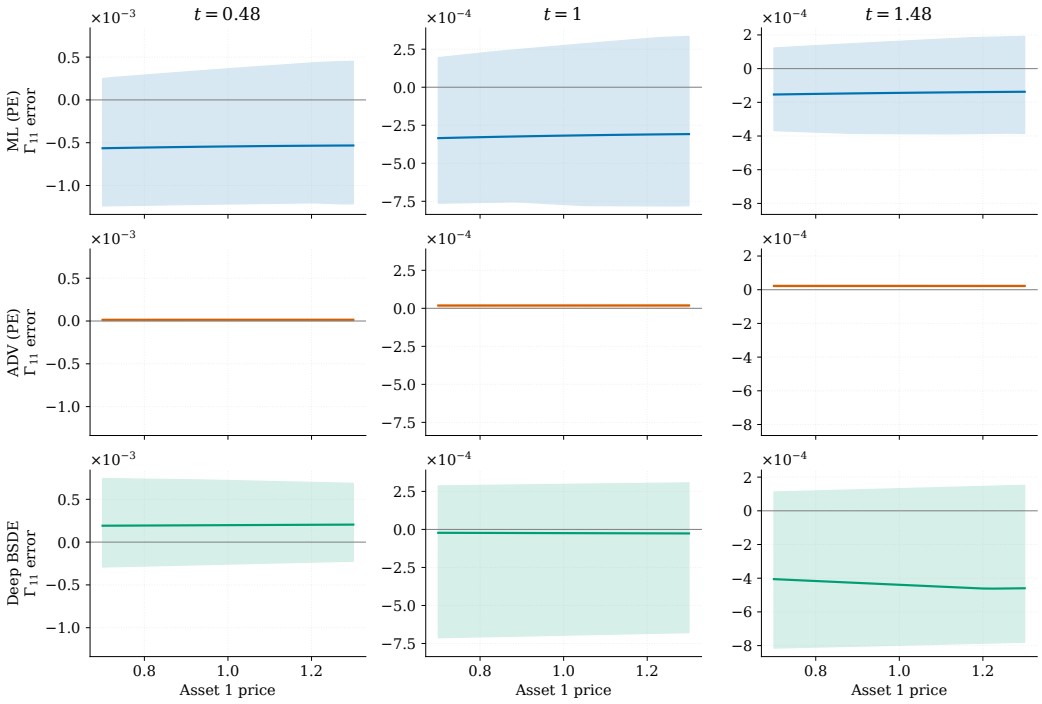


Figure A3. Signed Γ_{11} error on the individual-asset slice $x = (s, 1, \ldots, 1)$, with the same evaluation and plotting conventions as Figure A2.

Table A1. Conventional Γ_{11} errors across all 20 fitted EMA models.

Method/Slice	Normalized Error (%), Mean ± SD	Median (%)	Mean Absolute RMSE
ADV/common	52.6 ±65.6	25.5	7.17×10^{-5}
ADV/individual	31.1 ±39.6	15.3	4.27×10^{-5}
BSDE/common	616.4 ±278.0	572.7	8.41×10^{-4}
BSDE/individual	600.6 ±269.9	549.0	8.26×10^{-4}
ML/common	433.3 ±251.3	418.4	5.91×10^{-4}
ML/individual	407.8 ±247.4	397.9	5.61×10^{-4}

The normalized error is $100\|\widehat{\Gamma} - \Gamma_{\text{ref}}\|_2 / \|\Gamma_{\text{ref}}\|_2$. The common-price primary summary uses the 72 off-strike states retained in the protocol. Interior derivatives exist at the strike under the smooth representation. The individual-asset summary uses all 75 states.

Appendix C.2. Paired Option Comparisons

The independent repeated unit is the training seed. Each comparison uses one paired aggregate error per seed, conditional on the common held-out reference stream. Table A2 reports paired seedwise comparisons. These comparisons separate initial-price fitting, path-value approximation, Delta, and Gamma, in accordance with their different numerical objectives. Deep BSDE gives the lowest mean initial-price error, while both martingale methods improve path and Delta accuracy.

Table A2. Paired EMA error comparisons across 20 training seeds.

Candidate/Control	Metric	Mean Error Ratio	Wins	p_{Holm}
ML/BSDE	Path	0.168	20/20	7.63×10^{-6}
ML/BSDE	Delta	0.222	20/20	7.63×10^{-6}
ADV/BSDE	Path	0.287	20/20	7.63×10^{-6}
ADV/BSDE	Delta	0.0853	20/20	7.63×10^{-6}
ADV/ML	Path	1.71	0/20	3.81×10^{-6}
ADV/ML	Delta	0.385	19/20	4.01×10^{-5}
ML/BSDE	Initial price	16	2/20	8.05×10^{-4}
ADV/BSDE	Initial price	42.7	1/20	1.20×10^{-4}
ADV/ML	Initial price	2.67	4/20	0.0118
ML/BSDE	Gamma: common	0.703	14/20	0.2306
ML/BSDE	Gamma: individual	0.679	14/20	0.2306
ADV/BSDE	Gamma: common	0.0853	20/20	1.14×10^{-5}
ADV/BSDE	Gamma: individual	0.0517	20/20	1.14×10^{-5}
ADV/ML	Gamma: common	0.121	18/20	0.0012
ADV/ML	Gamma: individual	0.0762	20/20	1.14×10^{-5}

The mean error ratio is candidate/control; values below one indicate a lower candidate mean error. Wins count seeds with lower candidate error. Two-sided exact sign tests use one paired error per training seed. Holm correction is applied within four stated families: four baseline path/Delta comparisons, two within-martingale path/Delta comparisons, three initial-price comparisons, and six Gamma comparisons. These grouped EMA comparisons were added in the completed-run analysis.

Appendix C.3. PE Discretization and Test-Network Sensitivity

We vary one factor at a time over three fixed seeds: the bounded Test map, Test-network capacity, inner learning rates, or the Euler time grid. The main norm projection maps a raw output z to $z \min\{1, 3/(\|z\|_2 + 10^{-8})\}$. The sine and tanh variants use

$(3/\sqrt{128})\sin(z)$ and $(3/\sqrt{128})\tanh(z)$, respectively, and hence share the same vector-norm bound. These maps allow a comparison of bounded test representations at matched output scale. The sine representation is useful for nonmonotone test features; the numerical comparison shows similarly sized value and Delta errors for both alternatives.

For the Euler comparison, $n_t = 25, 50, 100$ gives $h = T/n_t$. A common 100-step Brownian grid is aggregated to couple the three simulations. Each training run retains the same 5000 outer iterations and 128 paths per iteration. Figure A4 displays the resulting changes in price, path, and Delta errors. Under this fixed optimization budget, the errors reflect both the discretization and the training of the resulting objective: component Delta RMSE means are 5.78×10^{-4}, 6.32×10^{-4}, and 6.82×10^{-4}. Reporting the entire grid distinguishes this numerical sensitivity from the small-step consistency analysis.

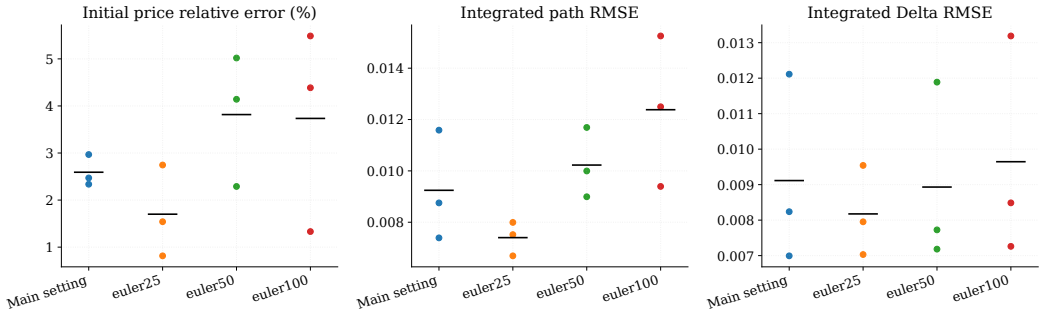


Figure A4. PE sensitivity to the true Euler grid, with coupled Brownian increments and the same outer training budget. Colored points show the three individual-seed values, and black horizontal bars denote means. Sample SDs are reported in Table A3. The integrated path and Delta units match the main comparison.

Table A3. ADV sensitivity over the same three fixed seeds: mean and sample SDs.

Setting	Initial Price Error (%)	Component Path RMSE (10^{-3})	Component Delta RMSE (10^{-3})
Sine bound	2.531 $\pm$0.381	6.540 $\pm$1.482	0.623 $\pm$0.143
Tanh bound	2.606 $\pm$0.322	6.585 $\pm$1.504	0.641 $\pm$0.188
Test $(128, 64)$	2.591 $\pm$0.334	6.538 $\pm$1.516	0.644 $\pm$0.187
Test $(512, 256)$	2.593 $\pm$0.337	6.539 $\pm$1.518	0.644 $\pm$0.188
Half inner rates	2.587 $\pm$0.405	6.187 $\pm$1.620	1.206 $\pm$0.666
Double inner rates	2.354 $\pm$0.620	7.159 $\pm$1.814	0.733 $\pm$0.297
Euler, 25 steps	1.700 $\pm$0.975	5.233 $\pm$0.466	0.578 $\pm$0.090
Euler, 50 steps	3.817 $\pm$1.393	7.231 $\pm$0.965	0.632 $\pm$0.182
Euler, 100 steps	3.735 $\pm$2.153	8.756 $\pm$2.073	0.682 $\pm$0.221

Here, component path RMSE divides the squared path integral by T; component Delta RMSE divides the squared vector integral by Td. Multiplication by $\sqrt{T}$ and $\sqrt{Td}$ recovers the main-table units. The reference 50-step exact-GBM ADV configuration on these three seeds has means 2.592%, 6.536×10^{-3}, and 0.644×10^{-3}, respectively. Both bounded sine and tanh variants satisfy the same test-vector-norm bound of three.

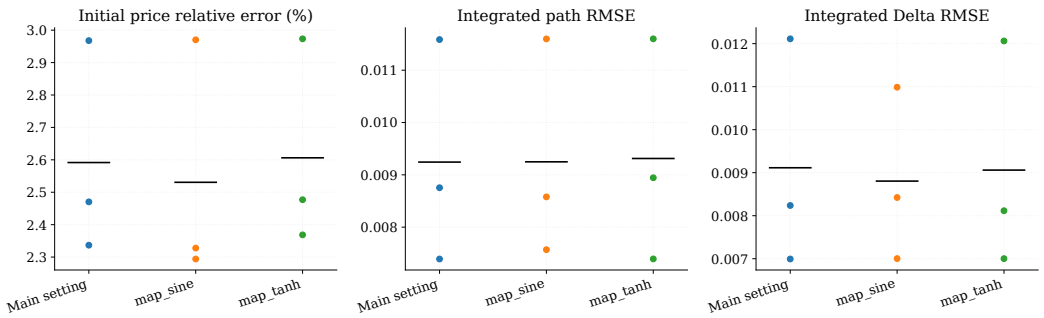


Figure A5. Matched-bound Test-map sensitivity for ADV: norm projection, sine, and tanh. Colored points show the same three individual-seed values, and black horizontal bars denote means. Sample SDs are reported in Table A3.

Appendix C.4. Measured Option-Training Cost

The controlled timing runs use the first three confirmation seeds, one Tesla T4 at a time, one CPU thread, and synchronized GPU timing. Table A4 separates full training cost from the time to reference-error targets. The target events use EMA predictions on the monitoring stream and three consecutive recorded evaluations. Full paths, local successor states, and optimizer update counts are logged separately. For 5000 outer iterations, the value-update counts are 5000 for ML and BSDE and 20,000 for ADV, with 10,000 Test updates for ADV. All three generate 640,000 full paths; ADV additionally samples 7.68×10^6 local successors for its value and Test updates.

ML has the shortest training and target times in this comparison. ADV combines a shorter full training time than Deep BSDE with substantially lower Delta error. Deep BSDE reaches the initial-price target earlier than ADV, while the joint path/Delta target emphasizes the function and derivative approximation used by the martingale framework.

Table A4. Serial option-training times in seconds on one Tesla T4 (three fixed seeds).

Method	Full Training Mean $\pm$ SD	Price Target Mean (Hits)	Joint Target Mean (Hits)
ML	20.49 ± 0.59	4.21 (3/3)	4.49 (3/3)
ADV	275.00 ± 26.50	71.82 (3/3)	71.82 (3/3)
BSDE	310.75 ± 35.94	37.62 (3/3)	$>$ budget (0/3)

Price target: relative initial-price error ≤ 0.05. Joint target: price ≤ 0.05, component path RMSE ≤ 0.03, and component Delta RMSE ≤ 0.005, equivalent to integrated thresholds 0.04243 and 0.07071. Three successive recorded evaluations are required. Full-training time includes simulation, optimization, scheduled monitoring, and gradient instrumentation. Target times are cumulative training-operation times at the confirming checkpoints, subtracting common monitoring time, and are averaged among hits. Construction, two-pass warmup, and reference generation precede the timed loop; final assessment and exports follow it. The complete 5000-iteration budget defines the censoring point.

Appendix C.5. Actual Critic and Test Training Records

Figure A6 reports raw objectives from dense reruns of the prespecified default seed 10001. The reruns reproduce the corresponding stored wealth paths. The Critic curve is its actual weak-moment-plus-anchor minimization loss. The Test curve is the minimization form of the negative cross-moment used in the maximizing update, so it can take either sign. The online boundary separates the 3600 pretraining Critic/Test updates from adaptation on completed holding periods. The plotted records are unsmoothed; the Critic uses a logarithmic scale and the signed Test uses a symmetric logarithmic scale. The two dense runs have finite recorded losses and gradients throughout.

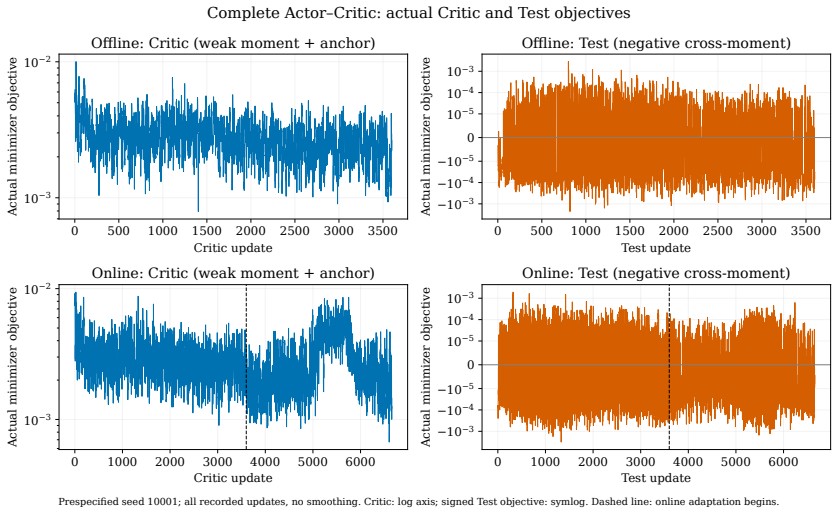


Figure A6. Actual Critic and Test objectives for the fixed diagnostic seed. The offline and online records are shown separately, with the online pretraining boundary marked. Curves use the recorded objectives without smoothing; the Test loss is the signed negative cross-moment.

Appendix C.6. AC Capacity, Learning-Rate, and Update-Frequency Sensitivity

Each configuration uses seeds 10001–10010 in both modes. The default ten-seed records are reused from the main experiment. Seven alternatives per mode contribute 140 new runs, and the two dense default diagnostic reruns contribute two further runs. Test hidden widths are 32, 64, and 128; Critic and Test rates are jointly scaled by $0.5, 1, 2$; their update counts per outer iteration are $1, 2, 4, 8$, with the Actor schedule fixed. Tables A5 and A6 give all six performance criteria.

Across these settings, offline mean annualized returns range from 16.302% to 16.416%, and online means range from 15.614% to 15.696%. The corresponding Sharpe ranges are 2.3782–2.3898 and 2.3370–2.3412. These changes show that the main empirical behavior persists across the tested factor ranges. The default four-step inner schedule retains the original main-experiment configuration. In the offline runs, the inner Critic/Test counts are 900, 1800, 3600, or 7200, while the Actor count remains 850. The online totals are 1664, 3328, 6656, or 13,312 inner updates and 1614 Actor updates.

Table A5. AC offline sensitivity: mean and sample SD over ten fixed seeds per setting.

Setting	Ann. Ret. (%)	Sharpe	Sortino	Calmar	MDD (%)	RT (Years)
Default	16.365 ±0.185	2.384 ±0.010	3.881 ±0.019	0.388 ±0.005	42.195 ±0.106	1.958 ±0.008
Width 32	16.416 ±0.210	2.389 ±0.019	3.890 ±0.035	0.389 ±0.005	42.185 ±0.082	1.958 ±0.006
Width 128	16.371 ±0.145	2.384 ±0.009	3.881 ±0.016	0.388 ±0.004	42.191 ±0.098	1.959 ±0.009
Inner LR ×0.5	16.396 ±0.177	2.390 ±0.017	3.893 ±0.032	0.388 ±0.005	42.225 ±0.084	1.948 ±0.037
Inner LR ×2	16.328 ±0.191	2.378 ±0.010	3.869 ±0.017	0.387 ±0.005	42.172 ±0.106	1.963 ±0.009
Ratio 1:1	16.302 ±0.189	2.381 ±0.013	3.876 ±0.025	0.386 ±0.005	42.236 ±0.111	1.937 ±0.049
Ratio 2:1	16.327 ±0.183	2.384 ±0.016	3.882 ±0.030	0.387 ±0.005	42.241 ±0.075	1.958 ±0.006
Ratio 8:1	16.349 ±0.196	2.379 ±0.012	3.869 ±0.022	0.388 ±0.005	42.149 ±0.094	1.967 ±0.011

Default: Test width 64, Critic/Test learning rates $10^{-3}/3 \times 10^{-4}$, and four Critic/Test steps per outer iteration. Capacity changes the Test hidden width; inner-rate changes scale both Critic and Test rates; the ratio changes their update count with the Actor schedule fixed. Default records are reused from the same ten main-study seeds.

Table A6. AC online sensitivity: mean and sample SD over ten fixed seeds per setting.

Setting	Ann. Ret. (%)	Sharpe	Sortino	Calmar	MDD (%)	RT (Years)
Default	15.647 ±0.232	2.340 ±0.020	3.800 ±0.037	0.368 ±0.007	42.578 ±0.186	1.998 ±0.014
Width 32	15.633 ±0.229	2.339 ±0.019	3.798 ±0.036	0.367 ±0.007	42.593 ±0.189	1.996 ±0.013
Width 128	15.674 ±0.216	2.341 ±0.018	3.802 ±0.034	0.368 ±0.007	42.560 ±0.187	1.998 ±0.016
Inner LR ×0.5	15.614 ±0.234	2.338 ±0.020	3.797 ±0.036	0.367 ±0.007	42.573 ±0.182	1.992 ±0.014
Inner LR ×2	15.668 ±0.161	2.337 ±0.010	3.794 ±0.019	0.368 ±0.005	42.590 ±0.159	2.010 ±0.015
Ratio 1:1	15.623 ±0.209	2.339 ±0.018	3.800 ±0.034	0.367 ±0.006	42.550 ±0.165	1.988 ±0.015
Ratio 2:1	15.622 ±0.213	2.339 ±0.019	3.799 ±0.035	0.367 ±0.006	42.581 ±0.169	1.988 ±0.017
Ratio 8:1	15.696 ±0.191	2.337 ±0.013	3.795 ±0.023	0.368 ±0.006	42.605 ±0.166	2.014 ±0.015

Default: Test width 64, Critic/Test learning rates $10^{-3}/3 \times 10^{-4}$, and four Critic/Test steps per outer iteration. Capacity changes the Test hidden width; inner-rate changes scale both Critic and Test rates; the ratio changes their update count with the Actor schedule fixed. Default records are reused from the same ten main-study seeds.

Appendix C.7. Entropy Comparisons and Statistical Procedure

Table A7 reports all 24 paired comparisons underlying Table 5. The baseline is the unchanged main-experiment temperature $\gamma = 0.01$. We rank the seedwise alternative-minus-default differences, using Pratt handling for zero differences and tie-adjusted normal approximation, and apply Holm correction jointly to the 24 two-sided tests. Eighteen comparisons have adjusted $p \leq 0.05$; their signs identify the associated performance tradeoffs. For example, $\gamma = 0.05$ raises mean annual return by 0.1216 percentage points and mean MDD by 0.1499 percentage points. This distinction permits the statistical and financial effects to be read together.

Table A7. Paired entropy-temperature comparisons against $\gamma = 0.01$ (100 matched training seeds in every comparison).

γ	Metric	Preferred	Mean Diff.	p (Raw)	p (Holm)
0.005	Ann. ret. (pp)	↑	−0.0989	7.34×10^{-6}	7.34×10^{-5}
0.005	Sharpe	↑	+0.0039	0.1252	0.3157
0.005	Sortino	↑	+0.0084	0.0807	0.3157
0.005	Calmar	↑	−0.0029	2.69×10^{-6}	2.96×10^{-5}
0.005	MDD (pp)	↓	+0.0630	1.43×10^{-6}	1.72×10^{-5}
0.005	RT (years)	↓	+0.0146	2.17×10^{-14}	3.91×10^{-13}
0.02	Ann. ret. (pp)	↑	+0.0258	0.0028	0.0197
0.02	Sharpe	↑	+0.0066	2.87×10^{-11}	4.88×10^{-10}
0.02	Sortino	↑	+0.0130	3.46×10^{-11}	5.53×10^{-10}
0.02	Calmar	↑	+0.0001	0.7439	0.7439
0.02	MDD (pp)	↓	+0.0547	1.21×10^{-9}	1.69×10^{-8}
0.02	RT (years)	↓	−0.0038	7.72×10^{-10}	1.16×10^{-8}
0.05	Ann. ret. (pp)	↑	+0.1216	3.29×10^{-8}	4.27×10^{-7}
0.05	Sharpe	↑	+0.0237	1.90×10^{-17}	4.19×10^{-16}
0.05	Sortino	↑	+0.0479	1.15×10^{-17}	2.64×10^{-16}
0.05	Calmar	↑	+0.0015	0.0142	0.0854
0.05	MDD (pp)	↓	+0.1499	6.93×10^{-17}	1.45×10^{-15}
0.05	RT (years)	↓	−0.0038	4.41×10^{-4}	0.0035
0.1	Ann. ret. (pp)	↑	+0.0588	0.0166	0.0854
0.1	Sharpe	↑	+0.0200	5.46×10^{-16}	1.04×10^{-14}
0.1	Sortino	↑	+0.0425	1.07×10^{-16}	2.14×10^{-15}
0.1	Calmar	↑	−0.0009	0.0789	0.3157
0.1	MDD (pp)	↓	+0.2499	3.90×10^{-18}	9.35×10^{-17}
0.1	RT (years)	↓	+0.0062	1.02×10^{-5}	9.19×10^{-5}

Difference is alternative minus default. "pp" denotes percentage points. Arrows give the preferred direction. Tests are two-sided Wilcoxon signed-rank tests with Pratt zero handling, tie-adjusted normal approximation, and no continuity correction; differences are rounded to 12 decimal places before ranking. All 24 raw p-values enter one Holm family.

Appendix C.8. Fixed-Workload Scaling

The scaling study uses the same learning implementation at $d = 10, 100, 1000$, three fixed seeds per dimension, and a balanced dimension order on one Tesla T4. Each run performs 900 outer iterations, 850 Actor updates, and 3600 Critic and Test updates. The shared-asset parametrization keeps the trainable parameter count constant across these dimensions. Table A8 times the complete training loop after learner and historical-tensor preparation; market generation, history materialization, initialization, and final backtesting are recorded separately. This operation scope supplies a direct comparison with the per-update complexity calculation. Dense market-generation operations occur before this timer. Mean training time increases by 19.34% as the asset count grows from 10 to 1000 under this fixed workload.

Table A8. Fixed-workload training times on one Tesla T4: mean and sample SD over three seeds, for 900 outer iterations at each asset dimension.

Assets d	**10**	**100**	**1000**
Training time (seconds)	102.738 ± 4.922	102.196 ± 3.157	122.610 ± 3.076

Appendix C.9. Reproducibility Settings

The experiments are run in Kaggle with Tesla T4 devices, Python 3.13, PyTorch 2.11 with CUDA 12.8, and NumPy 2.1.3. Float32 is used for training, float64 for conditional references and Gamma evaluation. Controlled timing uses one GPU and one CPU thread serially; the long main and sensitivity suites dispatch independent jobs across the available T4 devices. The scaling accelerator configuration is confirmed from the run setup. Tables A9 and A10 give the method configurations.

The portfolio timing analysis evaluates Sharpe thresholds of 2.25, 2.30, and 2.35 from the recorded training checkpoints, using the same timing and confirmation rules.

For the option networks, writing $\tau = T - t$ and $z_x = d^{-1}\sum_i x_i - K$, the terminal-conditioned representation is

$$v^\theta(t,x) = e^{-r\tau} b_\tau(z_x) + \tau u^\theta(t,x), \quad b_\tau(z) = 0.15\sqrt{\tau}\log\{1 + \exp(z/(0.15\sqrt{\tau}))\}, \quad \tau > 0, \tag{A44}$$

and $b_0(z) = z^+$. The payoff is therefore imposed exactly at maturity, with a smooth interior lift for derivative evaluation. Both ML and ADV use this representation. In ADV, the discounted residual rate is $[e^{-rh}v^\theta(t+h, X') - v^\theta(t,X)]/h$. With two independently drawn local batches conditional on the pilot paths, the Critic minimizes $0.1\widehat{G}_1^\top \widehat{G}_2 + 0.3\widehat{A}_{\text{path}}$, and the Test minimizes $-\widehat{G}_1^\top \widehat{G}_2$. The anchor averages squared differences between $e^{-rt_n}v^\theta(t_n, X_n)$ and $e^{-rT}g(X_T)$ on 16 sampled full paths and all nonterminal grid points. The bounded Test has 128 outputs with norm at most three. Updates use gradient clipping at norm five.

For the portfolio Critic, the three outputs represent remaining gross-return first moment m_1, second moment m_2, and accumulated entropy e. Their terminal constraints are $m_1 = m_2 = 1$ and $e = 0$, enforced through the episode time variable. The value is

$$v_w^\theta(s,X) = -X^2 m_2^\theta(s,w) + 2wXm_1^\theta(s,w) - w^2 + (w-z)^2 + e^\theta(s,w). \tag{A45}$$

For a sampled episode, set $F_t = X_T/X_t$ and $E_t = \sum_{j\geq t}\gamma\Delta\overline{H}(\pi_j)$. The physical-moment anchor is

$$\widehat{A}_{\text{mom}} = \widehat{\mathbb{E}}\big[(m_1 - F_t)^2 + 0.25(m_2 - F_t^2)^2 + (e - E_t)^2\big]. \tag{A46}$$

The local residual retains the same weak-moment gradient construction, while these terminal physical quantities supply the anchor targets. All source and target features for the online updates are drawn from observations completed by the update time.

Table A9. Basket-option numerical configuration. Hidden-width tuples exclude input and output layers.

Component	Setting
Value network (ML, ADV)	$(128, 128, 64)$ softplus; 101 inputs, one output
Test network (ADV)	$(256, 128)$ softplus; 101 inputs, 128 bounded outputs
Deep BSDE	50 separate $(128, 64)$ softplus Delta subnetworks and scalar Y_0
Optimizers	ML/BSDE: Adam; ADV: AdamW
Initial learning rates	ML, BSDE, ADV value: 10^{-3}; ADV Test: 5×10^{-4}
ADV weight decay	Value: 10^{-3}; Test: 5×10^{-4}
Rate schedules	ML/ADV: multiply by 0.2 at 60%, 85% of outer budget
BSDE rate schedule	Multiply by 0.1 at 55%, 80% of outer budget
Updates/outer iteration	ML/BSDE: one; ADV: four value and two Test
Full-path/local batch	128/128; 5000 outer iterations
Anchor/weak weights	0.3/0.1; anchor uses 16 sampled paths
Parameter averaging	EMA decay 0.99, all value/Delta parameters, including Y_0
Initialization	Shared ML/ADV initial weights; value output SD 3, first-layer scale 4
Development/confirmation	Three development seeds per rate; all 20 fixed confirmation seeds
Sensitivity/controlled timing	Seeds 5101, 5107, 5113; separate from development

The fixed confirmation seeds are 5101, 5107, 5113, 5119, 5147, 5153, 5167, 5171, 5179, 5189, 5197, 5209, 5227, 5231, 5233, 5237, 5261, 5273, 5279, 5281. Development seeds 6101, 6113, 6121 compare ADV value rates 10^{-3} and 2.5×10^{-4}; the recorded selection rule retains 10^{-3}. Validation and final reference streams use seeds 9710101 and 9710201.

Table A10. Portfolio market, representation, and learning configuration.

Component	Setting
Market and split	100 assets; market seed 231; 3999 training and 4000 test returns
Regime blocks	Bull/bear/bull/bear/bull: 2700/300/2700/300/2000 prices
Annual drift across assets	Bull: 0.05–0.25; bear: -0.60–-0.45
Annual volatility	Bull: 0.15–0.25; bear: 0.05–0.15
Transitions/correlation	25-day parameter transitions; correlation 0.1
Episode/holding period	252/21 trading days; exact observed block gross factors
Target/entropy/cost	$z = 1.15$; $\gamma = 0.01$; normalized Dirichlet entropy; fee 0.001
Causal features	Market mean/volatility (63 days), drawdown; asset means (252/1260), volatility (252)
Shared Actor	Two hidden layers of width 64, SiLU; per-asset input 9
Critic/Test	Two hidden layers of width 128/64, SiLU; 12/11 inputs
Critic/Test outputs	Three physical-moment outputs/nine fixed plus five learned features
Optimizer and rates	Adam; Actor 3×10^{-4}, Critic 10^{-3}, Test 3×10^{-4}
Outer/inner updates	900 outer; four Critic/Test steps; 50 Actor warmup iterations
Episodes/mini-batch	32 one-year episodes; mini-batch 384
Anchor/weak weights	0.25/1; gradient clipping five

Table A11. Portfolio adaptation, benchmarks, and repeated-run configuration.

Component	Setting
Multiplier	Rate 0.25 every five outer updates; projection to $[0, 10]$
Online adaptation	Four outer updates per completed block; recent 1260 days plus 50% pretraining replay
Online learning rates	All network and multiplier rates multiplied by 0.1
PPO	Four epochs; clip 0.2; GAE 0.95
DDPG	Polyak rate 0.01; replay 100,000; four Critic/Actor steps
MV/DRMV/CTMV	Rolling 252 days; covariance shrinkage 0.1; DRMV radius 0.02
Classical optimizer	SLSQP; max 400 iterations, tolerance 10^{-9}; fallback events logged
Fixed main seeds	10001–10100 in each mode; all 600 learning runs included
Target-time seeds/budget	89001–89020; 180 s; nominal five-second evaluations
Scaling seeds	10001–10003 at each of $d = 10, 100, 1000$

All methods are long-only and fully invested. If their estimated target is infeasible, the maximum attainable estimated-mean allocation is used; CTMV uses EW if its plug-in direction has no positive component. The Actor, Critic, and Test parameter counts stay fixed with d under the shared-asset structure. At $d = 100$, the benchmark input data and costs match the main comparison.

References

1. Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2016; pp. 1329–1338.
2. Osband, I.; Van Roy, B.; Wen, Z. Generalization and exploration via randomized value functions. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2016; pp. 2377–2386.
3. Fleming, W.H.; Soner, H.M. *Controlled Markov Processes and Viscosity Solutions*; Springer: Berlin/Heidelberg, Germany, 2006.
4. Yong, J.; Zhou, X.Y. *Stochastic Controls: Hamiltonian Systems and HJB Equations*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999; Volume 43.
5. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533.
6. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
7. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An introduction*; MIT Press: Cambridge, MA, USA, 1998; Volume 1.
8. Munos, R. Policy gradient in continuous time. *J. Mach. Learn. Res.* **2006**, *7*, 771–791.
9. Tallec, C.; Blier, L.; Ollivier, Y. Making deep q-learning methods robust to time discretization. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019; pp. 6096–6104.
10. Doya, K. Reinforcement learning in continuous time and space. *Neural Comput.* **2000**, *12*, 219–245.
11. Jia, Y.; Zhou, X.Y. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *J. Mach. Learn. Res.* **2022**, *23*, 1–55.
12. Li, G.; Cai, C.; Chen, Y.; Wei, Y.; Chi, Y. Is Q-learning minimax optimal? a tight sample complexity analysis. *Oper. Res.* **2024**, *72*, 222–236.
13. Mao, W.; Zhang, K.; Zhu, R.; Simchi-Levi, D.; Başar, T. Model-Free Nonstationary Reinforcement Learning: Near-Optimal Regret and Applications in Multiagent Reinforcement Learning and Inventory Control. *Manag. Sci.* **2025**, *71*, 1564–1580.
14. Yang, Y.; Luo, R.; Li, M.; Zhou, M.; Zhang, W.; Wang, J. Mean field multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 5571–5580.
15. Liu, X. Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Mark. Sci.* **2023**, *42*, 637–658.
16. Yan, Y.; Li, G.; Chen, Y.; Fan, J. Model-based reinforcement learning for offline zero-sum markov games. *Oper. Res.* **2024**, *72*, 2430–2445.
17. Ata, B.; Harrison, J.M.; Si, N. Drift control of high-dimensional reflected brownian motion: A computational method based on neural networks. *Stoch. Syst.* **2025**, *15*, 111–146.
18. You, H.; Xiang, M.; Yang, Z.; Jing, Y.; Guo, Y.; Cui, J.Z. A multiscale phase field model for anisotropic brittle and ductile fracture in geomaterials with randomly distributed microcracks. *J. Mech. Phys. Solids* **2026**, *217*, 106795. https://doi.org/10.1016/j.jmps.2026.106795.
19. Ma, Q.; Wu, Y.; Bi, L.; Cui, J.Z.; Wang, H.; Chen, T. Multiscale asymptotic analysis and algorithm for the quadratic eigenvalue problem in composite materials. *Comput. Appl. Math.* **2023**, *42*, 219.
20. Bénézech, J.; Seelinger, L.; Bastian, P.; Butler, R.; Dodwell, T.; Ma, C.; Scheichl, R. Scalable multiscale-spectral GFEM with an application to composite aero-structures. *J. Comput. Phys.* **2024**, *508*, 113013.
21. Wang, H.; Zhou, X.Y. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Math. Finance* **2020**, *30*, 1273–1308.
22. Jia, Y.; Zhou, X.Y. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *J. Mach. Learn. Res.* **2022**, *23*, 1–50.
23. Jia, Y.; Zhou, X.Y. q-Learning in continuous time. *J. Mach. Learn. Res.* **2023**, *24*, 1–61.
24. Karatzas, I.; Shreve, S. *Brownian Motion and Stochastic Calculus*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 113.
25. Cai, W.; Fang, S.; Zhang, W.; Zhou, T. Martingale deep learning for very high dimensional quasi-linear partial differential equations and stochastic optimal controls. *arXiv* **2024**, arXiv:2408.14395.
26. Cohen, S.N.; Elliott, R.J. *Stochastic Calculus and Applications*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 2.
27. Van Kampen, N.G. Stochastic differential equations. *Phys. Rep.* **1976**, *24*, 171–228.
28. Tang, W.; Zhang, Y.P.; Zhou, X.Y. Exploratory HJB equations and their convergence. *SIAM J. Control Optim.* **2022**, *60*, 3191–3216.
29. Sutton, R.S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **1988**, *3*, 9–44.
30. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407.
31. E, W.; Han, J.; Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.* **2017**, *5*, 349–380.
32. Han, J.; Jentzen, A.; E, W. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 8505–8510.

33. Markowitz, H.M. Portfolio Selection. *J. Finance* **1952**, *7*, 71–91.
34. Zhou, X.Y.; Li, D. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Appl. Math. Optim.* **2000**, *42*, 19–33.
35. Merton, R.C. On estimating the expected return on the market: An exploratory investigation. *J. Financ. Econ.* **1980**, *8*, 323–361.
36. Ledoit, O.; Wolf, M. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *J. Empir. Finance* **2003**, *10*, 603–621.
37. Black, F.; Litterman, R. Asset allocation: combining investor views with market equilibrium. *Goldman Sachs Fixed Income Res.* **1990**, *115*, 7–18.
38. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; JHU Press: Baltimore, MD, USA, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.