Physics-Informed Optimal Control by Control-Only Supervision with Error Guarantees on Value and Policy

Zihua Wang[1,2,3] **Xiaopei Jiao**[4*] **Yunfeng Cai**[1]
[1]Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing, China
[2]State Key Laboratory of Mathematical Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Academy for Multidisciplinary Studies, Capital Normal University, Beijing, China
wangzihua@bimsa.cn, jiaoxiaopei@cnu.edu.cn, caiyunfeng@bimsa.cn

Abstract

We propose a physics-informed method for solving high-dimensional optimal control problems that requires only control labels, not value function labels. A neural network parameterizes the value function, and the control policy is derived from its gradient. The network is trained by minimizing a weighted sum of the standard Hamilton-Jacobi-Bellman (HJB) residual and the mismatch between the networks control output and control labels, which are precomputed by a state-dependent Riccati equation (SDRE) solver. Theoretically, we establish error bounds for the value function and control policy through the viscous HJB approach, full gradient estimate, and local projected gradient estimate. Numerical experiments show that the learned controller closely reproduces the SDRE reference policy while achieving substantial inference speedup and favorable closed-loop performance. Additional experiments demonstrate robustness to label noise, distribution shifts, and moderate dynamics perturbations.

1 Introduction

The optimal control problem (OCP) provides a systematic framework for determining control laws that minimize a specified performance criterion while satisfying system dynamics and constraints. By formulating control objectives as mathematical optimization problems, this approach enables the design of controllers that achieve desired behaviors such as trajectory tracking [1], disturbance rejection [2], or minimal energy consumption [3] in an optimal sense. The resulting control policies balance competing objectives, such as speed of response and control effort, and are fundamental to applications ranging from aerospace and robotics to economics [4] and biomedical engineering [5]. The central approach of this theory is the Hamilton-Jacobi-Bellman (HJB) equation, which characterizes the optimal cost-to-go and serves as the foundation for both analytical solutions and numerical approximations in modern control design. Depending on the finiteness of the horizon, the value function satisfies the finite-horizon (FH) or steady-state (infinite-horizon) HJB, characterizing a necessary condition of optimality. The major difference is that the FH-HJB is a time-dependent nonlinear equation involving three unknown terms. There are several extensions of the HJB equation (HJBE) under different settings. Wang et al. [6] introduced the HJBE for an explorative reward function. Wilzer et al. [7] proposed the distributional HJBE.

*Corresponding author. Email: jiaoxiaopei@cnu.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

For general nonlinear OCP, the HJBE does not always have a closed-form explicit solution. For the special case of a continuous-time (CT) linear quadratic regulator (LQR), the HJBE can be reduced to an algebraic Riccati equation (ARE), as a matrix-valued equation. For complex scenarios and nonlinear systems, iterative LQR [8] was designed based on local linearization. For those system which drift term can be decomposed into semi-linear form, state-dependent Riccati equation (SDRE) [9] can provide a reliable suboptimal feedback control maintaining computational efficiency and stabilizing nonlinear dynamics. Several deep neural network (DNN) based nonlinear control algorithms have emerged in recent years. Lin et al. [10] exploited FH approximate dynamic programming (FH-ADP) to solve CT OCP by deriving an explicit time derivative. Nakamura-Zimmerer et al. [11] designed a pure data-driven adaptive framework to solve high-dimensional HJB by utilizing the Pontryagin Minimum Principle (PMP). Albi et al. [12] proposed a gradient-augmented supervised learning-based algorithm to deal with optimal feedback laws using SDRE. Lee et al. [13] utilized deep operator learning to accelerate Hamilton-Jacobi-based policy iteration.

The development of scientific machine learning has led to numerous methods that effectively address complex problems. Physics-Informed Neural Networks (PINNs) were introduced by Raissi et al. [14] for solving PDEs, incorporating physical laws as optimization constraints to enable unsupervised training. As a mesh-free technique, PINNs offer flexibility in handling complex boundaries and solving high-dimensional problems at arbitrary collocation points. Subsequent enhancements have focused on improving both accuracy and efficiency for the challenging benchmark problems [15, 16]. There has been a great interest in the Physics-Informed optimal control strategy. Fotiadis et al. [17] regarded the static HJBE as a steady state of an FH-OCP, which can be solved by PINNs as a time-dependent HJBE. Shilova et al. [18] proposed an ε-scheduling iteration to solve the viscosity HJBE by utilizing PINNs. Li et al. proposed an adaptive weighted PINN to solve Euler-Lagrange systems in optimal control [19]. Banerjee et al. present a survey on physics-informed reinforcement learning (RL) [20] to clarify how physical priors are integrated in RL. Neural scalar fields also connect PDE-based learning with motion planning. NTFields [21] learns an arrival-time field governed by the Eikonal equation and constructs paths by following its gradients.

In this work, we propose a novel physics-informed method for solving high-dimensional infinite-horizon quadratic cost regulation problems. A neural network parametrizing the value function is trained by integrating the HJBE residual and a control policy label mismatch, using SDRE as a reference. To enhance accuracy and stability, the network architecture enforces zero boundary conditions and positivity. We test the method on high-dimensional nonlinear examples: the 40D CuckerSmale model and the 51D Allen-Cahn equation. Numerically, the learned value predictions closely track the finite-horizon cost-to-go of the SDRE reference controller along the tested trajectories, providing evidence of value–cost consistency. Moreover, the NN controller achieves about 20–30$\times$ faster execution than SDRE. Theoretically, using the viscosity solution of HJB as an intermediate, we identify factors affecting value approximation error, including residual and viscous errors. To further characterize policy error, we derive a Liouville-type inequality for a weighted energy, showing that the weighted policy error is bounded by the weighted residual. The novelty of this paper is summarized as follows.

- The loss function consists of the HJB residual and the reference suboptimal control policy, without any requirement for a reference value function.
- This approach can handle high-dimensional problems with reliable accuracy, effectively overcoming the heavy computational complexity of traditional SDRE.
- Considering the mismatch between the approximated value function and cost-to-go, our method achieves lower mismatch than traditional SDRE, indicating improved value–cost consistency.
- We provide rigorous error analysis for the approximated value function by the NN and the control policy.

The remainder of the paper is organized as follows. Section 2 presents basic concepts on optimal control and the infinite horizon HJBE. In Section 3, we shall introduce a detailed methodology. Section 4 provides a theoretical guarantee about the approximation error of the value function and control policy. Section 5 will show numerical examples to demonstrate the effectiveness and accuracy of the proposed algorithm. We conclude the whole paper in Section 6. The appendix can be found to review the detailed proofs and supplementary materials.

2 Preliminaries

In this section, we introduce the basic formulations for infinite-horizon optimal control. Our goal is to minimize the following cost function, which depends on the control input $u(t)$ for $t \in [0, \infty)$:

$$\min_{u(\cdot)\in\mathbb{U}} J(u(\cdot), x_0) := \int_0^\infty e^{-\lambda s}\, \ell(x(s), u(s))\, \mathrm{d}s, \tag{1}$$

where state dynamics $\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t)) + B(x(t))u(t), x(0) = x_0$. Running cost $\ell(x, u) := x^\top Q x + u^\top R u$. The state vector $x(t) = [x_1(t), \ldots, x_n(t)]^\top$ belongs to a bounded domain $\Omega \subset \mathbb{R}^n$. The control signal $u(\cdot)$ takes values in the control space $\mathbb{U} \subset \mathbb{R}^m$, and the admissible control set $\mathbb{U}$ consists of measurable functions mapping $[0, \infty)$ to $\mathbb{U}$. The exponential discount factor $e^{-\lambda s}$ with $\lambda \geq 0$ ensures convergence of the infinite-horizon integral. The matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are symmetric positive definite. The drift dynamics $f : \mathbb{R}^n \to \mathbb{R}^n$ and the control coefficient $B : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are assumed differentiable.

The value function is defined as $V(x) := \min_{u(\cdot)\in\mathbb{U}} J(u(\cdot), x)$ and the Hamiltonian operator is given by

$$\mathcal{H}(x, u, \nabla V) := \nabla V(x)^\top \big(f(x) + B(x)u\big) + \ell(x, u). \tag{2}$$

When the control space is unconstrained, i.e., $\mathbb{U} = \mathbb{R}^m$, the optimal control policy admits a closed-form expression:

$$u(x, \nabla V) = -\frac{1}{2} R^{-1} B(x)^\top \nabla V(x). \tag{3}$$

The discounted HJBE reads

$$\lambda V(x) - \min_{u\in\mathbb{U}} \mathcal{H}(x, u, \nabla V) = 0,$$

and substituting the optimal policy (3) yields the HJBE satisfied by the value function:

$$\begin{cases} \lambda V(x) - x^\top Q x - \nabla V(x)^\top f(x) + \dfrac{1}{4}\nabla V(x)^\top B(x) R^{-1} B(x)^\top \nabla V(x) = 0, \quad x \in \Omega, \\ V(0) = 0. \end{cases} \tag{4}$$

When the discount factor vanishes, $\lambda = 0$, equation (4) reduces to the static HJBE involving only the gradient ∇V. Once $V(x)$ is determined, the optimal feedback law is given by (3).

An alternative description of the system dynamics is the semi-linear form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = A(x)x + B(x)u(t), \tag{5}$$

where $A(x)$ is a state-dependent matrix. This structure enables the SDRE approach, which provides a suboptimal feedback law by solving the ARE at each state x:

$$A(x)^\top P(x) + P(x)A(x) - P(x)B(x)R^{-1}B(x)^\top P(x) + Q - \lambda P(x) = 0. \tag{6}$$

Analogously to the linear quadratic regulator, the SDRE-based approximation of the value function is constructed as

$$V_S(x) = x^\top P(x) x, \tag{7}$$

where $P(x)$ satisfies (6) and is positive semi-definite. The corresponding exact feedback control is

$$u_{V_S}(x) := -\frac{1}{2} R^{-1} B(x)^\top \nabla V_S(x) = -R^{-1} B(x)^\top \left[P(x)x + \frac{1}{2}\varphi(x) \right],$$

with the components $\varphi_i(x) = x^\top \frac{\partial P(x)}{\partial x_i} x, 1 \leq i \leq n$. The standard SDRE feedback used to generate the control labels is

$$u_S(x) := -R^{-1} B(x)^\top P(x) x. \tag{8}$$

This feedback omits the derivatives of $P(x)$.

3 Methodology

We adopt a PINN approach for high-dimensional nonlinear optimal control problems. The value function $V(x)$ is approximated by a neural network. This approach is motivated by several key advantages: First, learning a scalar-valued $V(x)$ aligns with dynamic programming principles, and the feedback policy is derived from the value gradient; Second, approximating a scalar field mitigates the curse of dimensionality more effectively than learning a vector field $\nabla V(x)$, improving sample efficiency; Third, the gradient $\nabla V(x)$ required for the control law is computed accurately and efficiently via automatic differentiation, enabling fast online evaluation; Finally, the framework incorporates system physics by directly embedding the HJB residual into the loss function.

The subsequent subsections detail the key components of this method.

3.1 Data Collection

We construct the dataset $\mathcal{D}$ by simulating short trajectories under the SDRE controller. First, N_{traj} initial states $\{x_0^{(j)}\}_{j=1}^{N_{\text{traj}}}$ are drawn independently from a chosen distribution over the region of interest. For each initial state, we iterate the following procedure for N_T steps: at step k ($0 \le k < N_T$), the state is $x_k^{(j)}$; we compute the SDRE-based control $u_k^{(j)} = u_S(x_k^{(j)})$ via (8); then the system is advanced by a time increment Δt to obtain the next state $x_{k+1}^{(j)}$, using the dynamics (5) under control $u_k^{(j)}$. The integration is performed with a standard numerical solver (e.g., RungeKutta RK45). In the end, we collect all state-control pairs:

$$\mathcal{D} = \big\{(x_i, u_i)\big\}_{i=1}^{N}, \qquad N = N_{\text{traj}} \times N_T,$$

where the index i runs over all trajectories and time steps. Under such a sampling strategy, the training process will be implemented more on the state region visited by reference trajectories.

3.2 Physics-informed optimal feedback algorithm

Let $V_\theta(x)$ denote the neural network approximation of the value function with parameters $\theta = \{W_i, b_i\}$ including weights and biases. The optimal control policy is approximated from the gradient of V_θ as $u(x, \nabla V_\theta)$, where u is defined in (3). A key limitation of the classical SDRE approach is the heavy computational burden of solving a Riccati equation for $P(x)$ at every state encountered along a trajectory. Our framework alleviates this burden by using the SDRE control only as a soft label during training, while simultaneously enforcing the HJBE as a physical constraint. The loss function comprises two terms: the HJB term and the control policy term.

HJB term. This term enforces the HJB optimality condition by penalizing its residual:

$$\mathcal{L}_{\text{HJB}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\big|\lambda V_\theta(x_i) - \mathcal{H}\big(x_i, u_\theta(x_i), \nabla V_\theta(x_i)\big)\big|^2, \tag{9}$$

where $\mathcal{H}$ is the Hamiltonian defined in (2), $u_\theta(x_i) = u(x_i, \nabla V_\theta(x_i))$, and u is defined in (3).

Control policy term. This term encourages the network output to match the near-optimal SDRE control signals from the dataset:

$$\mathcal{L}_c(\theta) = \frac{1}{N}\sum_{i=1}^{N}\big|u_S(x_i) - u_\theta(x_i)\big|^2, \tag{10}$$

where u_S is given by (8).

The total training loss is a weighted sum of the two terms:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{HJB}}(\theta) + \gamma\mathcal{L}_c(\theta), \tag{11}$$

where $\gamma > 0$ is a tuning parameter that balances the influence of the HJB optimality condition and the data fidelity provided by the SDRE labels.

Network parameterization. To improve numerical conditioning, we normalize the inputs using full-pool state statistics. Let $\mu \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}^d$ denote the coordinate-wise empirical mean and standard deviation computed from the collected state dataset before the pointwise training/held-out split. The held-out states contribute to these preprocessing statistics; closed-loop performance is evaluated using separate test initial conditions. We define the normalized state as $\tilde{x} = \frac{x-\mu}{\sigma}, \tilde{x}_0 = \frac{-\mu}{\sigma}$. We then construct $z_\theta(x) = \mathcal{NN}_\theta(\tilde{x}) - \mathcal{NN}_\theta(\tilde{x}_0)$, where $\mathcal{NN}_\theta(\cdot)$ is a standard feedforward neural network parameterized by θ. The value function is given by

$$V_\theta(x) = z_\theta(x) + \big(\mathrm{ReLU}(z_\theta(x)) - z_\theta(x)\big)_{\mathrm{sg}},$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. By construction, $V_\theta(0) = 0$ and $V_\theta(x) \geq 0$ for all x since $V_\theta(x) = \mathrm{ReLU}(z_\theta(x)) \geq 0$ and $\nabla_\theta V_\theta(x) = \nabla_\theta z_\theta(x)$. Two operations, i.e., ReLU and the stop-gradient operation, ensure positivity and invariant gradient calculation, respectively. In practice, the SDRE-guided training reliably drives z_θ into the non-negative region, making the stop-gradient a harmless safeguard.

Training procedure. The network is trained by minimizing the total loss (11) over the dataset $\mathcal{D}$ using a stochastic gradient-based optimizer. The training procedure is summarized in Algorithm 1.

Algorithm 1: SDRE-guided PINN for optimal control

Input: domain $\Omega \subset \mathbb{R}^n$, system parameters f, A, B, Q, R, discount factor λ, weight γ, dataset $\mathcal{D} = \{(x_i, u_i)\}_{i=1}^N$.
Initialize neural network parameters θ.
for *epoch* $k = 1$ *to MAX_ITER* **do**
  for *mini-batch* $S \subset \mathcal{D}$ **do**
    Let $S_x = \{x_i : (x_i, u_i) \in S\}$.
    Compute HJB term $\mathcal{L}_{\mathrm{HJB}}$ on S_x via (9).
    Compute control policy term $\mathcal{L}_c$ on S_x via (10).
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{HJB}} + \gamma \mathcal{L}_c$.
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$.
Return Approximate value function V_θ.

4 Error analysis

In this section, we shall focus on the estimate of the error between the NN solution V_θ and the exact value function V. From a PDE perspective, it is relatively hard to directly consider V, which might be non-smooth and have multiple solutions. A good choice here we considered is to introduce a viscosity solution V^ε as an intermediate. In the first part, we show that under certain bounded residual conditions, the error itself $|V - V_\theta|$ can be arbitrarily close to zero. In the second part, the gradient error $|\nabla V - \nabla V_\theta|$ will be considered by applying PDE-related estimation. This term quantifies a certain metric to compare the NN-approximated policy and the optimal policy.

4.1 Generalization error of value function

For a smooth function U, define the viscous HJB residual by

$$r_\varepsilon[U](x) := \lambda U(x) - \min_{u \in \mathbb{U}} \mathcal{H}(x, u, \nabla U) - \varepsilon \Delta U(x).$$

For the SDRE value function approximation (7), there is an explicit expression for the residual.

$$\begin{aligned} r_\varepsilon[V_S](x) = & -\varphi(x)^\top [A(x) - B(x)R^{-1}B(x)^\top P(x)]x + \frac{1}{4}\varphi(x)^\top B(x)R^{-1}B(x)^\top \varphi(x) \\ & - \varepsilon \left[4(\nabla \cdot P(x))^\top x + 2\,\mathrm{tr}(P(x)) + x^\top \Delta P(x) x\right]. \end{aligned}$$

This expression follows by differentiating (7) and using (6). The notations $[\varphi(x)]_i = x^\top \partial_{x_i} P(x) x$, $(\nabla \cdot P(x))_j := \sum_{k=1}^n \frac{\partial P_{kj}(x)}{\partial x_k}$ and $(\Delta P(x))_{ij} := \sum_{k=1}^n \frac{\partial^2 P(x)}{\partial x_k^2}$.

For neural network approximation V_θ, similarly we have corresponding residual $r_\varepsilon[V_\theta](x)$. Rewriting the equation, it can be seen that V_θ satisfies the viscosity solution of the HJB with an extra residual,

$$\lambda V_\theta(x) - \min_{u \in \mathbb{U}} \{\mathcal{H}(x, u, \nabla V_\theta) + r_\varepsilon[V_\theta](x)\} - \varepsilon \Delta V_\theta(x) = 0. \tag{12}$$

Notice that this is a second-order equation with the term $-\varepsilon\Delta V_\theta$, which can be interpreted in a stochastic optimal control framework hereafter. Let us define a modified running cost $\tilde{\ell}_\varepsilon(x,u) = x^\top Qx + u^\top Ru + r_\varepsilon[V_\theta](x)$. With this choice, we get the second-order modified viscousity form of the HJBE accordingly.

$$\lambda V_\theta(x) - \min_{u\in\mathbb{U}}\left\{\nabla V_\theta(x)^\top(f(x)+B(x)u) + \tilde{\ell}_\varepsilon(x,u)\right\} - \varepsilon\Delta V_\theta(x) = 0. \tag{13}$$

The solution of the above equation corresponds to the following stochastic optimal control problem with discount factor:

$$\begin{cases} \min\limits_u \tilde{J}_\varepsilon(x;u) = \mathbb{E}\left[\int_0^\infty e^{-\lambda t}\tilde{\ell}_\varepsilon(X_t,u_t)dt\right], \\ \quad \text{s.t.} \quad dX_t = [f(X_t)+B(X_t)u_t]dt + \sqrt{2\varepsilon}dW_t. \end{cases}$$

In the following, we show that $V_\theta(x)$ can be equivalently described by the stochastic dynamic programming principle (SDPP). The detailed proof can be found in section A.3.

Lemma 4.1. *Let $V_\theta \in C^2(\mathbb{R}^n)$ satisfy the modified viscous HJBE* (13)*. Assume that the controlled stochastic dynamics are well-posed, the relevant expectations are finite, and the Hamiltonian-minimizing feedback is admissible. Then, for any $T > 0$,*

$$V_\theta(x) = \inf_{u\in\mathcal{U}} \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_t)\,dt + e^{-\lambda T}V_\theta(X_T)\right].$$

In the following result, we shall provide an upper bound for the error $|V_\theta - V^\varepsilon|$. A detailed proof can be found in the section A.4.

Theorem 4.1. *Let u^* be an optimal control for V^ε, and let $\widetilde{u}$ be an optimal control for the modified problem associated with V_θ. Denote by X_t^* and $\widetilde{X}_t$ the corresponding state processes. Assume furthermore that $\lim_{T\to\infty}\mathbb{E}_x\left[e^{-\lambda T}\left|V_\theta(X_T^*) - V^\varepsilon(X_T^*)\right|\right] = 0$ and $\lim_{T\to\infty}\mathbb{E}_x\left[e^{-\lambda T}\left|V_\theta(\widetilde{X}_T) - V^\varepsilon(\widetilde{X}_T)\right|\right] = 0$. Then*

$$|V_\theta(x) - V^\varepsilon(x)| \le \max\left\{\mathbb{E}_x\int_0^\infty e^{-\lambda t}|r_\varepsilon(X_t^*)|\,dt,\ \mathbb{E}_x\int_0^\infty e^{-\lambda t}|r_\varepsilon(\widetilde{X}_t)|\,dt\right\}.$$

In particular, if $\lambda > 0$ and $r_\varepsilon \in L^\infty$, then $|V_\theta(x) - V^\varepsilon(x)| \le \frac{1}{\lambda}\|r_\varepsilon\|_{L^\infty}$.

Corollary 4.1. *Let V be the optimal value function defined by the standard HJBE. We have the following*

(i) If $\lambda > 0$ and $r_\varepsilon \in L^\infty$, the value error $|V_\theta(x) - V(x)| \le \frac{1}{\lambda}\|r_\varepsilon\|_{L^\infty} + C\sqrt{\varepsilon}$;

(ii) If $\lambda = 0$, then $|V_\theta(x) - V(x)| \le \max\left\{\mathbb{E}_x\int_0^\infty |r_\varepsilon(X_t^*)|\,dt,\ \mathbb{E}_x\int_0^\infty |r_\varepsilon(\widetilde{X}_t)|\,dt\right\} + C\sqrt{\varepsilon}$.

The proof can be easily obtained by combining the result of Theorem 4.1 and the well-known estimate (17) together. On the right-hand side, the first term reflects the mismatch between NN V_θ and the viscosity solution V^ε. The second term gives the convergence rate when considering a vanishing viscous sequence $V^\varepsilon \to V$ as $\varepsilon \to 0$.

4.2 Gradient estimate and policy error

In deep learning-based optimal control, quantifying the policy approximation error is essential. We analyze the viscous HJBE and establish two complementary estimates. The first controls the full gradient error $\nabla V^\varepsilon - \nabla V_\theta$ through a weighted energy estimate under specific structural assumptions on the control matrix and drift. A Liouville-type inequality and contradiction argument yield the corresponding policy-error bound in Theorem 4.2.

The second estimate directly controls the projected gradient error $B^\top(\nabla V^\varepsilon - \nabla V_\theta)$, which determines the feedback policy. By exploiting the nonlinear structure of the HJBE, Theorem 4.3 establishes a local, unweighted policy-error bound without requiring full row rank of B or a gradient-form drift, making it applicable to more general underactuated systems. Detailed proofs are provided in Section A.6 and Section A.7.

Theorem 4.2. *Assume that the control matrix $B(x)$ has full rank and $B(x)R^{-1}B(x)^\top = I$ is an identity matrix. Assume the drift function has gradient form $f = \nabla\phi$. Denote $\Phi(x) := \frac{1}{2}V^\varepsilon(x) - \phi(x)$ and $\rho(x) := \exp\left(-\frac{\Phi(x)}{\varepsilon}\right)$. Let total residual error $R_\varepsilon(x) := r_\varepsilon[V_\theta](x) + \lambda w(x)$. Denote $M(r) := \int_{B_r}\rho$ and $H(r) := \int_{B_r}\rho|R_\varepsilon|$. Assume*

(i) $M'(r) = \int_{\partial B_r}\rho \le C$; *(ii)* $\left\|R^{-1}B(x)^\top\right\| \le K_r, x \in B_r$; *(iii)* $\int_{B_r}\rho|\nabla(V^\varepsilon - V)|^2 \le \eta_\varepsilon(r)$.

If the global weighted residual error satisfies $H_\infty := \int_{\mathbb{R}^n}\rho|R_\varepsilon| \le \frac{\delta}{8}$, then for every $r > 0$, We shall have the policy estimate:

$$\int_{B_r}\rho|u_\theta - u|^2 \le \frac{K_r^2}{2}(\delta + \eta_\varepsilon(r)).$$

We provide an intuitive connection between the training process of NN and the above upper bound. Let $S_N = \{x_i\}_{i=1}^N$ be the training points with quadrature weights $\{\omega_i\}_{i=1}^N$. Define the empirical weighted source loss by $\mathcal{E}_T^R := \sum_{i=1}^N \omega_i\rho(x_i)\,|r[V_\theta](x_i)|$, where $r[V_\theta] = r_\varepsilon[V_\theta] + \varepsilon\Delta V_\theta$. Assume that the quadrature rule satisfies $\left|\int_{\mathbb{R}^n}\rho|r[V_\theta](x)| - \sum_{i=1}^N \omega_i\rho(x_i)|r[V_\theta](x_i)|\right| \le C_{\mathrm{quad}}N^{-\alpha}$. Then we conclude the upper bound for the weighted residual:

$$H_\infty \le \underbrace{\mathcal{E}_T^R}_{\text{training error}} + \underbrace{C_{\mathrm{quad}}N^{-\alpha}}_{\text{quadrature error}} + \underbrace{\mathcal{O}(\varepsilon)}_{\text{viscosity error}} + \underbrace{|\lambda|\cdot\|w\|_{L^1_\rho(\mathbb{R}^n)}}_{\text{value error}}.$$

This is the direct analogue of the PINNs generalization estimate: the global weighted source error H_∞ is upper bounded by the empirical training source loss, the quadrature error, viscosity error, and value error.

The full gradient estimate in Theorem 4.2 involves nonlinear PDE techniques but provides a fruitful understanding. Instead, if estimating the projected gradient $B^\top\nabla V$, the control policy could be directly calculated without several conditional assumptions in Theorem 4.2. We provide the following simpler analysis for unweighted policy error.

Theorem 4.3. *Let V^ε be a classical solution of the viscous HJBE, and let V_θ be the neural network approximation, with $V^\varepsilon, V_\theta \in C^2(\Omega)$. Assume $\varepsilon > 0$, $\lambda \ge 0$, and $R \succ 0$. Define $G(x) = B(x)R^{-1}B(x)^\top, w(x) = V^\varepsilon(x) - V_\theta(x)$, and $b_0(x) = -f(x) + \frac{1}{2}G(x)\nabla V^\varepsilon(x) \in W^{1,\infty}_{\mathrm{loc}}(\Omega)$. Let $u_\theta = u(x, \nabla V_\theta(x))$ and $u^\varepsilon := u(x, \nabla V^\varepsilon(x))$ defined in* (3)*. For any compact subsets sequences $B_r \Subset B_{r_1} \Subset \Omega$, there exists a constant $C_{\varepsilon,r,r_1} > 0$, independent of V_θ such that if the vanishing-viscosity policy error satisfies $\|u^\varepsilon - u\|^2_{L^2(B_r)} \le \eta_\varepsilon(r)$, then*

$$\|u_\theta - u\|^2_{L^2(B_r)} \le \frac{2}{\lambda_{\min}(R)}\Big(\|r_\varepsilon[V_\theta]\|_{L^1(B_{r_1})} + C_{\varepsilon,r,r_1}\|V^\varepsilon - V_\theta\|_{L^1(B_{r_1})}\Big) + 2\eta_\varepsilon(r).$$

5 Numerical Experiments

For the main-text experiments, we set $\gamma = 1$ in the loss (11). For each benchmark, the value network has three hidden layers of 256 $\tanh$ units and is trained with Adam. Initialization and split seeds are fixed at 42. The collected state–control samples are split pointwise into training and held-out sets at ratios of 80/20 for Cucker–Smale and 90/10 for Allen–Cahn. Cucker–Smale uses 4000 epochs, batch size 128, and an initial learning rate of 2×10^{-4}; Allen–Cahn uses 3000 epochs, batch size 4096, and an initial learning rate of 2×10^{-3}. All simulations are implemented in **Pytorch** and run on a laptop with an AMD Ryzen 9 8945HX (16 cores, 2.50 GHz base, 3.50 GHz boost) and an NVIDIA GeForce RTX 5060 Laptop GPU (8 GB dedicated memory). [2]

For a given trajectory, we denote by $J_T^*(x(t))$ the accumulated cost driven by the SDRE controller from state $x(t)$ over the finite horizon $[t, T]$. This quantity serves as a reference in comparisons.

Cucker-Smale model. We consider a multi-agent system with $N_a = 20$ agents governed by:

$$\begin{bmatrix}\dot{y}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}0_{N_a} & I_{N_a}\\ 0_{N_a} & A_{N_a}(y)\end{bmatrix}\begin{bmatrix}y\\ v\end{bmatrix} + \begin{bmatrix}0_{N_a}\\ I_{N_a}\end{bmatrix}u, \tag{14}$$

[2] Code and the datasets and trained models used in the main-text experiments are available at `https://github.com/myspark2026/PI_optimal-control-by-control-only-supervision`.

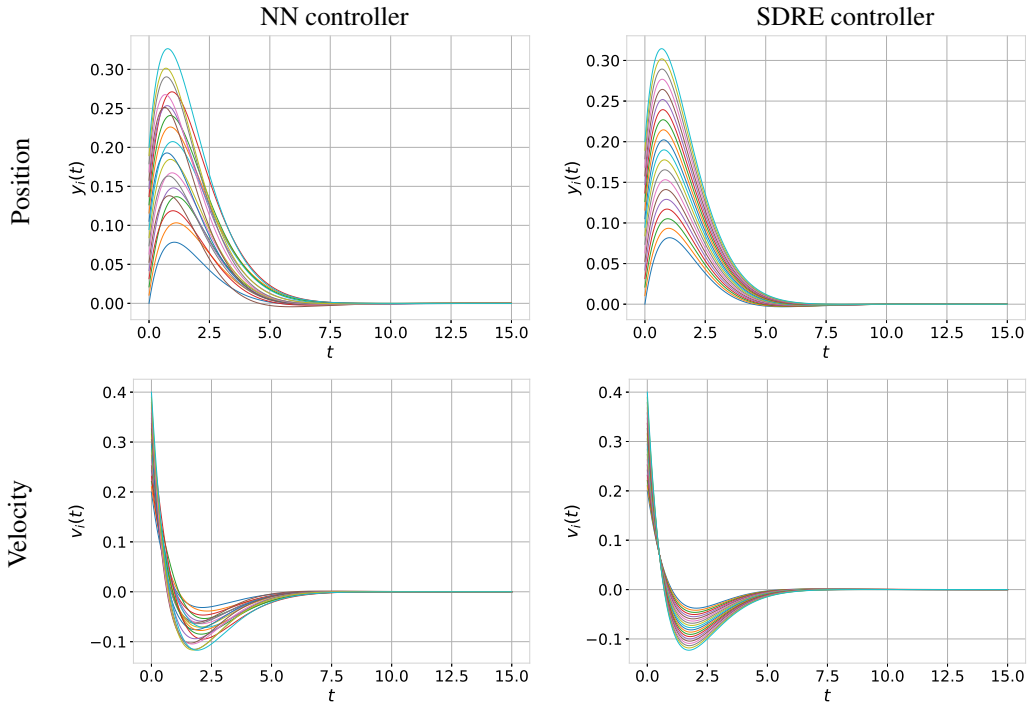

Figure 1: Position and velocity trajectories of the Cucker-Smale model under the NN controller (left column) and the SDRE controller (right column).

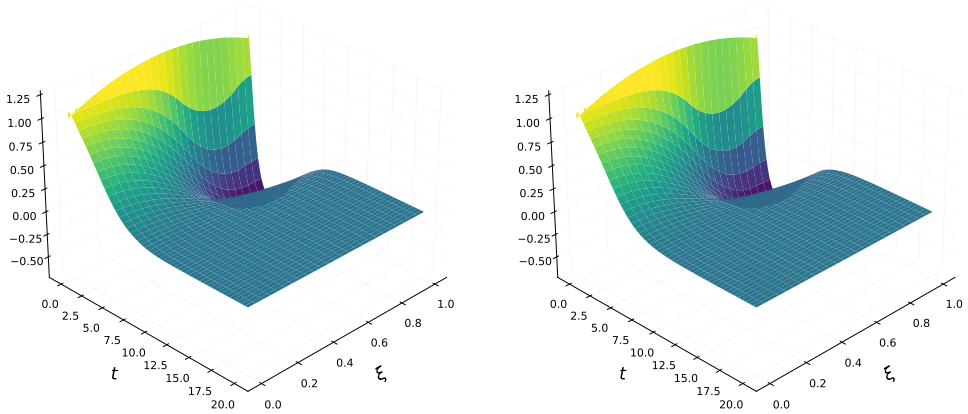

Figure 2: Solution of the Allen-Cahn equation under the NN controller (left) and the SDRE controller (right).

where 0_n denotes the $n \times n$ zero matrix, and $[A(y)]_{i,j} = -\dfrac{1}{N_a}\displaystyle\sum_{k\neq i} P(y_i, y_k)$ if $i = j$ otherwise $\dfrac{1}{N_a}P(y_i, y_j)$. with $P(y_i, y_j) = 1/(1 + \|y_i - y_j\|^2)$. The cost functional is $J(y, v, u) = \frac{1}{N_a}\int_0^T \sum_{i=1}^{N_a}\left(\|y_i\|^2 + \|v_i\|^2 + \|u_i\|^2\right) \mathrm{d}t$. To generate the training set $\mathcal{D} = \{(x_i, u_i)\}_{i=1}^N$, we sample initial states $x_0 = [y_0^\top, v_0^\top]^\top$ uniformly from $(-1, 1)^{40}$. For each of $N_{\text{traj}} = 400$ independent initial conditions, we simulate the closed-loop system under the SDRE controller for $N_T = 50$ steps with a fixed $\Delta t = 0.2$; state-control pairs are recorded at every step, yielding $N = 20\,000$ samples in total.

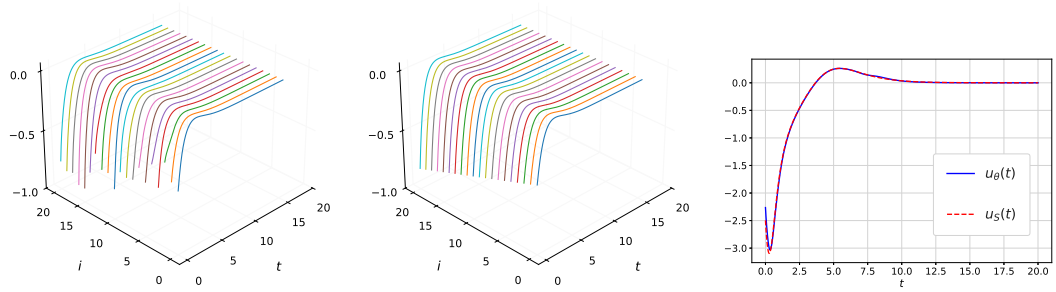

Figure 3: Control signals. Left: Cucker–Smale under NN controller. Middle: Cucker–Smale under SDRE controller. Right: Allen–Cahn under NN and SDRE controllers.

Allen–Cahn equation. We consider the nonlinear Allen–Cahn PDE with Neumann boundary conditions [12]:

$$\partial_t x(\xi,t) = 0.1\partial_{\xi\xi}x + x(1-x^2) + \chi_\omega(\xi)u(t),$$

where $\xi \in [0,1]$ and $t \in (0,\infty)$. The control $u : [0,\infty) \to \mathbb{R}$ is scalar, and $\chi_\omega(\xi)$ is the indicator function of the interval $\omega = [0.6, 0.9]$. We aim to minimize $\mathcal{J}(u,x) = \int_0^{+\infty}\left(\int_0^1 |x(\xi,t)|^2 d\xi + 0.1u^2(t)\right)\mathrm{d}t$. Using finite differences on a uniform grid of $d = 51$ points, the PDE is semi-discretized as

$$\dot{x}(t) = A(x)x(t) + Bu(t), \quad \text{with} \quad A(x) = 0.1A_0 + I_d - diag(x \odot x), \tag{15}$$

where $\odot$ is the Hadamard product, A_0 is the tridiagonal discrete Laplacian matrix and

$$B_i = \chi_{[0.6,0.9]}(\xi_i) = \begin{cases} 1, & \xi_i \in [0.6, 0.9], \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, d.$$

We generate $N_{\text{traj}} = 50$ training trajectories. The initial profiles are constructed as $x_0(\xi) = s\sum_{k=1}^{5} a_k \cos(2\pi k\xi + \phi_k) + b$, where $a_k \sim \mathcal{U}(-\frac{1}{k}, \frac{1}{k})$, $\phi_k \sim \mathcal{U}(0, 2\pi)$, $s \sim \mathcal{U}(0.5, 2)$, and $b \sim \mathcal{U}(-1, 1)$. The random scaling factor s and baseline shift b provide diverse initial conditions. For each t initial condition, we simulate $N_T = 200$ steps with $\Delta t = 0.01$ under the SDRE controller and collect statecontrol pairs. This gives $N = 10\,000$ samples.

The Cucker–Smale and Allen–Cahn benchmarks further evaluate the proposed method under general control structures. Specifically, $B \in \mathbb{R}^{40\times 20}$ and $B \in \mathbb{R}^{51\times 1}$, respectively, so neither system satisfies the structural condition $BR^{-1}B^\top = I$ required by Theorem 4.2. However, the new unweighted policy-error estimate in Theorem 4.3 removes this restriction, as well as the gradient-form assumption on the drift, and is applicable to underactuated systems under its stated regularity and residual conditions. Moreover, Theorem 4.1 provides value-error bounds without requiring these structural assumptions. Although the theoretical error conditions are not directly verified by finite trajectory samples, the numerical results provide complementary empirical evidence of the method's effectiveness in the tested operating regions.

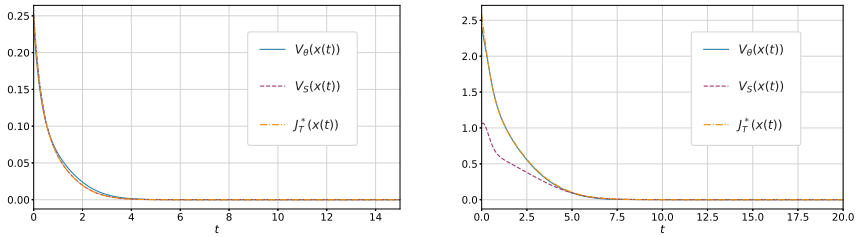

Figure 4: Comparison of the learned value function $V_\theta(x(t))$, the SDRE-based approximation $V_S(x(t))$, and the cost-to-go $J_T^*(x(t))$. Left: Cucker–Smale. Right: Allen–Cahn.

Result. We evaluate the trained NN controller and the baseline SDRE controller on test trajectories that were not used during training. We use equally spaced initial positions in $[0, 0.2]$ and velocities in $[0.2, 0.4]$ for Cucker–Smale ($T = 15$), and $x_0(\xi) = 1 + \xi(1 - \xi)$ for Allen–Cahn ($T = 20$), with a control-update interval of 0.01 in both systems. Figures 1 and 2 show that the NN controller stabilizes both systems to the origin, with a performance visually comparable to that of the SDRE controller. The control signals in Figure 3 confirm this observation: for the Cucker–Smale model, the NN and SDRE controls are almost identical, and for the Allen–Cahn equation, the two controllers produce virtually overlapping signals.

Figure 4 compares the value function learned by our method V_θ, with the SDRE approximation V_S and the cost-to-go J_T^*. In the Cucker–Smale example, all three curves nearly coincide. In the Allen–Cahn example, however, V_S clearly deviates from J_T^*, while V_θ remains in close agreement with the cost-to-go. This indicates that, at least for some nonlinear systems, the SDRE value function V_S is not a faithful surrogate for the realized cost-to-go, and using it as a supervised label could introduce systematic bias. By relying solely on control labels and the HJB residual, our method avoids this pitfall and achieves better agreement with the realized SDRE cost-to-go.

Another advantage of the proposed method is its computational speed at inference. Table 1 reports the average wall-clock time required to evaluate a single control action. In both examples, the NN controller is approximately 20–30 times faster than the SDRE solver, which must solve a state-dependent Riccati equation at every step. The speedup becomes even more significant in higher-dimensional settings or when the controller is called repeatedly inside a simulation loop.

Table 1: Average evaluation time (seconds) per state: SDRE controller vs. NN controller.

Example	SDRE	u_θ
Cucker–Smale ($d = 40$)	0.0063	0.0003
Allen–Cahn ($d = 51$)	0.0115	0.0004

Additional experiments show that the HJB term improves value–cost agreement, while control-only training can retain comparable closed-loop costs. HJB-only training fails in the tested cases. Loss and data-size ablations, loss-weight sensitivity, label-noise robustness, OOD evaluation, tests under perturbed dynamics, and full experimental details are reported in Appendix B.

6 Conclusion

We presented a physics-informed method that learns optimal feedback laws from control labels only, without value function data. A neural network parameterizes the value function and is trained by jointly minimizing the HJB residual and the mismatch with SDRE-precomputed controls. Error bounds on the value function and its gradient are established through the viscous HJBE. Experiments on the Cucker–Smale and Allen–Cahn systems show that the resulting controller matches SDRE accuracy while being 20–30 times faster, and faithfully recovers the cost-to-go even when SDRE-based value approximations deviate. This method enables fast neural control with potential application in robotics and autonomous systems, though safety-critical deployment requires additional validation.

The proposed method has several limitations. First, it requires full knowledge of the system dynamics. Second, the method relies on precomputed SDRE controls, which can be computationally expensive in high dimensions. Third, the weighted loss formulation introduces a hyperparameter balancing different components, and optimal tuning may be problem-dependent. In addition, our theoretical guarantees are conditional. The numerical results therefore provide empirical evidence within the tested operating regions. Future work includes adaptive residual-based sampling and loss weighting, considering a partially unknown control system and combining Physics-informed learning with inverse optimal control.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China grant (No. 12501613). The authors declare no competing interests.

References

[1] H. Peng, F. Li, J. Liu, and Z. Ju. A symplectic instantaneous optimal control for robot trajectory tracking with differential-algebraic equation models. *IEEE Transactions on Industrial Electronics*, 67(5):3819–3829, 2019.

[2] M. Vidyasagar. Optimal rejection of persistent bounded disturbances. *IEEE Transactions on Automatic Control*, 31(6):527–534, 2003.

[3] S. Sardellitti, S. Barbarossa, and A. Swami. Optimal topology control and power allocation for minimum energy consumption in consensus networks. *IEEE transactions on signal processing*, 60(1):383–399, 2011.

[4] R. Dorfman. An economic interpretation of optimal control theory. *The American Economic Review*, 59(5):817–831, 1969.

[5] Y. Cherruault. *Mathematical modelling in biomedicine: optimal control of biomedical systems*. Springer Science & Business Media, 2012.

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

[7] H. E. Wiltzer, D. Meger, and M. G. Bellemare. Distributional Hamilton-Jacobi-Bellman equations for continuous-time reinforcement learning. In *International Conference on Machine Learning*, pages 23832–23856. PMLR, 2022.

[8] J. Chen, W. Zhan, and M. Tomizuka. Autonomous driving motion planning with constrained iterative LQR. *IEEE Transactions on Intelligent Vehicles*, 4(2):244–254, 2019.

[9] T. Çimen. State-dependent Riccati equation (SDRE) control: a survey. *IFAC Proceedings Volumes*, 41(2):3761–3775, 2008.

[10] Z. Lin, J. Duan, S. E. Li, H. Ma, J. Li, J. Chen, B. Cheng, and J. Ma. Policy-iteration-based finite-horizon approximate dynamic programming for continuous-time nonlinear optimal control. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5255–5267, 2022.

[11] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing*, 43(2):A1221–A1247, 2021.

[12] G. Albi, S. Bicego, and D. Kalise. Gradient-augmented supervised learning of optimal feedback laws using state-dependent Riccati equations. *IEEE Control Systems Letters*, 6:836–841, 2021.

[13] J. Y. Lee and Y. Kim. Hamilton–Jacobi based policy-iteration via deep operator learning. *Neurocomputing*, 646:130515, 2025.

[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[15] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[16] X. Jiao, J. Shi, X. Chen, and F. Xiong. GsPINN: A novel fast green kernel solver based on symmetric physics-informed neural networks. *Neurocomputing*, page 132706, 2026.

[17] F. Fotiadis and K. G. Vamvoudakis. A physics-informed neural networks framework to solve the infinite-horizon optimal control problem. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6014–6019. IEEE, 2023.

[18] A. Shilova, T. Delliaux, P. Preux, and B. Raffin. Learning HJB viscosity solutions with PINNs for continuous-time reinforcement learning. In *ICML 2024 Workshop: Foundations of Reinforcement Learning and Control–Connections and Perspectives*, 2024.

[19] C. Li and R. Zeng. Aw-el-pinns: A multi-task learning physics-informed neural network for euler-lagrange systems in optimal control problems. *Neural Networks*, 199:108694, 2026.

[20] C. Banerjee, K. Nguyen, C. Fookes, and M. Raissi. A survey on physics informed reinforcement learning: Review and open problems. *Expert Systems with Applications*, 287:128166, 2025.

[21] R. Ni and A. H. Qureshi. NTFields: Neural time fields for physics-informed robot motion planning. In *International Conference on Learning Representations*, 2023.

[22] L. C. Evans. Adjoint and compensated compactness methods for Hamilton–Jacobi PDE. *Archive for rational mechanics and analysis*, 197(3):1053–1088, 2010.

A Technical Appendices and Supplementary Material

We attached the corresponding proof details here.

A.1 Viscosity solution method

Since the weak regularity admitted by the HJBE, a type of weak solution called a viscosity solution was proposed.

Definition A.1. *V(x) is called a viscosity subsolution (supersolution) in Ω if for arbitrary test function $\phi \in C(\Omega)$, considering local maximum (minimum) x_0 of function $V - \phi$ satisfying $V(x_0) = \phi(x_0)$, the following inequality holds:*

$$\mathcal{H}[\phi](x_0) \leq 0, \quad (\mathcal{H}[\phi](x_0) \geq 0)$$

If a solution $V \in C(\Omega)$ is both a subsolution and a supersolution, we call it a viscosity solution.

In the following, we shall introduce one type of technique to get a viscosity solution called the viscosity vanishing method. For the static HJBE (4), it usually has multiple solutions that satisfy the optimal condition. In order to solve this issue, the following viscosity form of the HJB is considered

$$\lambda V^\varepsilon(x) - \min_{u \in \mathbb{U}} \left\{ \nabla V^\varepsilon(x)^\top (f(x) + B(x)u) + \ell(x,u) \right\} = \varepsilon \Delta V^\varepsilon(x).$$

Substituting the closed-form optimal control (3), we get the expansion form.

$$\lambda V^\varepsilon(x) - x^\top Q x - \nabla V^\varepsilon(x)^\top \cdot f(x) + \frac{1}{4} \nabla V^\varepsilon(x)^\top B(x) R^{-1} B(x)^\top \nabla V^\varepsilon(x) = \varepsilon \Delta V^\varepsilon(x), \tag{16}$$

where $x \in \Omega$ and $\varepsilon \Delta V^\varepsilon$ denotes diffusion terms with small coefficient. Under some mild regularity condition on f and ℓ [22], the classical Bernstein method can be used to show (16) has a unique classical solution $V^\varepsilon(x)$, which is a viscosity solution at the same time. A viscosity solution is easier to calculate by numerical methods due to the additional diffusion term.

Under suitable regularity and growth assumptions, the vanishing-viscosity approximation admits the following error estimate on the domain of interest:

$$\|V(x) - V^\varepsilon(x)\|_{L^\infty(\Omega)} \leq C\sqrt{\varepsilon}, \tag{17}$$

where V denotes the viscosity solution of the original HJBE.

For the case of a bounded state space $\Omega \neq \mathbb{R}^n$, the control policy should satisfy certain boundary conditions expressed by the inequality of the Hamiltonian:

$$-\mathcal{H}(x, V, \nabla V + \alpha \hat{n}) \leq 0,$$

for any $\alpha \leq 0$ and $x \in \partial\Omega$. Here $\hat{n}$ denotes the outer normal vector at the point x. For infinite-horizon HJBEs without state constraints, i.e., $\Omega = \mathbb{R}^n$, the boundary condition is typically replaced by a growth condition at infinity, because the domain is unbounded. Heres a detailed explanation. The corresponding proof can be found in section A.2.

A.2 Growth rate estimate

Proposition A.1. *Consider the infinite-horizon discounted optimal control problem* (1)*. Assume that there exists an admissible control $\bar{u}(\cdot)$ such that the corresponding trajectory satisfies*

$$\|x(t;\bar{u})\| \le C'(1+\|x\|)e^{Kt}, \qquad t \ge 0,$$

for some constants $C' > 0$ and $K > 0$. Furthermore, suppose that the running cost along this trajectory satisfies

$$0 \le \ell(x(t;\bar{u}),\bar{u}(t)) \le C\left(1+\|x(t;\bar{u})\|^p\right),$$

for some constants $C > 0$ and integer $p \ge 1$. If $\lambda > Kp$, then the value function $V(x)$ satisfies the polynomial growth estimate

$$0 \le V(x) \le C''(1+\|x\|^p),$$

where $C'' > 0$ is independent of x.

Proof. By the definition of the value function and the non-negativity of the running cost, we have

$$0 \le V(x) = \inf_{u(\cdot)\in\mathcal{U}} J(u(\cdot),x) \le J(\bar{u}(\cdot),x),$$

where $\bar{u}(\cdot)$ is the admissible control specified in the statement.

Using the polynomial growth assumption on the running cost and the exponential estimate for the corresponding trajectory, we obtain

$$\begin{aligned}\ell(x(t;\bar{u}),\bar{u}(t)) &\le C\left(1+\|x(t;\bar{u})\|^p\right)\\ &\le C\left(1+C'^p(1+\|x\|)^p e^{Kpt}\right)\\ &\le C_1\left(1+\|x\|^p e^{Kpt}+e^{Kpt}\right),\end{aligned}$$

where $C_1 > 0$ is independent of x and t.

Consequently,

$$\begin{aligned}V(x) \le J(\bar{u}(\cdot),x) &= \int_0^\infty e^{-\lambda t}\ell(x(t;\bar{u}),\bar{u}(t))\,dt\\ &\le C_1\int_0^\infty e^{-\lambda t}\,dt + C_1(1+\|x\|^p)\int_0^\infty e^{-(\lambda-Kp)t}\,dt.\end{aligned}$$

Since $\lambda > Kp$, both integrals are finite, and hence

$$V(x) \le \frac{C_1}{\lambda} + \frac{C_1}{\lambda-Kp}(1+\|x\|^p) \le C''(1+\|x\|^p),$$

for some constant $C'' > 0$ independent of x. Therefore,

$$0 \le V(x) \le C''(1+\|x\|^p).$$

This completes the proof. □

A.3 Proof of Lemma 4.1

Recall that the modified viscous HJB equation (13) can be written as

$$\lambda V_\theta(x) - \min_{u\in U}\left\{\nabla V_\theta(x)^\top(f(x)+B(x)u) + \widetilde{\ell}_\varepsilon(x,u)\right\} - \varepsilon\Delta V_\theta(x) = 0.$$

For any admissible control u, we have

$$\nabla V_\theta(x)^\top(f(x)+B(x)u) + \varepsilon\Delta V_\theta(x) - \lambda V_\theta(x) \ge -\widetilde{\ell}_\varepsilon(x,u). \tag{18}$$

Consider the controlled stochastic dynamics

$$dX_t = (f(X_t)+B(X_t)u_t)\,dt + \sqrt{2\varepsilon}\,dW_t, \qquad X_0 = x.$$

Applying Itô's formula to $e^{-\lambda t}V_\theta(X_t)$ yields

$$d\left(e^{-\lambda t}V_\theta(X_t)\right) = e^{-\lambda t}\Big[\nabla V_\theta(X_t)^\top(f(X_t)+B(X_t)u_t)+\varepsilon\Delta V_\theta(X_t)-\lambda V_\theta(X_t)\Big]dt + \sqrt{2\varepsilon}\,e^{-\lambda t}\nabla V_\theta(X_t)^\top dW_t.$$

Integrating over $[0,T]$, taking expectations, and using (18), we obtain

$$\mathbb{E}_x\left[e^{-\lambda T}V_\theta(X_T)\right]-V_\theta(x)\geq -\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_t)\,dt\right].$$

Hence,

$$V_\theta(x)\leq \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_t)\,dt+e^{-\lambda T}V_\theta(X_T)\right]. \tag{19}$$

Since this holds for every admissible control,

$$V_\theta(x)\leq \inf_{u\in\mathcal{U}}\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_t)\,dt+e^{-\lambda T}V_\theta(X_T)\right].$$

On the other hand, consider the feedback control

$$u_\theta(x)=-\frac{1}{2}R^{-1}B(x)^\top\nabla V_\theta(x),$$

which minimizes the Hamiltonian. By its admissibility, the inequality in (18) becomes an equality under u_θ. Repeating the above argument gives

$$V_\theta(x)=\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_\theta(X_t))\,dt+e^{-\lambda T}V_\theta(X_T)\right].$$

Combining this equality with (19) yields

$$V_\theta(x)=\inf_{u\in\mathcal{U}}\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\widetilde{\ell}_\varepsilon(X_t,u_t)\,dt+e^{-\lambda T}V_\theta(X_T)\right].$$

This completes the proof.

■

A.4 Proof of Theorem 4.1

For every $T\geq 0$, we write down the SDPP formulation for the true viscosity solution.

$$V^\varepsilon(x)=\inf_{u\in\mathcal{U}}\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\ell(X_t,u_t)\,dt+e^{-\lambda T}V^\varepsilon(X_T)\right].$$

In particular, for the optimal control u^*,

$$V^\varepsilon(x)=\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\ell(X_t^*,u_t^*)\,dt+e^{-\lambda T}V^\varepsilon(X_T^*)\right]. \tag{20}$$

On the other hand, since V_θ solves the modified viscosity form HJB with running cost $\tilde{\ell}_\varepsilon$, its SDPP yields the following form

$$V_\theta(x)=\inf_{u\in\mathcal{U}}\mathbb{E}_x\left[\int_0^T e^{-\lambda t}\tilde{\ell}_\varepsilon(X_t,u_t)\,dt+e^{-\lambda T}V_\theta(X_T)\right].$$

Evaluating this identity along the control u^*, we get

$$V_\theta(x)\leq \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\tilde{\ell}_\varepsilon(X_t^*,u_t^*)\,dt+e^{-\lambda T}V_\theta(X_T^*)\right]. \tag{21}$$

Using the relation between running cost $\tilde{\ell}_\varepsilon(x,u) = \ell(x,u) + r_\varepsilon(x)$. Inequality (21) becomes

$$V_\theta(x) \le \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\big(\ell(X_t^*, u_t^*) + r_\varepsilon(X_t^*)\big)\,dt + e^{-\lambda T}V_\theta(X_T^*)\right]. \tag{22}$$

Subtracting (20) from (22), we obtain

$$V_\theta(x) - V^\varepsilon(x) \le \mathbb{E}_x\left[\int_0^T e^{-\lambda t} r_\varepsilon(X_t^*)\,dt\right] + \mathbb{E}_x\left[e^{-\lambda T}\big(V_\theta(X_T^*) - V^\varepsilon(X_T^*)\big)\right].$$

i.e.

$$V_\theta(x) - V^\varepsilon(x) \le \mathbb{E}_x\left[\int_0^T e^{-\lambda t}|r_\varepsilon(X_t^*)|\,dt\right] + \mathbb{E}_x\left[e^{-\lambda T}\big|V_\theta(X_T^*) - V^\varepsilon(X_T^*)\big|\right].$$

Passing to the limit as $T \to \infty$, and using the terminal condition assumption, we arrive at

$$V_\theta(x) - V^\varepsilon(x) \le \mathbb{E}_x \int_0^\infty e^{-\lambda t}|r_\varepsilon(X_t^*)|\,dt. \tag{23}$$

To obtain the reverse estimate, we apply the SDPP for V_θ along its own optimal control $\widetilde{u}$:

$$V_\theta(x) = \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\tilde{\ell}_\varepsilon(\widetilde{X}_t, \widetilde{u}_t)\,dt + e^{-\lambda T}V_\theta(\widetilde{X}_T)\right]. \tag{24}$$

Since V^ε is the infimum of the original cost, evaluating its SDPP along the same control $\widetilde{u}$ gives

$$V^\varepsilon(x) \le \mathbb{E}_x\left[\int_0^T e^{-\lambda t}\ell(\widetilde{X}_t, \widetilde{u}_t)\,dt + e^{-\lambda T}V^\varepsilon(\widetilde{X}_T)\right]. \tag{25}$$

Subtracting (24) from (25), and using again $\tilde{\ell}_\varepsilon = \ell + r_\varepsilon$, we obtain

$$V^\varepsilon(x) - V_\theta(x) \le \mathbb{E}_x\left[\int_0^T e^{-\lambda t}|r_\varepsilon(\widetilde{X}_t)|\,dt\right] + \mathbb{E}_x\left[e^{-\lambda T}\big|V^\varepsilon(\widetilde{X}_T) - V_\theta(\widetilde{X}_T)\big|\right].$$

Letting $T \to \infty$, we deduce

$$V^\varepsilon(x) - V_\theta(x) \le \mathbb{E}_x \int_0^\infty e^{-\lambda t}|r_\varepsilon(\widetilde{X}_t)|\,dt. \tag{26}$$

Combining (26) and (23), we conclude that

$$|V_\theta(x) - V^\varepsilon(x)| \le \max\left\{\mathbb{E}_x\int_0^\infty e^{-\lambda t}|r_\varepsilon(X_t^*)|\,dt,\, \mathbb{E}_x\int_0^\infty e^{-\lambda t}|r_\varepsilon(\widetilde{X}_t)|\,dt\right\}.$$

Finally, if $\lambda > 0$ and $r_\varepsilon \in L^\infty$, then for any admissible state process X_t,

$$\mathbb{E}_x\int_0^\infty e^{-\lambda t}|r_\varepsilon(X_t)|\,dt \le \|r_\varepsilon\|_{L^\infty}\int_0^\infty e^{-\lambda t}\,dt = \frac{1}{\lambda}\|r_\varepsilon\|_{L^\infty}.$$

Therefore,

$$|V_\theta(x) - V^\varepsilon(x)| \le \frac{1}{\lambda}\|r_\varepsilon\|_{L^\infty}.$$

This completes the proof. ■

A.5 Liouville-type inequality

Lemma A.1. *Let $B_r := \{x \in \mathbb{R}^n : |x| \le r\}$ be a bounded ball. Assume that the control matrix $B(x)$ has full rank and $B(x)R^{-1}B(x)^\top = I$. Assume the drift function is of the gradient form $f = \nabla\phi$. Denote $\Phi(x) := \frac{1}{2}V^\varepsilon(x) - \phi(x)$ and $\rho(x) := \exp\left(-\frac{\Phi(x)}{\varepsilon}\right)$. Let $R_\varepsilon(x) := r_\varepsilon[V_\theta](x) + \lambda w(x)$ be the error source term derived from the viscosity form HJB and the decay term. Then we have the following energy estimate for the weighted energy.*

$$\frac{1}{4}E(r) \le \varepsilon\sqrt{E'(r)M'(r)} + H(r). \tag{27}$$

where $E(r) := \int_{B_r}\rho|\nabla w|^2, M(r) := \int_{B_r}\rho, H(r) := \int_{B_r}\rho|R_\varepsilon|$.

Proof. Subtract (16) and (12), and re-express the equation in terms of the error.

$$\lambda w - \nabla w^\top f + \frac{1}{2}\nabla V^\varepsilon\cdot\nabla w - \frac{1}{4}|\nabla w|^2 = \varepsilon\Delta w - r_\varepsilon[V_\theta]. \tag{28}$$

For simplicity, we assume that the control matrix $B(x)$ has full rank and that $B(x)R^{-1}B(x)^\top = I$. We further assume that $f = \nabla\phi$. Define

$$\Phi(x) := \frac{1}{2}V^\varepsilon(x) - \phi(x), \qquad R_\varepsilon(x) := r_\varepsilon[V_\theta](x) + \lambda w(x).$$

Then (28) can be rewritten as

$$-\varepsilon\Delta w + \nabla\Phi\cdot\nabla w - \frac{1}{4}|\nabla w|^2 = -R_\varepsilon.$$

Let

$$\rho(x) := \exp\left(-\frac{\Phi(x)}{\varepsilon}\right).$$

Since

$$\rho\left(-\varepsilon\Delta w + \nabla\Phi\cdot\nabla w\right) = -\varepsilon\nabla\cdot(\rho\nabla w),$$

Multiplying the error equation by ρ gives

$$-\varepsilon\nabla\cdot(\rho\nabla w) - \frac{1}{4}\rho|\nabla w|^2 = -\rho R_\varepsilon.$$

Integrating over $B_r := \{x \in \mathbb{R}^n : |x| \le r\}$, and applying the divergence theorem, we obtain

$$-\varepsilon\int_{\partial B_r}\rho\nabla w\cdot\nu - \frac{1}{4}\int_{B_r}\rho|\nabla w|^2 = -\int_{B_r}\rho R_\varepsilon,$$

where ν denotes the unit outward normal vector on ∂B_r. Hence,

$$\frac{1}{4}\int_{B_r}\rho|\nabla w|^2 = \int_{B_r}\rho R_\varepsilon - \varepsilon\int_{\partial B_r}\rho\nabla w\cdot\nu.$$

Taking absolute values and applying the Cauchy-Schwarz inequality to the boundary term yields

$$\frac{1}{4}\int_{B_r}\rho|\nabla w|^2 \le \varepsilon\left(\int_{\partial B_r}\rho|\nabla w|^2\right)^{1/2}\left(\int_{\partial B_r}\rho\right)^{1/2} + \int_{B_r}\rho|R_\varepsilon|. \tag{29}$$

Now define

$$E(r) := \int_{B_r}\rho|\nabla w|^2, \qquad M(r) := \int_{B_r}\rho, \qquad H(r) := \int_{B_r}\rho|R_\varepsilon|.$$

Then, for almost every $r > 0$,

$$E'(r) = \int_{\partial B_r}\rho|\nabla w|^2, \qquad M'(r) = \int_{\partial B_r}\rho.$$

Therefore, estimate (29) becomes the following energy estimate

$$\frac{1}{4}E(r) \le \varepsilon\sqrt{E'(r)M'(r)} + H(r).$$

Equivalently,

$$\left(\frac{1}{4}E(r) - H(r)\right)_+^2 \le \varepsilon^2 E'(r)M'(r),$$

where $(a)_+ := \max\{a, 0\}$. □

A.6 Proof of Theorem 4.2

Recall we have derived error energy estimate (27):

$$\frac{1}{4}E(r) \le \varepsilon\sqrt{E'(r)M'(r)} + H(r). \tag{30}$$

By the statement assumptions for $M(r)$ and $H(r)$, we have

$$M'(r) \le C, \qquad H(r) \le H_\infty \le \frac{\delta}{8}.$$

Thus (30) implies

$$\frac{1}{4}E(r) \le \varepsilon\sqrt{CE'(r)} + H_\infty. \tag{31}$$

We now prove the desired estimate by contradiction. Suppose that there exists $r_0 > 0$ such that $E(r_0) > \delta$. Since $E(r)$ is non-decreasing in r, it follows that $E(r) \ge E(r_0) > \delta$ for all $r \ge r_0$. Hence,

$$H_\infty \le \frac{\delta}{8} < \frac{E(r)}{8} \qquad \text{for all } r \ge r_0.$$

Consequently,

$$\frac{1}{4}E(r) - H_\infty \ge \frac{1}{8}E(r).$$

From (31), we then obtain

$$\frac{1}{8}E(r) \le \varepsilon\sqrt{CE'(r)} \qquad \text{for all } r \ge r_0.$$

Squaring both sides gives

$$E(r)^2 \le 64\varepsilon^2 CE'(r).$$

Equivalently,

$$\frac{1}{64\varepsilon^2 C} \le \frac{E'(r)}{E(r)^2}.$$

Integrating this inequality over (r_0, R), we obtain

$$\frac{R - r_0}{64\varepsilon^2 C} \le \int_{r_0}^{R} \frac{E'(r)}{E(r)^2}\, dr.$$

Since

$$\int_{r_0}^{R} \frac{E'(r)}{E(r)^2}\, dr = \frac{1}{E(r_0)} - \frac{1}{E(R)},$$

we have

$$\frac{R - r_0}{64\varepsilon^2 C} \le \frac{1}{E(r_0)} - \frac{1}{E(R)} \le \frac{1}{E(r_0)}.$$

Letting $R \to \infty$, the left-hand side diverges, while the right-hand side remains finite. This contradiction shows that no such r_0 can exist.

Therefore, we get $E(r) \leq \delta \quad$ for every $r > 0$. That is, $\int_{B_r} \rho|\nabla w|^2 \leq \delta$. Substituting back $\rho = e^{-\Phi/\varepsilon}$, we conclude that $\int_{B_r} \exp\left(-\frac{\Phi}{\varepsilon}\right)|\nabla w|^2 \leq \delta$, which proves the desired weighted gradient estimate. By applying Young's inequality, we have

$$\int_{B_r} \rho|\nabla V_\theta - \nabla V|^2 \leq 2\int_{B_r} \rho|\nabla V_\theta - \nabla V^\varepsilon|^2 + 2\int_{B_r} \rho|\nabla V^\varepsilon - \nabla V|^2.$$

It follows that

$$\int_{B_r} \rho|\nabla V_\theta - \nabla V|^2 \leq 2\delta + 2\eta_\varepsilon(r).$$

In the following, we shall focus on the approximation of the control function by an NN. The neural-network-induced feedback control is

$$u_\theta(x) = -\frac{1}{2}R^{-1}B(x)^\top \nabla V_\theta(x).$$

Similarly, the viscosity optimal control associated with V^ε is

$$u^\varepsilon(x) = -\frac{1}{2}R^{-1}B(x)^\top \nabla V^\varepsilon(x).$$

and the true control function

$$u(x) = -\frac{1}{2}R^{-1}B(x)^\top \nabla V(x).$$

Therefore,

$$u_\theta(x) - u(x) = -\frac{1}{2}R^{-1}B(x)^\top \left(\nabla V_\theta(x) - \nabla V(x)\right).$$

Taking norms gives the point-wise estimate

$$|u_\theta(x) - u(x)| \leq \frac{1}{2}\left\|R^{-1}B(x)^\top\right\| |\nabla V_\theta - \nabla V|.$$

Assume that

$$\left\|R^{-1}B(x)^\top\right\| \leq K_r, \qquad x \in B_r.$$

Then

$$\int_{B_r} \rho|u_\theta - u|^2 \leq \frac{K_r^2}{4}\int_{B_r} \rho|\nabla V_\theta - \nabla V|^2 \leq \frac{K_r^2}{2}(\delta + \eta_\varepsilon).$$

This gives the corresponding optimal control error estimate. ■

A.7 Proof of Theorem 4.3

Recall that V^ε satisfies the viscous HJB equation

$$\lambda V^\varepsilon - x^\top Q x - f^\top \nabla V^\varepsilon + \frac{1}{4}(\nabla V^\varepsilon)^\top G \nabla V^\varepsilon - \varepsilon \Delta V^\varepsilon = 0, \tag{32}$$

where $G(x) = B(x)R^{-1}B(x)^\top$. Similarly, by the definition of the viscous HJB residual, V_θ satisfies

$$\lambda V_\theta - x^\top Q x - f^\top \nabla V_\theta + \frac{1}{4}(\nabla V_\theta)^\top G \nabla V_\theta - \varepsilon \Delta V_\theta = r_\varepsilon[V_\theta]. \tag{33}$$

Let $w = V^\varepsilon - V_\theta$ then subtracting (33) from (32) gives

$$\lambda w - f^\top \nabla w + \frac{1}{4}\Big((\nabla V^\varepsilon)^\top G \nabla V^\varepsilon - (\nabla V_\theta)^\top G \nabla V_\theta\Big) - \varepsilon \Delta w = -r_\varepsilon[V_\theta].$$

Since $\nabla V_\theta = \nabla V^\varepsilon - \nabla w$, the quadratic difference can be written as

$$(\nabla V^\varepsilon)^\top G \nabla V^\varepsilon - (\nabla V_\theta)^\top G \nabla V_\theta = 2(\nabla V^\varepsilon)^\top G \nabla w - (\nabla w)^\top G \nabla w.$$

Consequently, error w will satisfy the following:

$$-\varepsilon \Delta w + b_0^\top \nabla w - \frac{1}{4}(\nabla w)^\top G \nabla w + \lambda w = -r_\varepsilon[V_\theta], \tag{34}$$

where $b_0 := -f + \frac{1}{2}G\nabla V^\varepsilon$. Next, observe that $u_\theta - u^\varepsilon = \frac{1}{2}R^{-1}B^\top \nabla w$. Therefore, it is directly calculated that $\frac{1}{4}(\nabla w)^\top G \nabla w = |u_\theta - u^\varepsilon|_R^2$ where $|v|_R^2 = v^\top R v$.

Let $q(x) = |u_\theta - u^\varepsilon|_R^2 \geq 0$ and (34) then becomes

$$q = r_\varepsilon[V_\theta] - \varepsilon \Delta w + b_0^\top \nabla w + \lambda w. \tag{35}$$

For any $B_r \Subset B_{r_1} \Subset \Omega$, choose a smooth cutoff function $\chi \in C_c^\infty(B_{r_1})$ satisfying

$$0 \leq \chi \leq 1, \qquad \chi = 1 \quad \text{on } B_r.$$

Multiplying (35) on both sides by χ and integrating over B_{r_1}, we then use the integration by parts to get

$$\int_{B_{r_1}} \chi q \, dx = \int_{B_{r_1}} \chi r_\varepsilon[V_\theta] \, dx + \int_{B_{r_1}} w \left(\lambda\chi - \varepsilon\Delta\chi - \nabla \cdot (\chi b_0)\right) dx. \tag{36}$$

Since $q \geq 0$ and $\chi = 1$ on B_r, it follows that $\int_{B_r} q \, dx \leq \int_{B_{r_1}} \chi q \, dx$. Using (36), we conclude that

$$\int_{B_r} |u_\theta - u^\varepsilon|_R^2 \, dx \leq \|r_\varepsilon[V_\theta]\|_{L^1(B_{r_1})} + C_{\varepsilon,r,r_1} \|V^\varepsilon - V_\theta\|_{L^1(B_{r_1})}.$$

where $C_{\varepsilon,r,r_1} = \|\lambda\chi - \varepsilon\Delta\chi - \nabla \cdot (\chi b_0)\|_{L^\infty(B_{r_1})}$ which is independent of V_θ.

Since $R \succ 0$, we have

$$|v|^2 \leq \frac{1}{\lambda_{\min}(R)} |v|_R^2.$$

Thus,

$$\|u_\theta - u^\varepsilon\|_{L^2(B_r)}^2 \leq \frac{1}{\lambda_{\min}(R)} \Big(\|r_\varepsilon[V_\theta]\|_{L^1(B_{r_1})} + C_{\varepsilon,r,r_1} \|V^\varepsilon - V_\theta\|_{L^1(B_{r_1})} \Big).$$

Finally, by the triangle inequality,

$$|u_\theta - u|^2 \leq 2|u_\theta - u^\varepsilon|^2 + 2|u^\varepsilon - u|^2.$$

Integrating over B_r and using

$$\|u^\varepsilon - u\|_{L^2(B_r)}^2 \leq \eta_\varepsilon(r)$$

gives

$$\|u_\theta - u\|_{L^2(B_r)}^2 \leq \frac{2}{\lambda_{\min}(R)} \Big(\|r_\varepsilon[V_\theta]\|_{L^1(B_{r_1})} + C_{\varepsilon,r,r_1} \|V^\varepsilon - V_\theta\|_{L^1(B_{r_1})} \Big) + 2\eta_\varepsilon(r).$$

This completes the proof.

■

B Additional Experimental Details and Results

B.1 Ablations and Loss-Weight Sensitivity

Table 2 compares the proposed method with two baselines: the same value-function network trained using $\mathcal{L}_c$ alone, with controls still derived from its gradient, and a feedforward policy network that we denote by *PolicyNet*. PolicyNet directly maps states to controls and minimizes the mean squared error between its output and the SDRE labels; it does not estimate a value function. For this comparison, all three neural methods follow the hidden-layer configurations and optimization settings of the main-text experiments in Section 5. They share a common 80/20 pointwise training/held-out partition within each benchmark: 16,000/4,000 samples for Cucker–Smale and 8,000/2,000 for Allen–Cahn. Thus, for Allen–Cahn, this 80/20 split differs from the 90/10 split used in the main-text experiments.

Table 2 also examines sensitivity to γ in the objective $\mathcal{L} = \mathcal{L}_{\mathrm{HJB}} + \gamma\mathcal{L}_c$. Within each system, we vary γ while fixing the dataset, 80/20 pointwise training/held-out partition, network architecture, initialization seed, and nominal test initial condition. Here, $\gamma = 0$ denotes *HJB-only* training,

Table 2: Ablations and loss-weight sensitivity. Both studies use an 80/20 training/held-out split for each system. Each entry uses one trained model and the nominal test initial condition for that system. E_{AUC} is evaluated on the SDRE reference trajectory defined in (37). HJB-only results are retained as failure diagnostics. Dashes indicate that no value metric is reported.

	Cucker–Smale		Allen–Cahn	
Setting	Cost ratio	E_{AUC}	Cost ratio	E_{AUC}
$\gamma = 0$ (HJB-only)	4710.4202	0.8676	13.5656	0.6801
$\gamma = 10^{-3}$	1.1801	0.4705	1.0133	0.0113
$\gamma = 10^{-2}$	1.0186	0.0610	1.0075	0.0137
$\gamma = 10^{-1}$	1.0161	0.0343	1.0041	0.0094
$\gamma = 1$ (Ours)	1.0159	0.0449	1.0031	0.0361
$\gamma = 10$	1.0053	0.0671	1.0012	0.0738
$\gamma = 10^{2}$	1.0102	0.0382	1.0020	0.2099
$\gamma = 10^{3}$	1.0119	0.3563	0.9993	0.3950
$\gamma = \infty$ (u-only)	1.0077	0.5251	0.9987	0.9930
PolicyNet	1.0578	–	0.9983	–
SDRE reference	1.0000	–	1.0000	–

while $\gamma = \infty$ denotes *u-only* training using $\mathcal{L}_c$ alone. All comparisons use the nominal test initial conditions, horizons, and control-update interval specified in Section 5.

We report the closed-loop cost ratio relative to SDRE, with both costs evaluated from the same initial condition over the same horizon. To quantify value–cost agreement, all value networks are evaluated along the same SDRE reference trajectory $x_S(t)$. The SDRE controller's realized remaining cost is

$$\widehat{J}_S(t) = \int_t^T \ell\big(x_S(\tau), u_S(x_S(\tau))\big)\, d\tau.$$

We define the normalized area discrepancy between the predicted value and this reference cost-to-go as

$$E_{\mathrm{AUC}} = \frac{\displaystyle\int_0^T \left| V_\theta(x_S(t)) - \widehat{J}_S(t) \right| dt}{\displaystyle\int_0^T |\widehat{J}_S(t)|\, dt}. \tag{37}$$

This dimensionless metric measures the integrated absolute discrepancy relative to the area under the reference cost-to-go curve; lower values indicate closer agreement. It assesses agreement with the finite-horizon SDRE cost-to-go along the reference trajectory.

The three control-supervised methods achieve comparable nominal closed-loop costs. The proposed method additionally yields a value estimate that closely tracks the realized SDRE cost-to-go, whereas the u-only value network exhibits substantially larger value–cost discrepancies. The γ study further shows that large control-loss weights can preserve near-reference control costs despite large value discrepancies, highlighting the role of the HJB term in improving value–cost consistency. HJB-only training yields substantially higher closed-loop costs in both systems, underscoring the importance of control supervision.

B.2 Repeated Label-Noise Experiments

We add independent zero-mean Gaussian noise to each SDRE control-label entry, with a standard deviation equal to r times the sample standard deviation of all original control-label entries, where $r \in \{0.10, 0.25, 0.50, 0.75\}$. Five noise realizations use seeds 2026–2030, while initialization and split seeds remain fixed at 42. Within each system, all methods share the states, noisy labels, 80/20 pointwise training/held-out partition, and test initial conditions. We set $\gamma = 1$ for the proposed method.

We use method-specific network configurations. For Cucker–Smale, the proposed method and u-only use two hidden layers of 128 and 256 units, respectively; PolicyNet uses three hidden layers of

Table 3: Label-noise robustness under the reported method-specific network configurations. Entries report normalized closed-loop cost, mean ± sample standard deviation across five noise-realization averages, each based on five common test initial conditions.

System	r	u + HJB	u-only	PolicyNet
Cucker–Smale	0.10	1.0141 ± 0.0019	1.0494 ± 0.0104	1.0276 ± 0.0018
Cucker–Smale	0.25	1.0190 ± 0.0032	1.0586 ± 0.0142	1.0402 ± 0.0032
Cucker–Smale	0.50	1.0301 ± 0.0059	1.0843 ± 0.0222	1.0818 ± 0.0068
Cucker–Smale	0.75	1.0447 ± 0.0122	1.1181 ± 0.0156	1.1649 ± 0.0097
Allen–Cahn	0.10	1.0030 ± 0.0020	1.0079 ± 0.0043	1.0055 ± 0.0032
Allen–Cahn	0.25	1.0128 ± 0.0035	1.0206 ± 0.0156	1.0670 ± 0.0587
Allen–Cahn	0.50	1.0635 ± 0.0210	1.1313 ± 0.1535	1.6970 ± 0.4303
Allen–Cahn	0.75	1.2417 ± 0.1151	1.3815 ± 0.3132	3.9040 ± 1.3282

512 units. For Allen–Cahn, both value networks use two hidden layers of 128 units, and PolicyNet uses three of 512 units. All networks use $\tanh$ activations and Adam with batch size 4096. For Cucker–Smale, we use 4000 epochs and an initial learning rate of 8×10^{-4}; for Allen–Cahn, we use 3000 epochs and 2×10^{-3}.

Each trained model is evaluated on five common test initial conditions: uniform samples from $[-1, 1]^{40}$ for Cucker–Smale and Fourier profiles from the family in Section 5 for Allen–Cahn. For each noise realization, we average the cost ratio relative to SDRE over the five test conditions, then report the mean and sample standard deviation across the five realization averages. These results in Table 3 suggest a problem-dependent benefit under noisy supervision, with more pronounced improvements at higher noise levels, particularly for Allen–Cahn. The observations are specific to the tested systems and network configurations and do not establish a uniform advantage across problems or noise conditions.

B.3 OOD and Long-Horizon Evaluation

We evaluate the original main-text $\gamma = 1$ models without retraining or modifying their feedback laws. For Allen–Cahn, this is the model trained with the original 90/10 split. We extend the nominal horizons of 15 for Cucker–Smale and 20 for Allen–Cahn to $T = 50$, and test five initial conditions per system at each scale $s \in \{1, 1.25, 1.5, 2\}$.

For Cucker–Smale, we independently sample five pairs of position and velocity vectors $y, v \in [-1, 1]^{20}$. Each block is normalized separately, and the initial state is constructed as

$$x_0 = s \begin{bmatrix} y/\|y\|_\infty \\ v/\|v\|_\infty \end{bmatrix}.$$

Here, $\|\cdot\|_\infty$ is the largest absolute component. Thus, the normalized position and velocity blocks each have maximum absolute entry one before scaling. The same five normalized directions are reused across all scales. Since the training initial conditions lie in $[-1, 1]^{40}$, the cases with $s = 1$ lie on its boundary, whereas those with $s > 1$ lie outside it.

For Allen–Cahn, the test profiles are

$$x_0(\xi) = s\big[1 + \xi(1 - \xi) + \delta(\xi)\big].$$

One profile uses $\delta = 0$, while the other four use smooth Fourier perturbations. Each perturbation is centered to have zero mean over the 51 spatial grid points before being added to the baseline profile. The same five profiles are reused across scales, with the entire profile multiplied by s. At all tested scales, the test-profile means lie outside the empirical range of the training initial-profile means.

A rollout is classified as successful if it reaches $T = 50$ and its sampled states satisfy

$$\|x(t)\|_2 \le \max\big\{10^{-3},\, 0.05\|x_0\|_2\big\} \qquad \text{for all sampled } t \in [40, 50].$$

This requires the trajectory to enter a neighborhood of the origin by time 40 and remain there until time 50.

Table 4: OOD evaluation over a horizon of $T = 50$. Cost ratios relative to SDRE are medians over successful cases only. Final state norms and absolute HJB residuals are averaged over all five cases, including failures. The repeated mean final-state norms are indistinguishable at the reported precision.

System	Scale	Successes	Median cost ratio	Mean final state norm	Mean absolute HJB residual
Allen–Cahn	1.00	5/5	1.0022	0.0042535	0.004296
Allen–Cahn	1.25	5/5	1.0046	0.0042535	0.011913
Allen–Cahn	1.50	5/5	1.0050	0.0042535	0.019222
Allen–Cahn	2.00	5/5	1.0047	0.0042535	0.032905
Cucker–Smale	1.00	5/5	1.0674	0.00053947	0.005605
Cucker–Smale	1.25	5/5	1.2052	0.00053947	0.021064
Cucker–Smale	1.50	4/5	1.4951	47.2147	83.2411
Cucker–Smale	2.00	1/5	1.5540	352.3326	1782.6155

Table 5: Data-size ablation. Entries report mean cost ratio and mean E_{AUC}, defined in (37), over five common test initial conditions. All five cases satisfy the success criterion for each setting.

System	Train/held-out	Mean cost ratio	Mean E_{AUC}
Cucker–Smale	$16000/4000$	1.1210	0.0639
Cucker–Smale	$8000/2000$	1.3742	0.1132
Allen–Cahn	$9000/1000$	1.0022	0.0324
Allen–Cahn	$4500/500$	1.0027	0.0230

Table 4 shows that Allen–Cahn satisfies the success criterion at all tested scales. Cucker–Smale succeeds in all five cases through $s = 1.25$, but only four of five cases at $s = 1.5$ and one of five at $s = 2$. The larger final state norms and HJB residuals include the failed cases and reveal the deterioration under larger distribution shifts. These results characterize empirical reliability within the tested range.

B.4 Sensitivity to Data Size and Dynamics

Both studies use the five $s = 1$ initial conditions, $T = 50$, control-update interval of 0.01, and success criterion from Section B.3.

Data-size ablation. We compare the original full-data models with models trained on half of the state–control pairs. Subsets are sampled separately from the original training and held-out partitions, preserving the 80/20 split for Cucker–Smale and the 90/10 split for Allen–Cahn. The half-data models retain the main-text training settings in Section 5. With epoch counts and batch sizes fixed, halving the data reduces the training workload, measured by optimizer updates, from $500{,}000$ to $252{,}000$ for Cucker–Smale and from $9{,}000$ to $6{,}000$ for Allen–Cahn.

Table 5 shows problem-dependent sensitivity to data quantity. Halving the data increases both cost and value–cost discrepancy for Cucker–Smale, whereas Allen–Cahn retains near-reference costs with a lower discrepancy in this run. The reduced budget affects performance without preventing successful regulation in these fixed-seed evaluations.

Dynamics perturbation. We freeze the original $\gamma = 1$ models and their input normalization, without retraining or modifying their feedback laws. For Cucker–Smale, we multiply the communication kernel $P(y_i, y_j)$ used to construct the interaction matrix in (14) by 0.9, 1.0, or 1.1. For Allen–Cahn, we replace the diffusion coefficient 0.1 multiplying A_0 in (15) by 0.09, 0.10, or 0.11. All other dynamics and cost parameters remain unchanged.

The SDRE reference is recomputed using the actual perturbed dynamics, and the HJB residual is evaluated with those dynamics. All cases in Table 6 complete the full horizon and satisfy the regulation criterion. Cucker–Smale exhibits little change in median normalized cost. For Allen–Cahn, reduced diffusion increases both normalized cost and residual, although regulation remains success-

Table 6: Frozen-policy evaluation under dynamics perturbations. The parameter is the communication-kernel multiplier for Cucker–Smale and the diffusion coefficient for Allen–Cahn. Costs are normalized by the corresponding perturbed-system SDRE costs. The cost column reports the median among successful cases; terminal norms and absolute HJB residuals are averaged over all five cases. The mean terminal-state norms differ slightly across parameter settings but are indistinguishable at the reported precision for Cucker-Smale.

System	Parameter	Success	Median cost ratio	Mean final state norm	Mean absolute HJB residual
Cucker–Smale	0.9	5/5	1.0693	5.3947×10^{-4}	0.005802
Cucker–Smale	1.0	5/5	1.0674	5.3947×10^{-4}	0.005605
Cucker–Smale	1.1	5/5	1.0680	5.3947×10^{-4}	0.005504
Allen–Cahn	0.09	5/5	1.0247	6.1133×10^{-3}	0.010137
Allen–Cahn	0.10	5/5	1.0022	4.2535×10^{-3}	0.004296
Allen–Cahn	0.11	5/5	0.9984	3.2884×10^{-3}	0.007411

ful. These results support empirical tolerance to the tested $\pm 10\%$ single-parameter perturbations within the specified initial-condition and time-horizon settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the core contributions: (1) a physics-informed method for high-dimensional optimal control using only control labels (not value labels), (2) conditional theoretical error bounds for both the value function and its gradient (providing policy guarantees), and (3) numerical validation showing accuracy, cost-to-go matching, and faster neural controller performance. These claims are specific, verifiable, and appropriately scoped to the proposed HJB-based framework.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We clearly have listed the current limitations in the second paragraph of the Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For all theoretical results, corresponding detailed proofs have been posted in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all necessary information related to the experiments' reproducibility, including hyperparameters, size of datasets, sampling domain, etc.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, the main-text datasets and trained model checkpoints, and documentation are publicly available at `https://github.com/myspark2026/PI_optimal-control-by-control-only-supervision`, including training instructions and evaluation scripts. For supplementary experiments, the release provides training and evaluation code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All related information related to the training process has been contained. These detailed training settings can be found in Section 5 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Appendix B.2 reports means and sample standard deviations across five independent label-noise realizations, with initialization and split seeds fixed. The other experiments do not report variability across independent training runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Section 5 reports the CPU, GPU, and GPU memory, Table 1 reports inference times, and Appendix B.4 reports optimizer-update counts. Per-run training times and the total research compute are not fully reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves no human subjects, personal data, or datasets requiring consent, and all experiments are conducted on standard dynamical systems.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have mentioned the potential positive and negative societal impacts in the conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released assets are synthetic simulation datasets and task-specific models for the Cucker–Smale and Allen–Cahn systems. No high risk of misuse requiring special release safeguards has been identified.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [N/A]

Justification: The paper does not use any existing external assets such as datasets, pre-trained models, or third-party code beyond standard scientific computing libraries (e.g., Pytorch). All simulations are implemented using open-source frameworks with standard licenses, and the neural controllers and value functions are original contributions of this work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces trained neural network models (value function and control policy) for the Cucker–Smale and Allen–Cahn systems. Code, the main-text model checkpoints, and documentation are available in the public repository, including training instructions and evaluation scripts.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper involves no crowdsourcing, human subjects, or any form of human participant data. All experiments are conducted solely on simulated dynamical systems (CuckerSmale and AllenCahn) with no human interaction.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper involves no human subjects, study participants, or any form of human-centric research. Therefore, IRB approval and participant risk disclosure are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology, which consists of neural network training, HJB residual minimization, and SDRE control supervision. LLM assistance was used for writing, code organization, and implementation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.