

NEURAL POLICY ITERATION FOR STOCHASTIC OPTIMAL CONTROL: A PHYSICS-INFORMED APPROACH

Anonymous authors

Paper under double-blind review

ABSTRACT

We propose a physics-informed neural network policy iteration (PINN-PI) framework for solving stochastic optimal control problems governed by second-order Hamilton–Jacobi–Bellman (HJB) equations. At each iteration, a neural network is trained to approximate the value function by minimizing the residual of a linear PDE induced by a fixed policy. This linear structure enables systematic L^2 error control at each policy evaluation step, and allows us to derive explicit Lipschitz-type bounds that quantify how value gradient errors propagate to the policy updates. This interpretability provides a theoretical basis for evaluating the quality of policy during training. Our method extends recent deterministic PINN-based approaches to stochastic settings, inheriting the global exponential convergence guarantees of classical policy iteration under mild conditions. We demonstrate the effectiveness of our method on several benchmark problems, including stochastic cartpole, pendulum problems and high-dimensional linear quadratic regulation (LQR) problems in up to 20D.

1 INTRODUCTION

Solving infinite-horizon stochastic optimal control problems requires computing the value function, which satisfies a nonlinear Hamilton–Jacobi–Bellman (HJB) partial differential equation (PDE). In high dimensions, traditional numerical methods become intractable due to the curse of dimensionality. While Howard’s policy iteration (PI) Howard (1960); Puterman & Brumelle (1979); Puterman (1981) provides a theoretically grounded and convergent scheme, each iteration requires solving a linear PDE, which becomes the computational bottleneck in practice.

Recent theoretical works have extended PI to various settings: deterministic control Tang et al. (2025), stochastic control under viscosity solutions Jacka & Mijatović (2017); Kerimkulov et al. (2020), and entropy-regularized (exploratory) formulations Tran et al. (2025b); Huang et al. (2025). These developments underscore the robustness of the PI framework, but also highlight the need for scalable solvers that can handle high-dimensional PDEs with theoretical guarantees.

In this work, we propose a mesh-free, physics-informed policy iteration framework for solving stochastic optimal control problems governed by second-order HJB equations. Our method integrates classical PI with physics-informed neural networks (PINNs): at each iteration, the value function is approximated by a neural network trained to minimize the residual of the linear PDE associated with a fixed policy. By fixing the policy at each iteration, we obtain a linear PDE for the value function, which contrasts with the fully nonlinear HJB equation. This linearity enables the use of classical energy estimates to control the L^2 error, which would be difficult to establish under direct optimization of the full HJB.

Unlike model-free reinforcement learning or trajectory-based PINN approaches, our method directly targets the PDE structure of the control problem. This yields both theoretical interpretability and numerical scalability. In particular, we show that value gradient error controls policy error through a Lipschitz-type bound, enabling policy quality monitoring throughout training.

We validate our approach on high-dimensional stochastic control tasks, including LQR, pendulum, and cartpole problems. Results confirm that our method retains the convergence and stability of classical PI while benefiting from the flexibility of neural PDE solvers. Our main contributions are:

- We propose a physics-informed, mesh-free policy iteration framework for solving high-dimensional stochastic control problems governed by nonlinear HJB equations.
- We establish a rigorous L^2 error analysis with a decomposition into iteration error, residual error, and policy mismatch, and prove global exponential convergence under standard assumptions.
- We demonstrate the accuracy and scalability of our approach on a variety of nonlinear stochastic control benchmarks.

2 RELATED WORK

Classical Policy Iteration. Policy iteration (PI) was first formalized by Howard Howard (1960) and later analyzed in depth by Puterman et al. Puterman & Brumelle (1979), who connected PI to Newton–Kantorovich iterations and established convergence rates. In continuous-time settings, Puterman Puterman (1981) extended these ideas to controlled diffusion processes, showing convergence under appropriate assumptions.

Viscosity Methods for HJB Equations. In the context of continuous-time stochastic control, monotone convergence of PI under a weak solution framework has been established Jacka & Mijatović (2017). Slightly later, global exponential convergence using BSDE-based techniques were proposed Kerimkulov et al. (2020). For deterministic control problems, the convergence of PI was analyzed within the framework of viscosity solution Tang et al. (2025).

Entropy-Regularized and Exploratory Control. Entropy-regularized HJB equations arise in exploratory control settings, where the optimal policy is stochastic due to the inclusion of an entropy term in the objective. Convergence of policy iteration in this context has been studied extensively Tran et al. (2025a); Huang et al. (2025). In particular, under the assumption that the diffusion coefficient depends weakly (or not at all) on the control variable, geometric convergence has been established.

Physics-Informed and Neural Approaches. Physics-informed neural networks (PINNs) Raissi et al. (2019) have emerged as mesh-free alternatives for solving high-dimensional PDEs, offering flexibility and scalability beyond traditional discretization methods. Neural variants of policy iteration have combined these tools with classical control frameworks. One such approach introduces ELM-PI and PINN-PI, which solve linearized PDEs in deterministic control problems and support Lyapunov-based stability verification Meng et al. (2024). Additional extensions include nonconvex formulations Yang et al. (2025), operator-learning-based architectures Lee & Kim (2025), and reinforcement learning methods that integrate differentiable physics or PDE solvers into model-based pipelines Ramesh & Ravindran (2023); Mukherjee & Liu (2023).

Our work addresses general stochastic control problems with nonlinear dynamics and compact action spaces, and develops a rigorous L^2 -error analysis aligned with residual loss minimization. We prove exponential convergence under classical policy iteration, offering a quantitative decomposition of total error that accounts for both approximation and policy mismatch.

3 INFINITE-HORIZON STOCHASTIC OPTIMAL CONTROL

Let $(W_t)_{t \geq 0}$ be a d -dimensional Brownian motion on a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{P})$. A bounded and measurable control process $a_t \in A \subset \mathbb{R}^m$ drives the controlled diffusion

$$dX_t = b(X_t, a_t) dt + \sigma dW_t, \quad X_0 = x \in \mathbb{R}^d, \quad (1)$$

where σ is a constant matrix. The goal is to maximize the infinite-horizon discounted cost

$$J(x, a) = \mathbb{E}_x \left[\int_0^\infty e^{-\lambda s} L(X_s, a_s) ds \right], \quad \lambda > 0. \quad (2)$$

With the value function defined as $V(x) := \sup_a J(x, a)$, it is known from literature Tran (2021); Evans (2022) that $V \in \text{Lip}(\mathbb{R}^d)$ is a unique viscosity solution to

$$\lambda V - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 V) - \sup_{a \in A} \{b \cdot \nabla_x V + L\} = 0, \quad (3)$$

under some regularity assumptions on b, L . Then the optimal control is given by

$$a^*(x) := \arg \max_{a \in A} \{b(x, a) \cdot \nabla_x V(x) + L(x, a)\},$$

where measurable selection guarantees the measurability of the control.

In the remainder of this section, we formalize the mathematical setting, assumptions, and notation used throughout the paper.

Notations Let us begin by introducing the notations used throughout the paper. For $x \in \mathbb{R}^d$, we write $|x|$ for the Euclidean norm. Given a function $f : \Omega \rightarrow \mathbb{R}^n$, we denote its standard L^p norm by

$$\|f\|_p := \left(\int_{\Omega} |f|^p dx \right)^{1/p}, \quad \text{for } p \in [1, \infty].$$

The Hessian of f is denoted $D_{xx}^2 f$. We say $f \in H^1(\mathbb{R}^d) = W^{1,2}(\mathbb{R}^d)$ if $\int_{\Omega} |f|^2 + |\nabla_x f|^2 < \infty$. For $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\text{div}_x g := \sum_{i=1}^d \partial_{x_i} g_i$ where g_i denotes the i th component of g .

Assumption 1. *We impose the following assumptions throughout the paper.*

(A1) *Control set $A \subset \mathbb{R}^d$ is compact and convex.*

(A2) *$b : \mathbb{R}^d \times A \rightarrow \mathbb{R}$ are continuously differentiable and Lipschitz continuous. In addition, $b(x, a)$ satisfies:*

- $\lambda > B/2$ where

$$B := \sup_{a \in A} (\|b(\cdot, a)\|_{\infty} + \|\text{div}_x b(\cdot, a)\|_{\infty}) < \infty,$$

- the Jacobian $\partial_a b(x, a)$ is uniformly bounded,
- there exists a constant $\tilde{B} > 0$ satisfying

$$|\partial_a b(x, a)| + \frac{|\partial_a b(x, a) - \partial_a b(x, a')|}{|a - a'|} \leq \tilde{B}.$$

(A3) *$L(x, a) \geq 0$ is uniformly L_a -Lipschitz continuous and μ_a -strongly convex in a . Furthermore, there exist constants $R > 0$, $\beta > d + 2$, and $C > 0$ such that*

$$\sup_{a \in A} L(x, a) \leq \frac{C}{(1 + |x|)^{\beta}}, \quad \text{for all } |x| \geq R.$$

(A4) *$\sigma \sigma^\top$ is uniformly elliptic with eigenvalues bounded between $0 < \nu \leq \Lambda$.*

Assumption (A3) ensures that the running cost $L(x, a)$ decays sufficiently fast at infinity, while (A4) guarantees nondegenerate diffusion. These together imply that the value function $V(x) = \sup_a \mathbb{E}_x[\int_0^\infty e^{-\lambda t} L(X_t, a_t) dt]$ is integrable over $\mathbb{R}^d$, i.e., $V \in L^2(\mathbb{R}^d)$, via standard estimates on stochastic processes with confining cost structure.

4 HOWARD’S POLICY IMPROVEMENT ALGORITHM

Algorithm 1 Policy Improvement

1: **Input:** initial Markov policy $a_0(\cdot)$.

2: **for** $n = 0, 1, 2, \dots$ **do**

3: *Policy evaluation:* solve for v_n on $\mathbb{R}^d$

$$\lambda v_n - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 v_n) - b(\cdot, a_n) \cdot \nabla_x v_n = L(\cdot, a_n).$$

4: *Policy improvement:*

$$a_{n+1}(x) = \arg \max_{a \in A} \{L(x, a) + b(x, a) \cdot \nabla_x v_n(x)\}.$$

5: **if** $\|v_{n+1} - v_n\|_\infty < \varepsilon$ **then stop.**

6: **end for**

Howard’s policy iteration alternates between evaluating the cost of a fixed policy and improving it by acting greedily with respect to the current value function. This structure naturally aligns with model-based reinforcement learning methods and enables interpretable control synthesis in continuous domains. Unlike value iteration, which updates the value function directly via a fixed-point operator, policy iteration produces stable value approximations by solving a linear PDE at each step.

The key advantage of this method lies in the decoupling of policy evaluation and improvement: the former reduces to solving a linear PDE, and the latter often admits a closed-form optimizer when $L(x, a) + b(x, a) \cdot \nabla_x V(x)$ is convex in a . This makes PI especially appealing for structured control problems where the policy improvement step can be implemented efficiently.

However, the main computational bottleneck in Howard’s method lies in solving the high-dimensional linear PDE in each policy evaluation step. Traditional finite-difference or finite-element schemes scale poorly in high dimensions. In the next section, we propose to overcome this limitation using physics-informed neural networks (PINNs), which serve as flexible, mesh-free solvers capable of approximating solutions in high-dimensional domains.

5 PHYSICS-INFORMED HOWARD POLICY ITERATION

We propose a PINN-based variant of Howard’s policy iteration, where the value function at each iteration is approximated by a neural network trained to minimize the PDE residual at sampled collocation points. Let $\{x_i\}_{i=1}^N \subset \Omega$ be a set of N collocation points sampled from the domain.¹ Given a fixed policy $a_n(x)$, we approximate the corresponding value function $v_n(x)$ by a neural network $v_n(x; \theta)$ and define the residual of the linear PDE:

$$\begin{aligned} \mathcal{L}(\theta) := & \frac{1}{N} \sum_{i=1}^N \left| \lambda v_n(x_i; \theta) - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 v_n(x_i; \theta)) \right. \\ & \left. - b(x_i, a_n(x_i)) \cdot \nabla_x v_n(x_i; \theta) - L(x_i, a_n(x_i)) \right|^2 \end{aligned} \quad (4)$$

Before analyzing the convergence of our proposed method, we first establish a stability result for the linear PDE solved at each policy evaluation step. Specifically, for a fixed measurable policy $a_n: \mathbb{R}^d \rightarrow A$, the value function v_n satisfies a linear elliptic PDE of the form:

$$\lambda v_n - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 v_n) - b_n \cdot \nabla_x v_n = k,$$

where $b_n(x) := b(x, a_n(x))$ and $k(x) \in L^2(\mathbb{R}^d)$ is a given forcing term. The following proposition shows that under mild assumptions, this equation admits a unique weak solution $v_n \in H^1(\mathbb{R}^d)$, with an energy estimate that is uniform in the choice of the measurable policy a_n . This result plays a key role in subsequent error analysis.

¹Although the theoretical analysis is conducted over $\mathbb{R}^d$, we assume a bounded domain $\Omega \subset \mathbb{R}^d$ in practice since only a finite number of collocation points are sampled.

Algorithm 2 Physics-Informed Neural Network Policy Iteration (PINN-PI)

```

1: Input: Initial policy  $a_0(\cdot)$ , number of collocation points  $N$ , domain  $\Omega$ , initial network parameters  $\theta_0$ 
2: for  $n = 0, 1, 2, \dots$  do
3:   Collocation sampling: Sample  $\{x_i\}_{i=1}^N \subset \Omega$ 
4:   Policy evaluation: Train neural network  $v_n(x; \theta)$  by minimizing the residual loss defined in equation 4
5:   Policy improvement:

$$a_{n+1}(x) := \arg \max_{a \in A} \{L(x, a) + b(x, a) \cdot \nabla_x v_n(x; \theta_n)\}$$

6:   If stopping criterion met (e.g.  $\|a_{n+1} - a_n\|_\infty < \varepsilon$ ) then stop
7: end for

```

Proposition 1 (L^2 estimate with a measurable policy a_n). *Suppose Assumption 1 holds. Let $a_n: \mathbb{R}^d \rightarrow A$ be any measurable policy and set $b_n(x) := b(x, a_n(x))$. For $k \in L^2(\mathbb{R}^d)$ consider the PDE*

$$\lambda v_n - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 v_n) - b_n \cdot \nabla_x v_n = k \quad \text{in } \mathbb{R}^d.$$

Then there is a unique weak solution $v_n \in H^1$ satisfying

$$(\lambda - \frac{1}{2}B) \|v\|_2^2 + \frac{\nu}{2} \|\nabla_x v\|_2^2 \leq (k, v_n),$$

where $(f, g) := \int_{\mathbb{R}^d} f g \, dx$. Therefore,

$$\|v\|_2 \leq C_\lambda \|\tilde{r}\|_2, \quad \|\nabla_x v\|_2 \leq C_\lambda \|k\|_2,$$

where $C_\lambda = \max\{\frac{1}{\lambda - \frac{1}{2}B}, \sqrt{\frac{1}{\nu(\lambda - \frac{1}{2}B)}}\}$.

For the sake of completeness, the proof is provided in Appendix A.1 While the proposition above ensures the stability of each policy evaluation step in the L^2 sense, it does not by itself guarantee that the updated policy improves over iterations. To analyze the overall convergence behavior of policy iteration, it is crucial to understand how the quality of the value function approximation, particularly its gradient, affects the resulting policy.

To this end, the next proposition shows that the policy improvement map is Lipschitz continuous with respect to the value gradient. This result allows us to quantify how errors in the value approximation propagate to the policy error in a stable manner, which is a key ingredient in establishing exponential convergence.

Proposition 2 (Policy error controlled by value-gradient error). *Assume (A1)–(A4) and let $|z|, |z'| \leq M$ for some M such that $\mu_a > M\tilde{B}$. Fix $x \in \mathbb{R}^d$ and define the selector*

$$a^*(x, z) := \arg \max_{a \in A} \{L(x, a) + b(x, a) \cdot z\}, \quad z \in \mathbb{R}^d.$$

Then a^ is globally Lipschitz in z with constant $\theta > 0$:*

$$|a^*(x, z) - a^*(x, z')| \leq \theta |z - z'| \quad \text{for } z, z' \in \mathbb{R}^d.$$

The proof is presented in Appendix A.2.

In Kerimkulov et al. (2020), the pointwise exponential convergence has been established, which yields that

$$0 \leq v_n(x) - V(x) \leq C\eta^n, \quad \forall x \in \mathbb{R}^d,$$

for some $\eta \in (0, 1)$. However, in the framework of PINNs, L^2 is more suitable so, we now establish the exponential convergence property of v_n to V in L^2 .

Throughout this section, we use

$$C_R := \theta(L_a + \tilde{B} \max_{p \in \{2, \infty\}} \{\|\nabla_x V\|_p, \|\nabla_x v_n\|_p\}),$$

where V is a unique solution to equation 3 and $\{v_n\}_{n \geq 0}$ is generated via Algorithm 1. Here, by classical theory of elliptic PDEs Evans (2022) and the Lipschitz continuity of V , this C_R is finite. For brevity, let us define

$$\mathcal{T}[v, a] := \lambda v - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 v) - b(\cdot, a) \cdot \nabla_x v - L(\cdot, a).$$

Theorem 1 (Global exponential convergence of Howard–PI). *Let Assumption 1 hold and V be a unique viscosity solution to equation 3 with continuous gradient. If $\tilde{\kappa} := \sqrt{\frac{C_R^2}{\nu(\lambda - \frac{1}{2}B)}} \in (0, 1)$, then we have*

$$\|v_n - V\|_2 \leq C \tilde{\kappa}^n, \quad x \in \mathbb{R}^d,$$

where $\{(v_n, a_n)\}_{n \geq 0}$ be produced by Algorithm 1, and C is a problem dependent constant.

We introduce a lemma that links the value functions and their gradients.

Lemma 1. *With C_R defined above, we have that*

$$\|v_n - v_m\|_2 \leq \tilde{C}_\lambda \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2,$$

and

$$\|\nabla_x v_n - \nabla_x v_m\|_2 \leq \tilde{C}_\lambda \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2,$$

where $\tilde{C}_\lambda = \max\{\frac{C_R}{\lambda - \frac{1}{2}B}, \sqrt{\frac{C_R^2}{\nu(\lambda - \frac{1}{2}B)}}\}$.

The proof of this lemma is provided in Appendix A.3

Proof of Theorem 1. Recalling

$$a^*(x) := \arg \max_{a \in A} \{b(x, a) \cdot \nabla_x V(x) + L(x, a)\},$$

$V \in \text{Lip}(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ is a unique viscosity solution to

$$\lambda V - \frac{1}{2} \text{Tr}(\sigma \sigma^\top D_{xx}^2 V) - b(\cdot, a^*) \cdot \nabla_x V - L(\cdot, a^*) = 0.$$

Subtracting from $\mathcal{T}[v_n, a_n]$, we achieve

$$\lambda e_n - \frac{1}{2} \text{Tr}(\sigma \sigma^\top D_{xx}^2 e_n) - b(\cdot, a_n) \cdot \nabla_x e_n = R_n,$$

where $e_n := v_n - V$ and $R_n := [b(\cdot, a_n) - b(\cdot, a^*)] \cdot \nabla_x V + [L(\cdot, a_n) - L(\cdot, a^*)]$.

By applying Lemma 1 with v_n and $v_m = V$, we deduce that

$$\|\nabla_x e_n\|_2 \leq \sqrt{\frac{C_R^2}{\nu(\lambda - \frac{1}{2}B)}} \|\nabla_x e_{n-1}\|_2,$$

and hence,

$$\|\nabla_x e_n\|_2 \leq \kappa^n \|\nabla_x e_0\|_2.$$

Now from

$$\|e_n\|_2 \leq \frac{C_R}{\lambda - \frac{1}{2}B} \|\nabla_x e_{n-1}\|_2,$$

we conclude that

$$\|e_n\|_2 \leq C \kappa^n.$$

for some global constant $C > 0$. □

Finally, combining Theorem 1 and Proposition 3 introduced below, we arrive at a quantitative bound on the total approximation error of the PINN-based policy iteration method. Specifically, with $\{\tilde{v}_n\}_{n \geq 0}$ be generated via Algorithm 2, we define the three error components

$$\delta_n := \tilde{v}_n - \hat{v}_n, \quad \varepsilon_n := \hat{v}_n - v_n, \quad \epsilon := v_n - V, \quad (5)$$

so that $\tilde{e}_n = \delta_n + \varepsilon_n + e_n$, where $e_n := v_n - V$ is the ideal policy-iteration error. There exists a (user-chosen) tolerance $p_n > 0$ such that

$$\|r_n\|_2 \leq p_n, \quad n = 0, 1, \dots, \quad (6)$$

where

$$r_n := \lambda \tilde{v}_n - \frac{1}{2} \text{tr}(\sigma \sigma^\top D_{xx}^2 \tilde{v}_n) - b(\cdot, \tilde{a}_n) \cdot \nabla_x \tilde{v}_n - L(\cdot, \tilde{a}_n). \quad (7)$$

The distinction between $\hat{v}_n$ and $\tilde{v}_n$ is essential for understanding the approximation error introduced by the PINN surrogate, $\tilde{v}_n$. Specifically, $\hat{v}_n$ denotes the exact solution to the linear PDE associated with the frozen policy $\tilde{a}_n$ derived from $\tilde{v}_{n-1}$, which means that $\mathcal{T}[\hat{v}_n, \tilde{a}_n] = 0$. In contrast, $\tilde{v}_n$ is the neural approximation to $\hat{v}_n$ obtained by minimizing the residual loss equation 4 at a finite set of collocation points. Therefore, $\delta_n = \tilde{v}_n - \hat{v}_n$ captures the discrepancy due to numerical training, discretization, and model capacity limitations of the PINN. Importantly, δ_n is fully controlled by the optimization procedure and serves as the primary source of empirical error in our framework.

Proposition 3 (Policy-mismatch recursion). *Let $\{(\tilde{v}_n, \tilde{a}_n)\}_{n \geq 0}$ be generated via Algorithm 2, and $\kappa := \tilde{C}_\lambda \in (0, 1)$ with λ sufficiently large. Then, under the same assumption as in Theorem 1, we have that*

$$\|\delta_n\|_2 + \|\varepsilon_n\|_2 \leq C(p + \kappa^n). \quad (8)$$

for some problem dependent constant $C > 0$ where $p = \sup_n p_n$.

Proof. To estimate δ_n , we recall that $\mathcal{T}[\tilde{v}_n, \tilde{a}_n] = r_n$ with r_n defined in equation 4 and $\mathcal{T}[\hat{v}_n, \tilde{a}_n] = 0$. Subtracting two, with C_λ from Proposition 1, we have

$$\|\delta_n\|_2 \leq C_\lambda p_n \leq C_\lambda p.$$

We now estimate $\varepsilon = \hat{v}_n - v_n$. Noting that $\hat{v}_n$ and v_n satisfy $\mathcal{T}[\hat{v}_n, \tilde{a}_n] = 0$ and $\mathcal{T}[v_n, a_n] = 0$, we invoke Lemma 1 with $\hat{v}_n$ and v_n . Hence,

$$\|\varepsilon\|_2 = \|\hat{v}_n - v_n\|_2 \leq C_\lambda \|\nabla_x \tilde{v}_{n-1} - \nabla_x v_{n-1}\|,$$

since $\tilde{a}_n$ is induced by $\tilde{v}_{n-1}$.

Applying Lemma 1 once again with $\tilde{v}_{n-1}$ and v_{n-1} , we have

$$\begin{aligned} \|\nabla_x \tilde{v}_{n-1} - \nabla_x v_{n-1}\|_2 &\leq \|\nabla_x \tilde{v}_{n-1} - \nabla_x \hat{v}_{n-1}\|_2 + \|\nabla_x \hat{v}_{n-1} - \nabla_x v_{n-1}\|_2 \\ &\leq C_\lambda p_{n-1} + \tilde{C}_\lambda \|\nabla_x \tilde{v}_{n-2} - \nabla_x v_{n-2}\| \end{aligned}$$

Denoting $g_n := \|\nabla_x \tilde{v}_n - \nabla_x v_n\|_2$, we have that

$$g_{n-1} \leq C_\lambda p_{n-1} + \tilde{C}_\lambda g_{n-2},$$

which leads to

$$g_{n-1} \leq (\tilde{C}_\lambda)^{n-1} g_0 + C_\lambda p \sum_{i=0}^{n-1} (\tilde{C}_\lambda)^i \leq g_0 \kappa^{n-1} + \frac{C_\lambda p}{1 - \kappa}.$$

since $\kappa = \tilde{C}_\lambda \in (0, 1)$. Therefore,

$$\|\nabla_x \tilde{v}_{n-1} - \nabla_x v_{n-1}\|_2 \leq \kappa^{n-1} \|\nabla_x \tilde{v}_0 - \nabla_x v_0\|_2 + \frac{C_\lambda p}{1 - \kappa},$$

and thereby,

$$\|\varepsilon_n\|_2 \leq C(\kappa^n + p),$$

for some $C > 0$. □

Theorem 1 establishes exponential convergence of Howard’s method under exact policy evaluation. In practice, however, our PINN-based framework introduces approximation errors due to finite training, neural network capacity, and collocation sampling. These errors manifest as discrepancies between the neural surrogate $\tilde{v}_n$ and the exact PDE solution v_n at each iteration.

To rigorously quantify the cumulative effect of these approximations, we now derive a global L^2 error bound that separates the ideal contraction behavior from the training-induced deviations. This result justifies the robustness of our approach and provides guidance on the choice of residual tolerance during training.

Theorem 2 (Global L^2 error bound). *Under the same assumption in Proposition 3 and $\tilde{\kappa} \in (0, 1)$, we have that*

$$\|\tilde{v}_n - V\|_2 \leq C(p + \kappa^n + \tilde{\kappa}^n), \quad (9)$$

where $\tilde{\kappa}$ is from Theorem 1.

Proof. The proof immediately follows from Theorem 1 and Proposition 3 after decomposition

$$\tilde{e}_n = \delta_n + \varepsilon_n + (v_n - V).$$

□

This result confirms that the overall error between the neural approximation $\tilde{v}_n$ and the optimal value function V can be decomposed into a controllable training error and an exponentially decaying ideal iteration error. In particular, so long as the residual tolerance p_n remains uniformly bounded, the cumulative error remains stable across iterations. This theoretical guarantee forms the basis for choosing the training accuracy of the PINN at each step in practice.

In the next section, we empirically validate these theoretical insights on a range of benchmark control problems, demonstrating both the convergence behavior and the accuracy of the resulting policies.

6 EXPERIMENTS

We empirically validate our proposed PINN-based policy iteration (PINN-PI) framework, which implements a physics-informed variant of Howard’s policy iteration scheme for stochastic optimal control. Our experiments span both linear-quadratic and nonlinear benchmark systems with stochastic dynamics and compact action spaces. Through these experiments, we aim to demonstrate: (1) scalability and stability of PINN-PI in high-dimensional settings, including monotonicity of value functions Howard (1960); Kerimkulov et al. (2020), (2) its advantage over model-free baselines such as SAC (3), and its robustness.

6.1 LINEAR-QUADRATIC REGULATOR (LQR) WITH COMPACT ACTION SPACE

We begin with stochastic LQR problems in dimensions $d = m \in \{5, 10, 20\}$ under a compact control set. The compactness assumption breaks the standard Riccati structure and eliminates closed-form solutions, yet the true value function remains nearly quadratic, making this setting a useful benchmark for evaluating learning performance. To evaluate the learned policies, we report rewards averaged over 30 trajectories starting from the zero initial state.

As a model-free comparison, we train Soft Actor-Critic (SAC) Haarnoja et al. (2018) on the same problem, using identical initializations and noise realizations. Unlike PINN-PI, SAC must discover both dynamics and cost structure purely from rollouts.

Figure 1 compares our method (PINN-PI) with Soft Actor-Critic (SAC) in 5D and 10D LQR settings. As demonstrated, PINN-PI consistently achieves higher reward and smoother convergence, while SAC struggles to generalize in high dimensions due to sample inefficiency. The result with $(d, m) = (20, 20)$ is provided in Appendix B together with experimental details.

6.2 NONLINEAR BENCHMARKS WITH STOCHASTIC DYNAMICS

To evaluate performance in more realistic and nonlinear scenarios, we consider two widely used benchmark environments Brockman et al. (2016): the stochastic inverted pendulum and cartpole. Both systems are modeled as stochastic control-affine dynamics with additive Brownian noise.

Figure 2 shows the evolution of performance over training, evaluated from 10 trajectory samples with the same random initial states. PINN-PI consistently stabilizes the system faster and achieves higher reward than SAC, while strictly enforcing control constraints. Notably, in high-noise regimes, SAC exhibits oscillatory behavior due to imperfect reward shaping, whereas PINN-PI produces smoother, more stable trajectories by leveraging model information and HJB structure. The monotonicity property of policy iteration is also confirmed in both tasks.

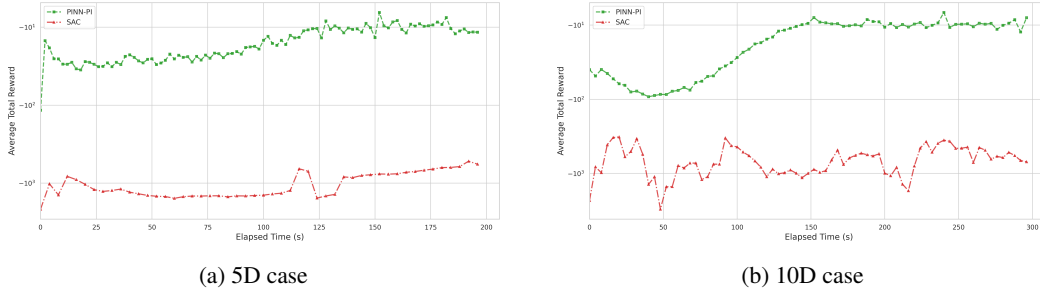


Figure 1: Comparison between PINN-PI (ours) and SAC in learning stochastic LQR problems with compact control sets. PINN-PI exhibits monotonic improvement of the value function, while SAC shows less stable learning behavior.

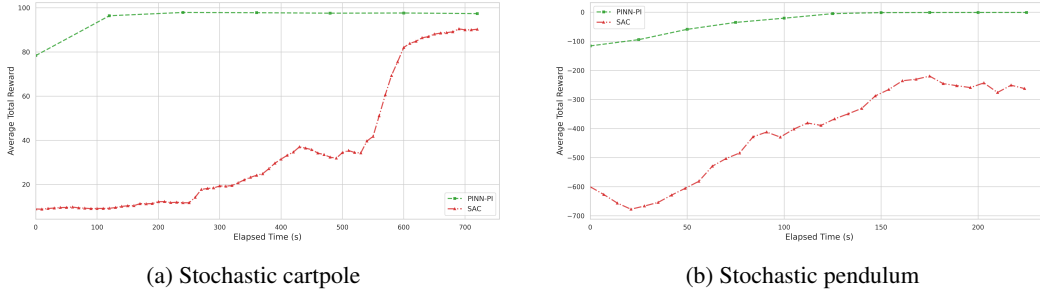


Figure 2: Comparison between PINN-PI and SAC. In both cases, PINN-PI exhibits a monotonic increase in the value function.

7 DISCUSSION

Our method extends the deterministic and affine-in-control setting of Meng et al. (2024) by establishing an L^2 -based convergence theory for general stochastic control problems with nonlinear dynamics and compact action spaces. A key feature of our method is that the total approximation error across iterations remains uniformly bounded, enabling systematic monitoring of policy quality through L^2 energy estimates.

While our approach requires full model knowledge (drift and diffusion), once the HJB equation is solved through our PINN-based framework, the resulting value function immediately yields the optimal policy for any new initial state. In this sense, our method leverages model information to produce a reusable global solution, in contrast to model-free methods that re-train for each task instance. This property highlights the scalability of our framework: solving a high-dimensional HJB not only establishes theoretical guarantees but also provides a direct mechanism for policy recovery across state space.

Several practical directions remain open, including reducing the cost of policy improvement for non-linear dynamics, developing dimension-adaptive sampling schemes, and extending the framework to settings with partially unknown dynamics. Nonetheless, the ability to recover optimal policies universally from a trained value function underscores the broader impact of combining PDE theory with physics-informed learning in high-dimensional control.

REFERENCES

- Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.
- Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.
- Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

- Ronald A Howard. Dynamic programming and markov processes. 1960.
- Yu-Jui Huang, Zhenhua Wang, and Zhou Zhou. Convergence of policy iteration for entropy-regularized stochastic control problems. *SIAM Journal on Control and Optimization*, 63(2):752–777, 2025.
- Saul D Jacka and Aleksandar Mijatović. On the policy improvement algorithm in continuous time. *Stochastics*, 89(1):348–359, 2017.
- B. Kerimkulov, D. Šiška, and Łukasz Szpruch. Exponential convergence and stability of howard’s policy improvement algorithm for controlled diffusions. *SIAM Journal on Control and Optimization*, 58(3):1314–1340, 2020. doi: 10.1137/18M1211650.
- Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton–Jacobi deep Q-learning for deterministic continuous-time systems with Lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206):1–34, 2021.
- Jae Yong Lee and Yeoneung Kim. Hamilton–jacobi based policy-iteration via deep operator learning. *Neurocomputing*, pp. 130515, 2025.
- Yiming Meng, Ruikun Zhou, Amartya Mukherjee, Maxwell Fitzsimmons, Christopher Song, and Jun Liu. Physics-informed neural network policy iteration: Algorithms, convergence, and verification. *arXiv preprint arXiv:2402.10119*, 2024.
- Amartya Mukherjee and Jun Liu. Bridging physics-informed neural networks with reinforcement learning: Hamilton-jacobi-bellman proximal policy optimization (hjbppo). *arXiv preprint arXiv:2302.00237*, 2023.
- M. L. Puterman. On the convergence of policy iteration for controlled diffusions. *Journal of Optimization Theory and Applications*, 33(1):137–144, 1981. doi: 10.1007/BF00935182.
- Martin L Puterman and Shelby L Brumelle. On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4(1):60–69, 1979.
- Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- Adithya Ramesh and Balaraman Ravindran. Physics-informed model-based reinforcement learning. In *Learning for Dynamics and Control Conference*, pp. 26–37. PMLR, 2023.
- Wenpin Tang, Hung Vinh Tran, and Yuming Paul Zhang. Policy iteration for the deterministic control problems—a viscosity approach. *SIAM Journal on Control and Optimization*, 63(1):375–401, 2025.
- Hung V Tran. *Hamilton–Jacobi equations: theory and applications*, volume 213. American Mathematical Soc., 2021.
- Hung Vinh Tran, Zhenhua Wang, and Yuming Paul Zhang. Policy iteration for exploratory Hamilton–Jacobi–Bellman equations. *Applied Mathematics & Optimization*, 91(2):50, 2025a.
- Hung Vinh Tran, Zhenhua Wang, and Yuming Paul Zhang. Policy iteration for exploratory hamilton–jacobi–bellman equations. *Applied Mathematics & Optimization*, 91(2):50, 2025b.
- Hee Jun Yang, Min Jung Kim, and Yeoneung Kim. Solving nonconvex hamilton–jacobi–isaacs equations with pinn-based policy iteration. *arXiv preprint arXiv:2507.15455*, 2025.

REPRODUCIBILITY STATEMENT

We have taken several steps to ensure the reproducibility of our results. All theoretical claims are stated with clear assumptions in Section 3 and rigorously proved in Appendix (A.1–A.3). The complete description of our proposed PINN-based policy iteration algorithm is provided in Algorithm 2 (Section 5), together with detailed error analysis and convergence guarantees (Theorems 1–2). Experimental setups, including hyperparameters, network architectures, and collocation sampling strategies, are fully described in Section 6 and Appendix B. To further facilitate reproducibility, we will release anonymous source code and scripts for reproducing all experiments (LQR, pendulum, cartpole) as supplementary material.

A APPENDIX

A.1 PROOF OF PROPOSITION 1

Since a_n takes values in A and b is uniformly bounded on $\mathbb{R}^d \times A$, we have $\|b_n\|_\infty \leq B$ and $\|\operatorname{div}_x b_n\|_\infty \leq B$. For $u, \varphi \in H^1(\mathbb{R}^d)$, we set

$$a(u, \varphi) := \frac{1}{2} \int_{\mathbb{R}^d} \sigma \sigma^\top \nabla_x u \cdot \nabla_x \varphi + \lambda \int_{\mathbb{R}^d} u \varphi - \int_{\mathbb{R}^d} b_n \cdot \nabla_x u \varphi.$$

The right-hand side is $\ell(\varphi) := \int k \varphi$. We now see the boundedness of the coercivity of a as

$$\begin{aligned} |a(u, \varphi)| &\leq \frac{\lambda}{2} \|\nabla_x u\|_2 \|\nabla_x \varphi\|_2 + \lambda \|u\|_2 \|\varphi\|_2 + B \|\nabla_x u\|_2 \|\varphi\|_2 \\ &\leq C_1 \|u\|_{H^1} \|\varphi\|_{H^1}. \end{aligned}$$

and

$$\begin{aligned} a(u, u) &= \frac{1}{2} \int_{\mathbb{R}^d} \nabla_x u^\top \sigma \sigma^\top \nabla_x u + \lambda \|u\|_2^2 - \frac{1}{2} \int_{\mathbb{R}^d} (\operatorname{div} b_n) u^2 \\ &\geq \frac{\nu}{2} \|\nabla_x u\|_2^2 + (\lambda - \frac{1}{2} B) \|u\|_2^2. \end{aligned}$$

Since $\lambda > \frac{1}{2} B$, the form is coercive.

The functional ℓ is continuous on H^1 . By the Lax–Milgram theorem a unique $v_n \in H^1(\mathbb{R}^d)$ solves $a(v_n, \varphi) = \ell(\varphi)$ for all φ .

To finish the energy estimate, we test inequality with $\varphi = v_n$, which leads to

$$\lambda \|v_n\|_2^2 + \frac{\nu}{2} \|\nabla_x v_n\|_2^2 \leq \frac{1}{2} B \|v_n\|_2^2 + (k, v_n).$$

Continuing from above, rearranging the inequality gives:

$$(\lambda - \frac{1}{2} B) \|v_n\|_2^2 + \frac{\nu}{2} \|\nabla_x v_n\|_2^2 \leq (k, v_n),$$

where $(f, g) := \int_{\mathbb{R}^d} f(x)g(x) dx$.

We now estimate the right-hand side using Cauchy–Schwarz and Young’s inequality. For any $\varepsilon > 0$, we have:

$$(k, v_n) \leq \|k\|_2 \|v_n\|_2 \leq \varepsilon \|v_n\|_2^2 + \frac{1}{4\varepsilon} \|k\|_2^2.$$

Substituting this into the inequality, we obtain:

$$(\lambda - \frac{1}{2} B - \varepsilon) \|v_n\|_2^2 + \frac{\nu}{2} \|\nabla_x v_n\|_2^2 \leq \frac{1}{4\varepsilon} \|k\|_2^2.$$

Taking $\varepsilon = \frac{1}{2}(\lambda - \frac{1}{2} B)$, which is valid because $\lambda > \frac{1}{2} B$, yields:

$$\|v_n\|_2^2 \leq \frac{1}{(\lambda - \frac{1}{2} B)^2} \|k\|_2^2, \quad \|\nabla_x v_n\|_2^2 \leq \frac{1}{\nu(\lambda - \frac{1}{2} B)} \|k\|_2^2.$$

A.2 PROOF OF PROPOSITION 2

Let $z, z' \in \mathbb{R}^d$ and denote

$$a := a^*(x, z), \quad a' := a^*(x, z').$$

By the definition of a^* as a maximizer over a convex set A , and the strong convexity of the objective function, the maximizers a, a' are unique and continuous.

The necessary condition for optimality (first-order variational inequality) yields:

$$\langle \nabla_a L(x, a) + \partial_a b(x, a)^\top z, a' - a \rangle \geq 0, \quad (10)$$

$$\langle \nabla_a L(x, a') + \partial_a b(x, a')^\top z', a - a' \rangle \geq 0. \quad (11)$$

Adding equation 10 and equation 11 gives:

$$\begin{aligned} & \langle \nabla_a L(x, a) - \nabla_a L(x, a'), a' - a \rangle \\ & + \langle [\partial_a b(x, a) - \partial_a b(x, a')]^\top z, a' - a \rangle \\ & + \langle \partial_a b(x, a')^\top (z - z'), a' - a \rangle \geq 0. \end{aligned}$$

Now use the μ_a -strong convexity of L in a :

$$\langle \nabla_a L(x, a) - \nabla_a L(x, a'), a - a' \rangle \geq \mu_a |a - a'|^2.$$

Therefore, we obtain:

$$\begin{aligned} \mu_a |a - a'|^2 & \leq |\langle [\partial_a b(x, a) - \partial_a b(x, a')]^\top z, a - a' \rangle| \\ & + |\langle \partial_a b(x, a')^\top (z - z'), a - a' \rangle|. \end{aligned}$$

Since $\partial_a b$ is $\tilde{B}$ -Lipschitz in a , the first term becomes

$$|\langle [\partial_a b(x, a) - \partial_a b(x, a')]^\top z, a - a' \rangle| \leq \tilde{B} |z| |a - a'|^2.$$

The second term is handled via

$$|\langle \partial_a b(x, a')^\top (z - z'), a - a' \rangle| \leq \tilde{B} |z - z'| |a - a'|.$$

Combine the bounds:

$$\mu_a |a - a'|^2 \leq \tilde{B} |z| |a - a'|^2 + \tilde{B} |z - z'| |a - a'|.$$

Now, subtract $\tilde{B} |z| |a - a'|^2$ from both sides:

$$(\mu_a - \tilde{B} |z|) |a - a'|^2 \leq \tilde{B} |z - z'| |a - a'|.$$

Since $\mu_a > \tilde{B} |z|$, we can divide both sides by $|a - a'|$:

$$|a - a'| \leq \frac{\tilde{B}}{\mu_a - \tilde{B} |z|} |z - z'|.$$

Therefore,

$$|a - a'| \leq \theta |z - z'|,$$

for some $\theta > 0$.

A.3 PROOF OF LEMMA 1

Recall that $\mathcal{T}[v_n, a_n] = \mathcal{T}[v_m, a_m] = 0$ and subtract two equations to achieve

$$\lambda e - \frac{1}{2} \text{Tr}(\sigma \sigma^\top D_{xx}^2 e) - b(\cdot, a_n) \cdot \nabla_x e = R,$$

where $e := v_n - v_m$ and

$$R := [b(\cdot, a_n) - b(\cdot, a_m)] \cdot \nabla_x v_m + [L(\cdot, a_n) - L(\cdot, a_m)].$$

We now test with respect to e and proceed with the L^2 coercivity argument of Proposition 1 yields

$$(\lambda - \frac{1}{2}B)\|e\|_2^2 + \frac{\nu}{2}\|\nabla_x e\|_2^2 \leq (R, e), \quad (12)$$

where $(f, g) := \int_{\mathbb{R}^d} fg \, dx$. Now the right-hand side of the inequality is estimated as

$$\begin{aligned} \|R\|_2 &\leq \tilde{B}\|a_n - a_m\|_2 \|\nabla_x v_m\|_2 + L_a\|a_n - a_m\|_2 \\ &\leq \underbrace{\theta(\tilde{B} \sup_m \|v_m\|_2 + L_a)}_{\leq C_R} \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2 \end{aligned}$$

where θ is from Proposition 2. Hence, we have that

$$\|R\|_2 \leq C_R \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2,$$

and therefore,

$$(R, e) \leq C_R \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2 \|e\|_2 \quad (13)$$

by Cauchy–Schwarz inequality.

Applying the Young’s inequality $ab \leq \frac{\varepsilon}{2}a^2 + \frac{1}{2\varepsilon}b^2$ with $\varepsilon = \frac{C_R}{2(\lambda - \frac{1}{2}B)}$, $a = \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2$, and $b = e$ in equation 13, we deduce that

$$\frac{\nu}{2}\|\nabla_x e\|_2^2 \leq \frac{C_R^2}{2(\lambda - \frac{1}{2}B)} \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2^2,$$

and hence,

$$\|\nabla_x e\|_2 \leq \sqrt{\frac{C_R^2}{\nu(\lambda - \frac{1}{2}B)}} \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2.$$

On the other hand, observing $\|e\|_2$ explicitly, we have

$$\begin{aligned} (\lambda - \frac{1}{2}B)\|e\|_2^2 &\leq \|R\|_2 \|e\|_2 \\ &\leq C_R \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2 \|e\|_2. \end{aligned}$$

Canceling $\|e\|_2$, we get

$$\|e\|_2 \leq \frac{C_R}{\lambda - \frac{1}{2}B} \|\nabla_x v_{n-1} - \nabla_x v_{m-1}\|_2.$$

B EXPERIMENTAL DETAILS AND ADDITIONAL RESULT

For all experiments, we set the discount factor $\lambda = 0.5$ and the temporal discretization step to $dt = 0.05$, which yields $\exp(-\lambda dt) \approx 0.99$.

Stochastic linear-quadratic regulator. For the stochastic LQR experiments, we considered systems of dimension $(d, m) = (5, 5), (10, 10), (20, 20)$ with the following dynamics:

$$dX_t = (AX_t + Bu_t) dt + \sigma dW_t,$$

where $\sigma = 0.1 \cdot I_d$ and $u_t \in A := \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \mathbf{u}\}$ with $\mathbf{u} = 1$. For all cases, the reward function is set to

$$L(x, u) = -x^\top Qx - u^\top Ru,$$

where $Q = 5I_d \succ 0$ and $R = I_d \succ 0$.

For the case $(d, m) = (5, 5)$, the system dynamics A and B are chosen as:

$$A = \begin{bmatrix} 0.4455 & 0.2996 & 0.4497 & 0.2813 & 0.3114 \\ 0.4317 & 0.2885 & 0.3214 & 0.2523 & 0.4193 \\ 0.1975 & 0.4295 & 0.3014 & 0.4225 & 0.2282 \\ 0.4361 & 0.4716 & 0.3812 & 0.3010 & 0.0037 \\ 0.2247 & 0.0629 & 0.3350 & 0.0424 & 0.1836 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0681 & 0.0544 & 0.0467 & 0.0152 & 0.0787 \\ 0.0970 & 0.0081 & 0.0145 & 0.0034 & 0.0984 \\ 0.0358 & 0.0833 & 0.0324 & 0.0839 & 0.0012 \\ 0.0116 & 0.0280 & 0.0056 & 0.0092 & 0.0432 \\ 0.0047 & 0.0848 & 0.0718 & 0.0977 & 0.0556 \end{bmatrix}.$$

For the case $(d, m) = (10, 10)$, the system matrices A and B are given as follows:

$$A = \begin{bmatrix} 0.0866 & 0.0191 & 0.1394 & 0.1392 & 0.1480 & 0.0241 & 0.0988 & 0.1435 & 0.1883 & 0.1923 \\ 0.1271 & 0.0900 & 0.0924 & 0.0766 & 0.0696 & 0.0466 & 0.1716 & 0.0371 & 0.1097 & 0.0946 \\ 0.0595 & 0.1656 & 0.1953 & 0.1353 & 0.1872 & 0.0587 & 0.0830 & 0.0035 & 0.0215 & 0.0740 \\ 0.1971 & 0.0808 & 0.1301 & 0.0157 & 0.1908 & 0.1505 & 0.0662 & 0.1334 & 0.1394 & 0.1951 \\ 0.0570 & 0.0419 & 0.0470 & 0.0916 & 0.1094 & 0.0640 & 0.0159 & 0.1687 & 0.1224 & 0.0294 \\ 0.1137 & 0.1033 & 0.0379 & 0.0881 & 0.1224 & 0.0139 & 0.0060 & 0.1857 & 0.0732 & 0.0989 \\ 0.1271 & 0.0414 & 0.1232 & 0.1896 & 0.1457 & 0.0997 & 0.1830 & 0.1309 & 0.0673 & 0.0855 \\ 0.0827 & 0.1076 & 0.1498 & 0.1164 & 0.0192 & 0.1888 & 0.1357 & 0.1352 & 0.1086 & 0.1959 \\ 0.1489 & 0.0223 & 0.0018 & 0.0002 & 0.1631 & 0.1272 & 0.0282 & 0.0075 & 0.0351 & 0.0478 \\ 0.0791 & 0.1215 & 0.0219 & 0.1653 & 0.0635 & 0.0230 & 0.1943 & 0.0373 & 0.0253 & 0.1129 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0728 & 0.0209 & 0.0336 & 0.0269 & 0.0197 & 0.0844 & 0.0358 & 0.0483 & 0.0033 & 0.0559 \\ 0.0459 & 0.0812 & 0.0411 & 0.0338 & 0.0581 & 0.0689 & 0.0084 & 0.0015 & 0.0198 & 0.0053 \\ 0.0136 & 0.0803 & 0.0666 & 0.0570 & 0.0277 & 0.0721 & 0.0613 & 0.0106 & 0.0115 & 0.0699 \\ 0.0848 & 0.0815 & 0.0915 & 0.0766 & 0.0683 & 0.0997 & 0.0389 & 0.0485 & 0.0630 & 0.0102 \\ 0.0858 & 0.0205 & 0.0769 & 0.0968 & 0.0722 & 0.0004 & 0.0201 & 0.0990 & 0.0836 & 0.0750 \\ 0.0168 & 0.0033 & 0.0286 & 0.0740 & 0.0314 & 0.0238 & 0.0183 & 0.0277 & 0.0889 & 0.0123 \\ 0.0795 & 0.0445 & 0.0037 & 0.0776 & 0.0038 & 0.0103 & 0.0183 & 0.0542 & 0.0722 & 0.0544 \\ 0.0399 & 0.0139 & 0.0071 & 0.0227 & 0.0556 & 0.0885 & 0.0062 & 0.0271 & 0.0382 & 0.0641 \\ 0.0142 & 0.0063 & 0.0456 & 0.0536 & 0.0993 & 0.0206 & 0.0264 & 0.0463 & 0.0695 & 0.0907 \\ 0.0301 & 0.0514 & 0.0583 & 0.0007 & 0.0900 & 0.0426 & 0.0385 & 0.0077 & 0.0110 & 0.0930 \end{bmatrix}.$$

For the case $(d, m) = (20, 20)$, we use the same A and B provided in Kim et al. (2021). The result is shown in Figure 3.

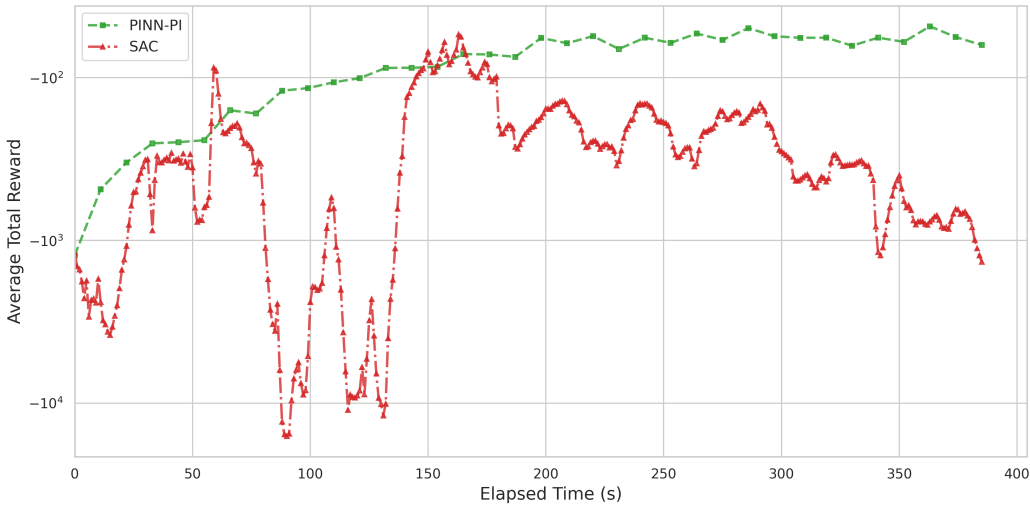


Figure 3: Stochastic LQR with $(d, m) = (20, 20)$. The green curve (PINN-PI) shows monotonic improvement in evaluation reward, whereas SAC exhibits unstable performance.

For PINN-PI, during each policy evaluation step, we train the value network for 200 gradient steps using the Adam optimizer. The number of collocation points was set to 256, and the learning rate was set to 10^{-4} for the case $(d, m) = (5, 5)$, and 10^{-5} for the rest cases.

The policy and Q network are optimized using the Adam optimizer with a learning rate of 10^{-3} .

Cartpole. For policy evaluation in PINN-PI, the value function is trained for 3,000 optimization steps with 64 collocation points using the Adam optimizer with a learning rate of 10^{-3} .

Pendulum. Similar to the previous task, for policy evaluation in PINN-PI, the value function is trained for 3,000 optimization steps with 64 collocation points using the Adam optimizer with a learning rate of 10^{-3} .

Model Architecture. All value function approximations $v_n(x; \theta)$ in this work were parameterized by multilayer perceptrons (MLPs) trained to minimize the residual loss in Eq. equation 4. Each MLP consists of 3 hidden layers, with hidden layer widths ranging between 100 and 256 depending on the task. Smooth activation functions such as $\tanh$ were used to ensure differentiability required for computing PDE residuals.