Physics-Informed Approach for Exploratory Hamilton–Jacobi–Bellman Equations via Policy Iterations

Yeongjong Kim[1*], Namkyeong Cho[2*], Minseok Kim[3], Yeoneung Kim[3†]

[1]Center for Mathematical Machine Learning and its Applications (CM2LA), Pohang University of Science and Technology
[2]Department of Finance and Big Data, Gachon University
[3]Department of Applied Artificial Intelligence, Seoul National University of Science and Technology
kim.yj@postech.ac.kr, nkcho@gachon.ac.kr, {minseokkim, yeoneung}@seoultech.ac.kr

Abstract

We propose a mesh-free policy iteration framework based on physics-informed neural networks (PINNs) for solving entropy-regularized stochastic control problems. The method iteratively alternates between soft policy evaluation and improvement using automatic differentiation and neural approximation, without relying on spatial discretization. We present a detailed error analysis that decomposes the total approximation error into three sources: iteration error, policy network error, and PDE residual error. The proposed algorithm is validated with a range of challenging control tasks, including high-dimensional linear-quadratic regulation in 5D and 10D, as well as nonlinear systems such as pendulum and cartpole problems. Numerical results confirm the scalability, accuracy, and robustness of our approach across both linear and nonlinear benchmarks.

Code —
https://github.com/tomatofromsky/pinn-spi-aaai2026
Extended version — https://arxiv.org/abs/2508.01720

1 Introduction

Solving nonlinear Hamilton–Jacobi–Bellman (HJB) equations lies at the core of stochastic optimal control. In continuous-time settings, control strategies must handle uncertainty while ensuring long-term optimality. Among various formulations, entropy-regularized control which is also known as exploratory or soft control has emerged as a powerful paradigm that augments the running cost with a Kullback–Leibler divergence penalty against a reference measure. This regularization not only induces stochasticity in control policies but also promotes robust exploration and analytical tractability (Wang, Zariphopoulou, and Zhou 2020; Ziebart et al. 2008). While entropy-regularized control in continuous time (Wang, Zariphopoulou, and Zhou 2020) is rigorously formulated, the work does not address the challenges of function approximation or high-dimensional computation. Our framework complements this theory by incorporating neural approximation and residual-driven learning in a mesh-free manner.

The resulting HJB equation is a nonlinear second-order elliptic partial differential equation (PDE), typically defined over an unbounded domain. Solving such equations numerically remains a central challenge, especially in high dimensions. Soft policy iteration (PI) (Tran, Wang, and Zhang 2025; Tang, Zhang, and Zhou 2022) alternates between policy improvement via a soft-max update and policy evaluation via solving a linear elliptic PDE. Their analysis rigorously establishes exponential convergence under two canonical regimes: bounded coefficients with small control in the diffusion, and unbounded dynamics under sufficient discounting. Their results extend classical Howard-type PI into the entropy-regularized setting with relaxed controls and randomized policies.

While the theory of soft PI is well understood, practical deployment in high-dimensional control remains difficult because each iteration requires solving a nonlinear PDE. Recent work has attempted to bypass grids with mesh-free physics-informed neural networks (PINNs), embedding the PDE residual into a variational loss (Lee and Kim 2025; Ramesh and Ravindran 2023). Yet rigorous *a priori* guarantees in the entropy-regularized setting are still missing. In the deterministic case, PINN-based policy iteration methods (Meng et al. 2024; Lee and Kim 2025) establish only L^∞ convergence on compact domains and do not incorporate entropy regularization or soft-max policies. Separately, a physics-informed model-based reinforcement learning framework (Ramesh and Ravindran 2023) constrains the learned dynamics model but does not formulate or analyze the underlying HJB. In contrast, our method targets the entropy-regularized HJB directly and derives an L^2 error analysis for a fully mesh-free PI scheme.

In this work, we propose a fully mesh-free implementation of the soft PI framework using PINNs. At each iteration, the value function is approximated by a neural network trained to minimize the residual of a linearized elliptic PDE. At the same time, the policy is updated analytically via the softmax formula. Crucially, this structure allows us to derive a quantitative L^2 energy estimate that explicitly tracks how approximation errors propagate across iterations. We decompose the total error into three interpretable sources, policy update error, PDE residual error, and policy iteration error, and provide bounds for each component. Unlike prior work, our formulation exploits the linear structure of

*These authors contributed equally.
†Corresponding author.

each policy evaluation step to rigorously control approximation quality, while remaining scalable to high-dimensional domains through mesh-free optimization. This establishes a principled and practically viable foundation for solving entropy-regularized control problems in modern learning-based settings.

To summarize, our contributions are threefold:

- We propose a fully mesh-free implementation of the entropy-regularized policy iteration method via PINNs.
- We derive an L^2-based energy estimate for the value function error under approximation of the policy distribution.
- We present numerical experiments that demonstrate applicability to high-dimensional problems.

These results illustrate the potential of combining rigorous policy iteration theory with modern operator-learning techniques to enable reliable and scalable solvers for stochastic control problems.

2 Problem Setup

2.1 Exploratory HJB Equations

Let $U \subset \mathbb{R}^m$ denote the compact control space. We consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ equipped with a d-dimensional Brownian motion $(W_t)_{t\geq 0}$. For each relaxed control $\pi = \{\pi(x,\cdot)\}_{x\in\mathbb{R}^d}$, where $\pi(x,\cdot) \in \mathbb{P}(U)$ is a probability measure on U for each x,[1] the state evolves according to the following SDE

$$\begin{cases} \mathrm{d}X_t^\pi &= \left[\int_U b(X_t^\pi, u)\pi(X_t, u)\,\mathrm{d}u\right]\,\mathrm{d}t + \sigma(X_t^\pi)\,\mathrm{d}W_t, \\ X_0 &= x \in \mathbb{R}^d. \end{cases} \tag{1}$$

This corresponds to the situation where, at each state $X_t = x \in \mathbb{R}^d$, the control $u \in U$ is chosen randomly from $\pi(X_t, u)$, and the drift coefficient is averaged accordingly.

Let $\Pi := \{\pi : \mathbb{R}^d \to \mathbb{P}(U)\}$. Given policy $\pi \in \Pi$, define the value function as

$$V^\pi(x) := \mathbb{E}_x\left[\int_0^\infty e^{-\rho t} \times \left(\int_U \left(r(X_t^\pi, u) - \lambda \ln \pi(X_t, u)\right)\pi(X_t, u)\,\mathrm{d}u\right)\,\mathrm{d}t\right]. \tag{2}$$

The optimal value function is defined as

$$V(x) := \sup_{\pi\in\Pi} V^\pi(x).$$

Under suitable conditions, the value function solves the following nonlinear second-order elliptic PDE:

$$\rho V(x) = \sup_{\pi\in\Pi} F_\pi(x, \nabla_x V(x), D_{xx}^2 V(x)) \tag{3}$$

Here, for simplicity, we denote

$$\Sigma(x) := \sigma(x)\sigma(x)^\top,$$
$$f(x,u,p) := b(x,u)\cdot p + r(x,u),$$

and

$$F_\pi(x,p,X) := \int_U (f(x,u,p) - \lambda \ln(\pi(u))\pi(u)\,\mathrm{d}u + \tfrac{1}{2}\operatorname{tr}(\Sigma(x)X),$$

for $x, p \in \mathbb{R}^d$, $u \in U$ and $X \in \mathbb{R}^{d\times d}$.

Notations Let us begin with some notations used throughout the paper. For $x \in \mathbb{R}^d$, we write $|x|$ for the Euclidean norm. We denote the Hessian of a scalar function f by $D_{xx}^2 f$. For $p \in [1,\infty)$, we write $f \in L^p(\Omega)$, if

$$\|f\|_{L^p(\Omega)} := \left(\int_\Omega |f(x)|^p\,\mathrm{d}x\right)^{1/p}.$$

We write $C_b^k(\mathbb{R}^d)$ for the space of k-times continuously differentiable functions $f : \mathbb{R}^d \to \mathbb{R}$ such that all partial derivatives up to order k are bounded. The norm is defined by

$$\|f\|_{C_b^k} := \sum_{|\alpha|\leq k} \|D^\alpha f\|_{L^\infty(\mathbb{R}^d)}.$$

In particular, $f \in C_b^2(\mathbb{R}^d)$ implies that f, its gradient $\nabla_x f$, and its Hessian $D_{xx}^2 f$ are all bounded.

To ensure well-posedness of the control problem and regularity of the value function, we impose the following structural assumptions on the system dynamics and cost functions.

Assumption 1. *We assume the following:*

(A1) *The matrix $\Sigma = \sigma\sigma^\top$ satisfies the uniform ellipticity condition $\Sigma(x) \succeq \frac{1}{C_0} I_d$.*

(A2) *The functions $b(\cdot,u)$, $r(\cdot,u)$, and $\sigma(\cdot)$ belong to $C_b^2(\mathbb{R}^d)$ uniformly in u and*

$$\|b(\cdot,u)\|_{C^2(\mathbb{R}^d)} + \|r(\cdot,u)\|_{C^2(\mathbb{R}^d)} + \|\sigma(\cdot)\|_{C^2(\mathbb{R}^d)} < C_1$$

(A3) ***(Entropy regularization)*** *The constant $\lambda > 0$ is fixed and governs the strength of the entropy term in* (2) *and* (3).

2.2 Policy Iteration Scheme

To solve the entropy-regularized HJB equation (3), we adopt a classical policy iteration (PI) strategy extended to the randomized control setting. The key idea is to alternate between evaluating the value function under a fixed policy and improving the policy by optimizing a soft Bellman operator that reflects entropy-regularized performance.

This framework is particularly attractive because, although the original HJB equation is defined over an unbounded domain, each policy evaluation step reduces to solving a linear second-order elliptic PDE under a fixed control distribution. The policy improvement step, in turn, amounts to a pointwise softmax update that admits a closed-form expression.

In Algorithm 1, we present the exact policy iteration scheme proposed in the soft PI framework (Tang, Zhang, and Zhou 2022; Tran, Wang, and Zhang 2025), which we later extend to a mesh-free, PINN-based implementation. The algorithm alternates between two steps: a policy improvement step, where a soft-optimal policy is computed

[1] We consider relaxed controls $\pi(x,\cdot) \in \mathbb{P}(U)$ that admit densities with respect to the Lebesgue measure on U, so that $\pi(x,\,\mathrm{d}u) = \pi(x,u)\,\mathrm{d}u$ and all integrals $\int_U (\cdot)\pi(x,u)\,du$ are well-defined.

Algorithm 1: Soft Policy Iteration

1: **Initialize:** Value function $v^0 : \mathbb{R}^d \to \mathbb{R}$
2: **for** $n = 1, 2, \dots$ **do**
3: **(Policy improvement step):** Given v^{n-1}, define new policy π^n:

$$\pi^n(x,u) := \frac{\exp\left[\frac{1}{\lambda} f(x,u,\nabla_x v^{n-1}(x))\right]}{\int_U \exp\left[\frac{1}{\lambda} f(x,u',\nabla_x v^{n-1}(x))\right]\,\mathrm{d}u'} \tag{4}$$

4: **(Policy evaluation step):** Solve for v^n the PDE:

$$\rho v^n(x) = F_{\pi^n}(x, \nabla_x v^n(x), D^2_{xx} v^n(x)) \tag{5}$$

5: **end for**

via the Boltzmann distribution, and a policy evaluation step, which involves solving a linear elliptic PDE under the fixed policy. The key advantage of this formulation is that the original HJB equation is decomposed into a sequence of tractable subproblems, specifically, linear PDEs with frozen coefficients and closed-form policy updates. This structure not only simplifies analysis but also lends itself naturally to function approximation methods.

We initialize the algorithm with a value function $v^0 \in C_b^2(\mathbb{R}^d)$, which may be chosen as a zero function or a smooth prior based on approximate linearization (e.g., a quadratic ansatz from LQR theory). While the initial guess affects the transient phase, the policy iteration scheme (Tran, Wang, and Zhang 2025; Tang, Zhang, and Zhou 2022) is contractive and converges exponentially to the optimal value regardless of v^0.

For completeness, we collect the structural assumptions and key convergence results (Tran, Wang, and Zhang 2025; Ma, Wang, and Zhang 2024), as our L^2-analysis in Section 3.1 builds on these facts.

To enable rigorous energy estimates in unbounded domains, we begin by imposing a structural decay assumption on the system coefficients. This ensures that beyond a large ball, the dynamics are effectively inactive, allowing us to restrict attention to a bounded computational domain.

Assumption 2. *We impose the following assumption.* $B := \sup_{u\in U} \|\nabla_x \cdot b(\cdot,u)\|_{L^\infty(\mathbb{R}^d)} < \infty$*, and there exists* $R > 0$ *such that* $b(x,u) = 0$ *uniformly in* $u \in U$ *for* $|x| \geq R$*.*

The following classical L^2 energy estimate provides coercivity for linear elliptic equations with drift and diffusion. This lemma will be repeatedly used in our error analysis, particularly when quantifying the propagation of residuals across policy evaluation steps.

Lemma 1 (Energy estimate for linear elliptic PDE with drift and diffusion)**.** *Given* $\tilde{r}, \tilde{b}, \tilde{\sigma} \in C_b^2(\mathbb{R}^d)$*, assume that (i)* $\rho > \frac{1}{2}B$*, where* $B := \|\nabla_x \cdot \tilde{b}(\cdot,u)\|_{L^\infty(\mathbb{R}^d)}$*, (ii)* $\tilde{\Sigma}(x) \succeq \frac{1}{C_0} I_d$*, (iii)* $\tilde{r} \in L^2(\mathbb{R}^d)$*. Let* v *be a unique classical solution to*

$$\rho v(x) - \tilde{r}(x) - \tilde{b}(x)\cdot\nabla_x v(x) - \tfrac{1}{2}\mathrm{tr}(\tilde{\Sigma}(x) D^2_{xx} v(x)) = 0.$$

Then, $v \in C_b^2(\mathbb{R}^d)$ *and we have the* L^2 *energy bound:*

$$\left(\rho - \tfrac{1}{2}B\right)\|v\|^2_{L^2(\mathbb{R}^d)} + \tfrac{1}{2C_0}\|\nabla_x v\|^2_{L^2(\mathbb{R}^d)} \leq \|\tilde{r}\|_{L^2(\mathbb{R}^d)} \cdot \|v\|_{L^2(\mathbb{R}^d)}.$$

Therefore,

$$\|v\|_{L^2(\mathbb{R}^d)} \leq \frac{1}{\rho - \frac{1}{2}B}\|\tilde{r}\|_{L^2(\mathbb{R}^d)},$$

$$\|\nabla_x v\|_{L^2(\mathbb{R}^d)} \leq \sqrt{\frac{C_0}{\rho - \frac{1}{2}B}}\|\tilde{r}\|_{L^2(\mathbb{R}^d)}.$$

We next establish uniform bounds on the soft-optimal policy. This property ensures that the entropy term remains well-defined and Lipschitz continuous, which will be critical in both theoretical analysis and numerical stability.

Proposition 1. *Let* $\{(v^n, \pi^n)\}_{n\geq 0}$ *be generated via Algorithm 1. Then the policy map* π^n *is uniformly bounded above and below, that is, there exist* $M > m > 0$ *such that* $\pi^n \in [m, M]$*.*

The proof is given in Appendix A of the extended version.

To control how approximation errors in the value function affect the policy, we analyze the Lipschitz continuity of the softmax map. The following result shows that the mapping from $\nabla_x v$ to the induced policy $\pi[v]$ is Lipschitz continuous in the $L^2(U)$ norm, with an explicit constant.

Lemma 2. *We fix a state* $x \in \mathbb{R}^d$ *and the control variable* $u \in U \subset \mathbb{R}^m$*. Under Assumption 1, we have that*

$$\Phi(p) := \frac{\exp\left[\frac{1}{\lambda}\left(b(x,u)\cdot p + r(x,u)\right)\right]}{\int_U \exp\left[\frac{1}{\lambda}\left(b(x,u')\cdot p + r(x,u')\right)\right]\mathrm{d}u'}$$

satisfies

$$\|\Phi(p) - \Phi(q)\|_{L^2(U)} \leq L_\Phi |p - q|, \tag{6}$$

for all $|p|, |q| \leq \tilde{L}$*. Here, the constant*

$$L_\Phi := L_\phi := \frac{2\|b\|_{L^\infty}\sup_{|p|\leq\tilde{L}}\Phi(p)\sqrt{d}|U|^{1/2}}{\lambda}. \tag{7}$$

Sketch of Proof. Differentiating the softmax map $\Phi(p)$ with respect to p reveals a Jacobian structure involving $\Phi(p)$ itself and the control vector $b(x,u)$. The gradient is uniformly bounded due to the boundedness of b, and integration over U yields the global Lipschitz constant L_Φ. A full derivation appears in Appendix B of the extended version. □

Having established the regularity properties of the soft-optimal policy, we now recall the global convergence guarantee for the exact soft policy iteration scheme (Tran, Wang, and Zhang 2025; Tang, Zhang, and Zhou 2022). This result confirms that the sequence $\{v^n\}$ generated by Algorithm 1 converges exponentially fast to the unique solution V of the entropy-regularized HJB equation. It forms the backbone of our error decomposition in Section 3.1, where we extend this result to the approximate PINN-based implementation.

Theorem 1 (Convergence of policy iteration)**.** *Suppose Assumption 1 holds. Given any compact subset $\mathcal{X} \subset \mathbb{R}^d$, the exact policy iteration sequence $\{v^n\}$ defined by* (4) *and* (5) *converges exponentially in $L^2(\mathcal{X})$ to the unique solution V of the HJB equation* (3)*. Specifically, there exists $\kappa \in (0,1)$ and $C > 0$ such that*

$$\|v^n - V\|_{L^2(\mathcal{X})} \leq C_{\mathcal{X}} \kappa^n.$$

This result confirms that under standard smoothness and ellipticity assumptions, the exact soft policy iteration method converges exponentially in L^2 norm to the unique solution of the entropy-regularized HJB equation. However, each iteration requires solving a linear second-order elliptic PDE, which becomes computationally prohibitive in high-dimensional settings.

To address this challenge, we now turn to a mesh-free implementation of policy iteration based on physics-informed neural networks (PINNs). This approach approximates the value function using a residual-minimizing neural network and updates the policy analytically using softmax. Crucially, it avoids spatial discretization entirely and scales favorably with dimension.

3 Physics-Informed Mesh-free Approach

We now describe our proposed mesh-free implementation of the policy iteration scheme using physics-informed neural networks (PINNs). At each iteration step, the value function is approximated by a neural network trained to satisfy the linear PDE, while the policy is updated analytically via softmax optimization. No spatial grid or basis discretization is used; all computations rely solely on randomly sampled collocation points.

Algorithm 2: Physics-Informed Neural Network Soft Policy Iteration (PINN-SPI)

1: **Policy evaluation (value network update):**
2: **Input:** Value function $v^0 : \mathbb{R}^d \to \mathbb{R}$, neural network $\pi(\cdot, \cdot; \omega_0)$, $v(\cdot; \theta_0)$, tolerance ε.
3: Define residual at each sample x_i as:

$$\begin{aligned}&\mathcal{R}(x_i; \theta_n, \omega_n)\\ &:= \rho v(x_i;\theta_n) - \tfrac{1}{2}\operatorname{tr}(\sigma\sigma^\top D^2_{xx} v(x_i;\theta_n))\\ &- \int_U \big[b(x_i,u)\cdot\nabla_x v(x_i;\theta_n) + r(x_i,u)\\ &\qquad - \lambda \log \pi(x_i,u;\omega_n)\big]\pi(x_i,u;\omega_n)\,\mathrm{d}u\end{aligned}$$

4: Update $\theta_n \to \theta_{n+1}$ by minimizing:

$$\mathcal{L}_{\text{value}}(\theta) = \frac{1}{N}\sum_{i=1}^N |\mathcal{R}(x_i;\theta,\omega_n)|^2$$

5: **Policy improvement (policy network update):**
6: Train $\pi(x,u;\omega_n)$ to minimize $\mathcal{L}_{\text{policy}}(\omega; v(\cdot;\theta_n))$
7: Update $\omega_n \to \omega_{n+1}$
8: **Check convergence:**
If $\frac{1}{N}\sum_{i=1}^N |v(x_i;\theta_{n+1}) - v(x_i;\theta_n)|^2 < \varepsilon$ **then stop**.

Let $\mathcal{X} \subset \mathbb{R}^d$ denote the computational domain, and let $\{x_i\}_{i=1}^N \subset \mathcal{X}$ be a set of randomly sampled training points. We fix a neural network architecture $v(\cdot;\theta) : \mathbb{R}^d \to \mathbb{R}$, $\pi(\cdot,\cdot;\omega) : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ with trainable parameters θ and ω to represent the function approximation at each step. Given v and a policy $\hat{\pi}$ induced by v, we define a loss function as

$$\mathcal{L}_{\text{policy}}(\hat{\pi};\omega) := \frac{1}{NM}\sum_{i=1}^N\sum_{j=1}^M \pi(x_i,u_j;\omega)\log\frac{\pi(x_i,u_j;\omega)}{\hat{\pi}(x_i,u_j)},$$

to minimize the KL divergence between $\hat{\pi}$ and π, where

$$\hat{\pi}(x,u_j) := \frac{\exp\left[\frac{1}{\lambda}f(x,u_j,\nabla_x v(x;\theta))\right]}{\sum_{j'=1}^M \exp\left[\frac{1}{\lambda}f(x,u_{j'},\nabla_x v(x;\theta))\right]}.$$

The overall procedure alternates between value network training and policy updates as shown in Algorithm 2. In Step 1–3, the value function is represented by a neural network v_θ and optimized to minimize the residual of the linear PDE corresponding to a fixed policy π^n. The residual includes contributions from the drift, cost, and entropy terms and is computed pointwise over collocation samples.

In Steps 4–6, the policy is updated by fitting a neural network $\pi(x,u;\omega)$ to approximate the softmax distribution defined by the current value network. This corresponds to minimizing a residual loss against the analytic policy improvement formula in Equation (4).

The process iterates until convergence in the L^2 difference of the value network across iterations. The fully mesh-free nature of the method enables high-dimensional deployment without reliance on discretization or gridding.

This approach decouples the high-dimensional HJB solution into alternating supervised learning and analytical updates. The use of PINNs enables mesh-free approximation and naturally accommodates complex geometries and unstructured data. Moreover, as shown in the next subsection, the projection error from using an approximate policy $\tilde{\pi}^n$ in place of the exact softmax policy can be quantified using an energy estimate.

3.1 Error Estimates

For computation, we may restrict attention to a bounded domain $\mathcal{X} \subset \mathbb{R}^d$ that contains the effective support of the dynamics and cost functions. Specifically, under Assumption 2, there exists $R > 0$ such that

$$b(x,u) = 0 \qquad \text{for all } |x| \geq R, \quad u \in U.$$

This allows us to define the computational domain as the ball $\mathcal{X} := B_R(0) \subset \mathbb{R}^d$ without loss of generality.

To bound the global error $\|\tilde{v}^n - V\|_{L^2(\mathcal{X})}$ we introduce three intermediate objects and decompose the difference accordingly:

- **Exact PI value** v^n: the nth value function obtained from *exact* policy iteration (Algorithm 2) and its exact soft-max policy π^n.
- **Policy–consistent value** $\hat{v}^n$: the exact solution of the linear PDE when the *approximate* policy $\tilde{\pi}^n$ is frozen, i.e. $\rho\hat{v}^n = F_{\tilde{\pi}^n}(x, D\hat{v}^n, D^2\hat{v}^n)$.
- **PINN value** $\tilde{v}^n$: the neural approximation obtained by solving the same PDE with finite data and finite capacity.

Three-term decomposition. With these objects at hand we write

$$\tilde{v}^n - V = \underbrace{\tilde{v}^n - \hat{v}^n}_{\text{(PDE error)}} + \underbrace{\hat{v}^n - v^n}_{\text{(policy error)}} + \underbrace{v^n - V}_{\text{(iteration error)}}.$$

To evaluate the impact of approximation quality on the overall error, we define two key quantities at each iteration: the policy error r_n and the residual error q_n.

Assumption 3 (Policy and residual accuracy)**.** *For each $n \geq 0$ there exist non-negative functions $r_n, q_n \in L^2(\mathbb{R}^d)$ such that*

$$r_n(x) := \|\tilde{\pi}^n(x,\cdot) - \hat{\pi}^n(x,\cdot)\|_{L^2(U)},$$
$$q_n(x) := \rho\tilde{v}^n - F_{\tilde{\pi}^n}(x, \nabla_x\tilde{v}^n, D^2_{xx}\tilde{v}^n),$$

where $\hat{\pi}$ is an exact solution satisfying the policy improvement step with $\tilde{v}^n$. Let $r := \sup_n\{\|r_n\|_{L^2(\mathcal{X})}\}$ and $q := \sup_n\{\|q_n\|_{L^2(\mathcal{X})}\}$.

The quantity $\|r_n\|_{L^2(\mathcal{X})}$ and $\|q_n\|_{L^2(\mathcal{X})}$ measure the policy improvement loss and PDE residual loss, respectively. We note here that we will derive a convergence result in terms of r_n rather than the KL-divergence between $\tilde{\pi}^n$ and $\hat{\pi}^n$ as we have that

$$\|\tilde{\pi}^n - \hat{\pi}^n\|_{L^2(U)} \lesssim (\mathrm{D}_{\mathrm{KL}}(\tilde{\pi}^n||\hat{\pi}^n))^{1/4}$$

from the Pinsker's inequality (Csiszár and Körner 2011).

Finally, we present our main theoretical result.

Theorem 2 (L^2 error)**.** *Suppose Assumptions 1- 3 hold and let $\{(\tilde{v}^n, \tilde{\pi}^n)\}_{n\geq 0}$ be learned via Algorithm 2. For $\rho > 0$ sufficiently large so that $\gamma := L_\Phi\tilde{C}_\rho \in (0,1)$, the accumulated error satisfies*

$$\|\tilde{v}^n - V\|_{L^2(\mathcal{X})} \leq C(r+q) + C_{\mathcal{X}}\kappa^n.$$

where $C_{\mathcal{X}}$ and κ are from Theorem 1, and C is a problem-dependent constant defined in the proof.

Before we prove this theorem, we first introduce a stability result on the policy evaluation. Given a policy π, we define $b^\pi(x) := \int_U b(x,u)\pi(x,u)\,\mathrm{d}u$, $f^\pi(x) := \int_U f(x,u)\pi(x,u)\,\mathrm{d}u$, and $\mathcal{H}^\pi(x) := \int_U \log\pi(x,u)\cdot\pi(x,u)\,\mathrm{d}u$.

Lemma 3 (Local policy stability on B_R)**.** *Suppose Assumption 1 and 2 hold. Fix $R > 0$ and denote $\mathcal{X} := B_R(0) = \{x \in \mathbb{R}^d : |x| \leq R\}$ with boundary $\Gamma := \partial\mathcal{X}$. Let $\pi, \tilde{\pi} \in \mathbb{P}(U)$ be two policies satisfying $\pi, \tilde{\pi} \in [m, M]$ for some $m, M > 0$, and let $v^\pi, v^{\tilde{\pi}}$ solve*

$$\rho v^\pi = b^\pi\cdot\nabla v^\pi + \tfrac{1}{2}\operatorname{tr}(\Sigma D^2 v^\pi) + f^\pi - \lambda\mathcal{H}^\pi \quad \text{in} \quad \mathcal{X},$$

and

$$\rho\tilde{v}^{\tilde{\pi}} = b^\pi\cdot\nabla\tilde{v}^{\tilde{\pi}} + \tfrac{1}{2}\operatorname{tr}(\Sigma D^2_{xx} v^{\tilde{\pi}}) + f^{\tilde{\pi}} - \lambda\mathcal{H}^{\tilde{\pi}} \quad \text{in} \quad \mathcal{X},$$

respectively. Then

$$\|\tilde{v} - v\|_{L^2(\mathcal{X})} \leq \tilde{C}_\rho\|\tilde{\pi} - \pi\|_{L^2(\mathcal{X}\times U)},$$

and

$$\|\nabla_x\tilde{v} - \nabla v\|_{L^2(\mathcal{X})} \leq \tilde{C}_\rho\|\tilde{\pi} - \pi\|_{L^2(\mathcal{X}\times U)},$$

where $\tilde{C}_\rho := \max\{\frac{\sqrt{C}}{\rho - \frac{1}{2}B}, C\sqrt{\frac{C_0}{\rho - \frac{1}{2}B}}\}$ and C is a problem-dependent constant defined in the proof.

The proof of the lemma is presented in Appendix C of the extended version. We continue to give a proof of Theorem 2.

Proof of Theorem 2. We begin by decomposing the total error via triangle inequality:

$$\|\tilde{v}^n - V\|_{L^2(\mathcal{X})} \leq \underbrace{\|\tilde{v}^n - \hat{v}^n\|_{L^2(\mathcal{X})}}_{\text{(I)}} + \underbrace{\|\hat{v}^n - v^n\|_{L^2(\mathcal{X})}}_{\text{(II)}} + \underbrace{\|v^n - V\|_{L^2(\mathcal{X})}}_{\text{(III)}},$$

where $\hat{v}^n := T[\tilde{\pi}^n]$ is the exact PDE solution under the learned policy.

By Lemma 1, we have

$$\|\tilde{v}^n - \hat{v}^n\|_{L^2(\mathcal{X})} \leq C_\rho\|q_n\|_{L^2(\mathcal{X})}.$$

To estimate (II), we invoke Lemma 3 to deduce that

$$\|\hat{v}^n - v^n\|_{L^2(\mathcal{X})} = \|T[\tilde{\pi}^n] - T[\pi^n]\|_{L^2(\mathcal{X})} \leq \tilde{C}_\rho\|\tilde{\pi}^n - \pi^n\|_{L^2(\mathcal{X}\times U)}.$$

We now split the policy gap:

$$\|\tilde{\pi}^n - \pi^n\|_{L^2(U)} \leq \underbrace{\|\tilde{\pi}^n - \hat{\pi}^n\|_{L^2(U)}}_{=r_n(x)} + \underbrace{\|\hat{\pi}^n - \pi^n\|_{L^2(U)}}_{\text{(analytic softmax mismatch)}}, \tag{8}$$

where $\hat{\pi}^n := \pi[\tilde{v}^{n-1}]$, $\pi^n := \pi[v^{n-1}]$.

The softmax map $\pi[v] = \Phi(\nabla_x v)$ is Lipschitz in gradient as demonstrated in Proposition 2:

$$\|\tilde{\pi} - \pi\|_{L^2(U)} \leq L_\Phi|\nabla_x\tilde{v}^{n-1} - \nabla_x v^{n-1}|.$$

Therefore, we have that

$$\begin{aligned}&\|\tilde{\pi}^n - \pi^n\|_{L^2(\mathcal{X}\times U)}\\ &= \|\Phi(\nabla_x\tilde{v}^{n-1}) - \Phi(\nabla_x v^{n-1})\|_{L^2(\mathcal{X}\times U)}\\ &\leq L_\Phi\|\nabla_x\tilde{v}^{n-1} - \nabla_x v^{n-1}\|_{L^2(\mathcal{X})}.\end{aligned}$$

Again split:

$$\|\nabla_x\tilde{v}^{n-1} - \nabla_x v^{n-1}\|_{L^2(\mathcal{X})} \leq \|\nabla_x\tilde{v}^{n-1} - \nabla_x\hat{v}^{n-1}\|_{L^2(\mathcal{X})} + \|\nabla_x\hat{v}^{n-1} - \nabla_x v^{n-1}\|_{L^2(\mathcal{X})}.$$

Apply the residual bound and Lemma 1,

$$\|\nabla_x\tilde{v}^{n-1} - \nabla_x\hat{v}^{n-1}\|_{L^2(\mathcal{X})} \leq C_\rho\|q_{n-1}\|_{L^2(\mathcal{X})},$$

and by Lemma 3,

$$\|\nabla_x\hat{v}^{n-1} - \nabla_x v^{n-1}\|_{L^2(\mathcal{X})} \leq \tilde{C}_\rho\|\tilde{\pi}^{n-1} - \pi^{n-1}\|_{L^2(\mathcal{X}\times U)}.$$

By iteratively applying (8) with

$$k_n := \|\tilde{\pi}^n - \pi^n\|_{L^2(\mathcal{X}\times U)},$$

we have that

$$k_n \leq r + L_\Phi C_\rho q + \underbrace{L_\Phi\tilde{C}_\rho}_{=\gamma}k_{n-1},$$

which leads to

$$k_n \leq k_0\gamma^n + \underbrace{\sum_{i=0}^{n-1}(r + L_\Phi C_\rho q)\gamma^i}_{\leq\frac{r+L_\Phi C_\rho q}{1-\gamma}}.$$

Therefore,

$$\begin{aligned}\|\hat{v}^n - v^n\|_{L^2(\mathcal{X})} &\leq \tilde{C}_\rho(r + L_\Phi(C_\rho q + k_{n-1})) \\ &\leq \tilde{C}_\rho(r + L_\Phi C_\rho q + L_\Phi k_{n-1}) \\ &\leq C(r+q).\end{aligned}$$

Putting altogether and recalling Theorem 1,

$$\|\tilde{v}^n - V\|_{L^2(\mathcal{X})} \leq C(r+q) + C_{\mathcal{X}}\kappa^n.$$

□

An important implication of this theorem is that the total approximation error does not accumulate over policy iterations. Instead, it remains uniformly bounded in terms of the residual and policy approximation errors.

In the next section, we validate these theoretical findings through numerical experiments.

4 Experiments

We evaluate the proposed PINN-based soft policy iteration (PINN-SPI) framework on a suite of entropy-regularized stochastic control problems, ranging from low-dimensional nonlinear systems to high-dimensional linear-quadratic regulators (LQR). Our goals are to (i) demonstrate scalability to high-dimensional settings, (ii) demonstrate the monotonicity property (Tran, Wang, and Zhang 2025, Corollary 5.1). All implementation details, hyperparameter configurations, and reproducibility materials are available in our public repository.

4.1 Linear-Quadratic Regulator (LQR) with Compact Action Space

We consider entropy-regularized linear-quadratic regulator (LQR) problems in 5, 10 and 20 dimensions with compact action constraints. The system dynamics are

$$\mathrm{d}X_t = (AX_t + Bu_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

where the reward is given by

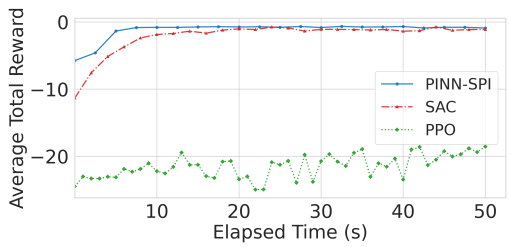
$$L(x,u) = -x^\top Q x - u^\top R u,$$

and our control set U is defined as

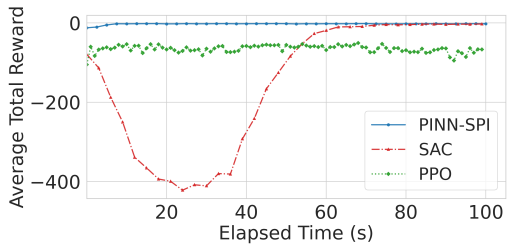
$$u \in U := \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \mathbf{u}\}.$$

Unlike classical LQR problems, where Riccati equations yield closed-form solutions, the compact control constraint requires direct numerical solution of the HJB equation. The value function, however, remains close to quadratic, making it a suitable benchmark for evaluating approximation quality and convergence.

We apply PINN-SPI and compare its performance with the model-free Soft Actor-Critic (SAC) (Haarnoja et al. 2018) and Proximal Policy Optimization (PPO) (Schulman et al. 2017) algorithm. Both methods are initialized with the same linear controller derived from the unconstrained problem. Our approach uses residual minimization with randomly sampled collocation points and softmax-based policy updates.



(a) 5D stochastic LQR with compact control set and entropy regularization.



(b) 10D stochastic LQR with compact control set and entropy regularization.

Figure 1: Comparison of PINN-SPI, SAC, and PPO on 5D and 10D stochastic LQR problems with compact action constraints.

Setup. Experiments are conducted with randomly generated matrices (A, B, Q, R), and set isotropic noise $\sigma = 0.1I_d$ where I_d denotes $d \times d$ identity matrix. We take $d = 5, 10$ and 20, and set $\mathbf{u} = 10$. Evaluation metrics include average discounted reward per iteration.

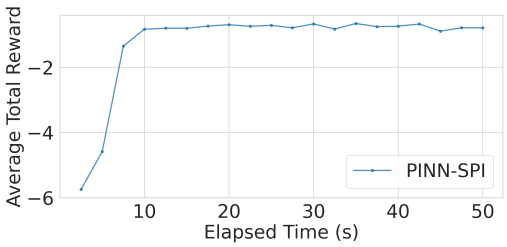
Results. Figure 1 compares PINN-SPI against SAC and PPO in 5D and 10D LQR settings. PINN-SPI consistently achieves higher reward and smoother convergence compared to other methods.

Figure 2 further illustrates the convergence behavior of PINN-SPI. The evaluation reward increases monotonically over training time, confirming the theoretical stability of policy iteration. The 10D case in particular highlights the scalability of our mesh-free implementation. An additional experiment for 20D case is provided in Appendix E of the extended version.

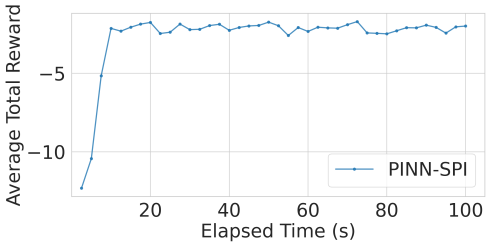
4.2 Nonlinear Benchmarks with Stochastic Dynamics

To demonstrate performance in more realistic and nonlinear settings, we evaluate our method on two standard control benchmarks: the stochastic inverted pendulum and cartpole. Each system is modeled as a stochastic control-affine system with additive Brownian noise. We adopt a standard stochasticized version of the cartpole and pendulum dynamics supported by OpenAI GYM (Brockman et al. 2016) with additive noise on state variables.

Setup. For each environment, we define entropy-regularized cost functionals and apply PINN-SPI with a

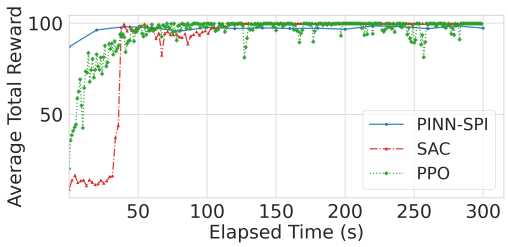


(a) 5D stochastic LQR problem.

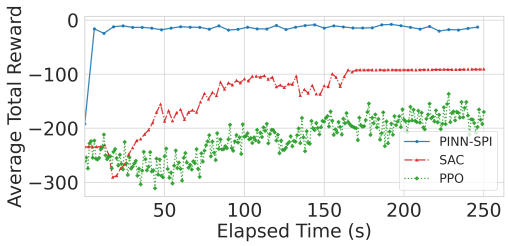


(b) 10D stochastic LQR problem.

Figure 2: Evaluation reward over training time for PINN-SPI on LQR tasks. The average total reward increases monotonically as policy iteration proceeds.



(a) Stochastic cartpole problem.



(b) Stochastic pendulum problem.

Figure 3: Comparison between PINN-SPI, SAC, and PPO on the cartpole and pendulum problems.

fixed noise level $\sigma = 0.1I_d$. We use neural networks for both value approximation and policy extraction, trained via policy iteration with residual loss minimization.

Result. Figure 3 shows the performance over elapsed time. PINN-SPI achieves faster stabilization and higher reward than SAC and PPO, while maintaining constraint satisfaction.

5 Discussion

Our proposed PINN-based soft policy iteration (PINN-SPI) framework provides a structured and scalable method for solving entropy-regularized stochastic control problems. Compared to classical policy iteration, the entropy-regularized formulation enables several theoretical and practical advantages, particularly when combined with mesh-free residual minimization. Our method offers several key advantages.

Soft updates enable stability. In standard policy iteration (e.g., Howard's method (Howard 1960; Kerimkulov, Šiška, and Szpruch 2020)), the policy improvement step requires a pointwise maximization, which can result in discontinuous or unstable policies in nonlinear or high-dimensional systems. By contrast, the softmax-based update in exploratory control is differentiable and admits explicit L^2-Lipschitz continuity with respect to the value function gradient. This structure enables more stable policy improvement and facilitates gradient-based learning.

Systematic error decomposition. The policy evaluation step in our method solves a linear PDE with frozen coefficients. This linearity allows a rigorous L^2 error decomposition into three sources: iteration error, residual error, and policy approximation error. Theorem 2 establishes that total approximation error remains uniformly bounded and does not accumulate across iterations, which is a critical stability guarantee for learning-based control methods.

Scalability to nonlinear and high-dimensional settings. Our framework applies to general nonlinear stochastic systems with compact control constraints. By using physics-informed neural networks (PINNs) and mesh-free residual minimization, we avoid spatial discretization and enable tractable approximation even in high dimensions.

We now address some limitations and future directions.

Policy differentiability assumption. Our theoretical results rely on the assumption that policies are differentiable functions of the value gradient (e.g., via softmax). This excludes problems with non-smooth or bang-bang optimal policies, or discrete action spaces, where such structure may not exist.

Model assumptions. Our framework assumes full knowledge of system dynamics (drift and diffusion). Extension to model-uncertain settings or learning dynamics jointly with control remains a promising but challenging direction.

Summary. The proposed PINN-SPI method bridges classical policy iteration theory and modern neural approximation by leveraging the analytic structure of entropy-regularized control. The result is a mesh-free, provably convergent algorithm that performs well across a broad range of benchmarks. Future work will aim to relax model assumptions, improve policy training stability, and extend to partially observed or data-driven settings.

Acknowledgements

This work was supported by Seoul National University of Science and Technology. Yeoneung Kim is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00219980, RS-2023-00211503). Namkyeong Cho was supported by the Gachon University research fund of 2025 (GCU-202502800001). The authors would like to thank Professor Hung Vinh Tran (University of Wisconsin–Madison) for his insightful suggestions and valuable guidance in developing and refining the ideas of this work.

References

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI GYM. *arXiv preprint arXiv:1606.01540*.

Csiszár, I.; and Körner, J. 2011. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press.

Evans, L. C. 2022. *Partial differential equations*, volume 19. American mathematical society.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. Pmlr.

Howard, R. A. 1960. Dynamic programming and Markov processes.

Kerimkulov, B.; Šiška, D.; and Szpruch, Ł. 2020. Exponential Convergence and Stability of Howard's Policy Improvement Algorithm for Controlled Diffusions. *SIAM Journal on Control and Optimization*, 58(3): 1314–1340.

Lee, J. Y.; and Kim, Y. 2025. Hamilton–Jacobi based policy-iteration via deep operator learning. *Neurocomputing*, 130515.

Ma, J.; Wang, G.; and Zhang, J. 2024. Convergence analysis for entropy-regularized control problems: A probabilistic approach. *arXiv preprint arXiv:2406.10959*.

Meng, Y.; Zhou, R.; Mukherjee, A.; Fitzsimmons, M.; Song, C.; and Liu, J. 2024. Physics-informed neural network policy iteration: Algorithms, convergence, and verification. *arXiv preprint arXiv:2402.10119*.

Ramesh, A.; and Ravindran, B. 2023. Physics-informed model-based reinforcement learning. In *Learning for Dynamics and Control Conference*, 26–37. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Tang, W.; Zhang, Y. P.; and Zhou, X. Y. 2022. Exploratory HJB equations and their convergence. *SIAM Journal on Control and Optimization*, 60(6): 3191–3216.

Tran, H. V.; Wang, Z.; and Zhang, Y. P. 2025. Policy iteration for exploratory Hamilton–Jacobi–Bellman equations. *Applied Mathematics & Optimization*, 91(2): 50.

Wang, H.; Zariphopoulou, T.; and Zhou, X. Y. 2020. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198): 1–34.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.