Journal of Machine Learning Research 27 (2026) 1-26 Submitted 5/25; Revised 6/26; Published 6/26

Viscosity Convergence Analysis for Deep Q-Networks

Qian Qi QIQIAN@PKU.EDU.CN
Peking University

Editor: Martha White

Abstract

Deep Q-Networks (DQNs) and related residual neural architectures are increasingly used for continuous-time reinforcement learning (CTRL), where optimal value functions solve fully nonlinear second-order Hamilton–Jacobi–Bellman (HJB) equations and may be non-smooth. In this regime, convergence should be analyzed in the viscosity-solution framework. We study a spatially-coupled monotone ResNet architecture whose one-step operator is a non-negative local aggregation, designed to satisfy monotonicity and stability at the operator level. Under a consistency assumption on the learned operator, we obtain operator-level convergence to the viscosity solution via the Barles–Souganidis framework. We then distinguish this idealized operator iteration from practical fitted value iteration with projection onto a neural function class, and discuss the resulting approximation gap. Empirically, on stochastic control benchmarks, the proposed architecture exhibits smaller overshoot near kinks and more stable numerical behavior than pointwise PINN-style baselines in our experiments.

Keywords: Deep Q-Networks, Continuous-Time Reinforcement Learning, Viscosity Solutions, Hamilton-Jacobi-Bellman Equations, Numerical Schemes, ResNets.

1. Introduction

Continuous-time reinforcement learning (CTRL) models the interaction of an agent with a stochastic environment over a continuous time horizon. Operating within a continuous state space $\mathcal{S} \subseteq \mathbb{R}^n$ and an action space $\mathcal{A} \subseteq \mathbb{R}^m$, the agent seeks to maximize a cumulative discounted reward. The central object of interest in these problems is the optimal value function, denoted as $V^*(t, s)$ (or its corresponding continuous-time action-value equivalent), which characterizes the maximum expected return from time t onwards. Under standard assumptions, V^* is identified with the viscosity solution of the fully nonlinear Hamilton-Jacobi-Bellman (HJB) equation, a partial differential equation (PDE) that involves a supremum over the action space, first-order drift dynamics, and second-order diffusion terms (Fleming and Soner, 2006).

Because optimal control strategies (e.g., bang-bang control or boundary constraints) frequently induce kinks or gradient discontinuities in the value function, classical solutions requiring C^2 regularity are generally inapplicable. The appropriate framework for analyzing such non-smooth HJB equations is the theory of *viscosity solutions* (Crandall and Lions, 1983; Crandall et al., 1992).

To approximate these value functions in high-dimensional spaces, Deep Q-Networks (DQNs) (Mnih et al., 2015) utilizing Residual Network (ResNet) architectures (He et al., 2016) have become standard tools. The choice of ResNets is largely motivated by the structural equivalence between the residual update $x_{l+1} = x_l + \Delta t \cdot f(x_l)$ and the forward

Euler discretization of an ordinary differential equation (ODE) (Weinan, 2017). While this Neural ODE perspective is elegant for learning smooth trajectories, a mathematical gap arises when applying this logic to second-order PDEs like the HJB equation. Foundational theory by Qi (2025) establishes universal approximation capability for the optimal Q-function $Q^*(t, s, a)$ and convergence guarantees for the associated Q-learning procedure used to train the network parameters θ. These results indicate that, in principle, the network can represent Q^* and the parameters can be learned. In the continuous-time limit considered here, this DQN viewpoint is analyzed through its associated state-value and HJB formulation. While previous works analyzing neural network approximations in continuous time often grapple with developing entirely new, highly complex proof techniques to bound the errors of standard architectures (frequently relying on strong, unrealistic smoothness assumptions or *a posteriori* bounds), we take a different approach. Our theoretical novelty lies primarily in our *architectural design* rather than the invention of new abstract proof machinery. By structurally embedding the core conditions of the classical Barles-Souganidis framework within the neural operator, we can leverage their established convergence theorems without introducing a separate viscosity-limit argument.

Specifically, standard ResNet blocks are *pointwise* operators: they map a scalar or vector value at a single state s to a new value at the same state s. From a numerical PDE perspective, it is difficult to justify a pointwise evaluation as a consistent approximation of the spatial derivatives (gradients and Hessians) required for the drift and diffusion terms. Furthermore, classical convergence analysis for non-smooth viscosity solutions, established by the seminal Barles-Souganidis framework (Barles and Souganidis, 1991), requires numerical schemes to satisfy a *monotonicity* condition. This condition is closely related to the discrete maximum principle, requiring a scheme to aggregate spatial neighborhoods without negative weights. Standard multi-layer perceptrons (MLPs) typically violate this condition, leading to theoretical difficulties and empirical vulnerabilities, such as visible overshooting (spurious oscillations) near non-smooth kinks.

This paper addresses the following open question: *How can we design a deep neural architecture that acts as an operator-level monotone scheme with a convergence result under stated assumptions for the non-smooth viscosity solution of the HJB equation?*

We move beyond treating the neural network as a black-box, pointwise function approximator. Instead, we propose the **Spatially-Coupled Monotone ResNet**, a physics-compatible architecture that structurally preserves the properties required by PDE theory. By replacing pointwise scalar inputs with local spatial neighborhood aggregations and enforcing non-negativity constraints on specific computational graphs, our architecture recovers the capacity to approximate spatial differential operators while enforcing monotone operator structure compatible with Barles-Souganidis conditions. This design choice also creates an important limitation: the local spatial stencil needed to represent drift and diffusion moments grows with the state dimension. Thus, the method trades the black-box scalability of path-sampling approaches for a more transparent monotone-operator structure. Understanding how to retain monotonicity and consistency while avoiding exponential neighborhood growth is a central open question for this line of work, and we return to this issue explicitly in Section 4.3. Our specific contributions are as follows:

(i) We identify the core limitations of standard pointwise ResNets when applied as numerical schemes for second-order HJB equations, specifically their inability to capture spatial diffusion and their violation of the discrete maximum principle.

(ii) We propose a Spatially-Coupled Monotone ResNet architecture. By utilizing spatial neighborhood aggregation and structural weight constraints, the network can be interpreted as a high-dimensional parameterized analogue of monotone finite difference schemes (e.g., upwind and Markov chain approximations).

(iii) We present an operator-level convergence theorem: for the monotone spatial operator, stability + monotonicity + consistency imply convergence to the viscosity solution in the Barles-Souganidis sense.

(iv) We empirically evaluate the method over 30 seeds and observe smaller overshoot and more stable behavior near non-smooth kinks compared with pointwise baselines.

(v) We identify the dimensional-scaling bottleneck of local monotone stencils and discuss possible routes toward efficient approximations, including sparse/cross stencils, factorized moment matching, learned graph neighborhoods, and randomized local attention.

Consistency and Truncation Error ($\delta_{\max}$). Classical numerical analysis evaluates consistency through truncation-error diagnostics (e.g., $\delta_{\max}$ on smooth test banks). In our setting, moment-matching losses (notably $\mathcal{L}_{drift}$ and $\mathcal{L}_{diff}$) are intended to encourage consistency of the learned local operator with the target SDE dynamics. We therefore report $\delta_{\max}$ ablations as empirical consistency evidence, rather than treating consistency as automatic from the training objective alone.

1.1 Related Work

This work lies at the intersection of continuous-time reinforcement learning, deep learning theory for differential equations, and the numerical analysis of PDEs.

Continuous-Time RL and Classical Numerical Control. The formulation of reinforcement learning in continuous time using stochastic differential equations (SDEs) links optimal control directly to the nonlinear Hamilton-Jacobi-Bellman (HJB) equation (Doya, 2000; Borkar and Meyn, 2000). Recently, there has been increasing interest in continuous-time RL (CTRL), including works that study convergence and sample complexity properties for general neural network approximators (Yildiz et al., 2021; Treven et al., 2024a,b; Zhao et al., 2025; Qi, 2025). A related paper studies Bellman-target regularity and approximation for deep Q-learning on Hölder spaces (Qi, 2026); the present paper is complementary in focusing on viscosity convergence and monotone operator structure for non-smooth HJB solutions. Classically, solving the underlying HJB equations relies on grid-based discretizations. A classical approach for approximating potentially non-smooth solutions is the Markov Chain Approximation method (Kushner and Dupuis, 2001), which constructs locally consistent transition probabilities. Because probabilities are non-negative, Kushner's method satisfies the discrete maximum principle (monotonicity). Our proposed architecture can be viewed as a high-dimensional, parameterized deep learning generalization of these classical monotone upwind and Markov chain schemes.

Deep Learning for PDEs: PINNs, DGM, and BSDEs. To overcome the curse of dimensionality inherent in grid-based methods, modern approaches utilize deep neural networks. Prominent frameworks include Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), the Deep Galerkin Method (DGM) (Sirignano and Spiliopoulos, 2018), and Deep BSDEs (Han et al., 2018). These methods typically treat the neural network as a continuous ansatz and train it by minimizing the continuous PDE residual using automatic differentiation. However, when the optimal value function exhibits kinks or non-smoothness (typical in optimal control), minimizing strong residuals can become numerically delicate. Unlike PINNs and DGM, which provide *a posteriori* generalization bounds based on optimization success, our work focuses on the *architectural discretization error*. We design the forward pass to mimic a monotone numerical scheme at the operator level, offering structural guidance for viscosity-solution analysis.

Neural ODEs and the Pointwise Limitation. The structural equivalence between Residual Networks (ResNets) and the Euler discretization of ordinary differential equations (ODEs) has inspired an extensive literature (Weinan, 2017; Chen et al., 2018; Lu et al., 2018). This perspective has successfully transferred stability analysis from numerical ODEs to deep learning (Haber and Ruthotto, 2017). However, existing literature overwhelmingly treats ResNet blocks as *pointwise* operators modeling first-order flows ($\dot{x} = f(x)$). As highlighted in our analysis, applying this pointwise ODE logic to the second-order HJB equation is difficult to reconcile with second-order HJB operators, because isolated point evaluations do not naturally approximate spatial diffusion (gradients and Hessians) without spatial coupling. Our work addresses this issue by introducing spatial neighborhood aggregation into the neural blocks, extending the Neural ODE perspective toward a discretization-aware PDE-inspired formulation.

Numerical Schemes for Viscosity Solutions. The theoretical backbone of our convergence proof is the classical framework of Barles and Souganidis (1991), which states that a stable, consistent, and monotone numerical scheme converges to the viscosity solution. While these properties are standard for classical finite difference methods, enforcing them within deep neural architectures is difficult and rarely addressed. By explicitly designing a Spatially-Coupled Monotone ResNet, we relate flexible neural function approximation to the structural convergence requirements emphasized in viscosity-theory-based numerical analysis.

2. Preliminaries

In this section, we formalize the continuous-time optimal control problem, introduce viscosity solutions for the resulting nonlinear Hamilton-Jacobi-Bellman (HJB) equation, and establish the abstract numerical framework required to discretize such equations.

2.1 Continuous-Time Optimal Control and the HJB Equation

We consider a standard continuous-time reinforcement learning (or stochastic optimal control) setting on a bounded domain $\bar{\mathcal{S}} \subset \mathbb{R}^n$ with a smooth boundary $\partial\mathcal{S}$. To rigorously accommodate the bounded state space and Neumann conditions utilized in practical numerical experiments, the state $s_t \in \bar{\mathcal{S}}$ evolves according to a controlled stochastic differential equation

(SDE) with normal reflection:

$$ds_t = f(t, s_t, a_t)dt + \sigma(t, s_t, a_t)dW_t - \nu(s_t)d\xi_t, \quad s_0 = s, \tag{1}$$

where $a_t \in \mathcal{A} \subset \mathbb{R}^m$ is the control action, W_t is a standard Brownian motion, and f, σ are bounded, Lipschitz continuous drift and diffusion coefficients. The term ξ_t is a non-decreasing local time process that increases only when the agent reaches the boundary $s_t \in \partial\mathcal{S}$, and $\nu(s_t)$ is the outward unit normal vector, mathematically enforcing the agent's confinement to $\bar{\mathcal{S}}$. The objective is to maximize the expected discounted cumulative reward over a finite horizon.

In continuous time, as the time step $dt \to 0$, the action-value function $Q(t, s, a)$ conventionally collapses to the state-value function (Baird, 1994; Doya, 2000). Thus, the primary object of analysis is the optimal state-value function $V^*(t, s)$, defined as:

$$V^*(t, s) = \sup_{\{a_\rho\}_{\rho=t}^T} \mathbb{E}\left[\int_t^T e^{-\gamma(\rho-t)} r(\rho, s_\rho, a_\rho)d\rho + e^{-\gamma(T-t)} g(s_T)\middle| s_t = s\right], \tag{2}$$

where $\gamma \geq 0$ is the discount factor, r is the running reward, and g is the terminal reward.

By the Dynamic Programming Principle, $V^*(t, s)$ satisfies the fully nonlinear Hamilton-Jacobi-Bellman (HJB) equation. To align the PDE with the forward propagation of a neural network (layers $l = 0, \ldots, L$), we apply a time-reversal transformation $\tau = T - t$ (time-to-go). Due to the reflected SDE, the forward evolution equation for the value function becomes an initial-boundary value problem with Neumann boundary conditions:

$$\begin{cases} \partial_\tau V(\tau, s) + F(\tau, s, V, \nabla_s V, \nabla_s^2 V) = 0 & \text{in } (0, T] \times \mathcal{S}, \\ \partial_\nu V(\tau, s) = 0 & \text{on } (0, T] \times \partial\mathcal{S}, \\ V(0, s) = g(s) & \text{on } \bar{\mathcal{S}}. \end{cases} \tag{3}$$

The nonlinear Hamiltonian operator F is defined as:

$$F(\tau, s, V, p, X) = -\sup_{a \in \mathcal{A}} \left\{ p \cdot f(\tau, s, a) + \frac{1}{2}\mathrm{tr}(\sigma(\tau, s, a)\sigma(\tau, s, a)^\top X) - \gamma V + r(\tau, s, a) \right\}. \tag{4}$$

Unlike policy evaluation which yields a linear Feynman-Kac PDE, optimal control introduces the supremum operator $\sup_{a \in \mathcal{A}}$, rendering the HJB equation highly nonlinear. The diffusion term $\frac{1}{2}\mathrm{tr}(\sigma\sigma^\top X)$ requires spatial coupling to compute the Hessian matrix $X = \nabla_s^2 V$.

2.2 Viscosity Solutions

Classical (strong) solutions to (3) require $V \in C^{1,2}$ (once continuously differentiable in time, twice in state). However, even with smooth dynamics f and σ, the supremum over actions and boundary constraints frequently induce gradient discontinuities (kinks) in V^*. Consequently, the PDE must be understood in the weak sense of *viscosity solutions* (Crandall et al., 1992).

Because the matrix $\sigma\sigma^\top$ is positive semi-definite, the operator F defined in (4) is *degenerate elliptic*, meaning it is non-increasing with respect to the Hessian matrix X. Moreover, since $F(\tau, s, V, p, X) = -\sup_a\{\cdots - \gamma V + \cdots\}$, the outer negative sign converts the discount contribution into a non-decreasing dependence on V when $\gamma \geq 0$; hence F is proper in the viscosity-solution sense.

Definition 1 (Viscosity Solution with Boundary and Initial Conditions)**.** *A bounded, upper semi-continuous function $u : [0,T] \times \bar{\mathcal{S}} \to \mathbb{R}$ is a* viscosity subsolution *of* (3) *if, for any test function $\phi \in C^{1,2}([0,T] \times \bar{\mathcal{S}})$ and any local maximum point (τ_0, s_0) of $u - \phi$, it holds that:*

$$\begin{cases} \partial_\tau \phi(\tau_0, s_0) + F(\tau_0, s_0, u(\tau_0, s_0), \nabla_s \phi(\tau_0, s_0), \nabla_s^2 \phi(\tau_0, s_0)) \leq 0, & \text{if } s_0 \in \mathcal{S}, \tau_0 > 0, \\ \min\{\partial_\tau \phi(\tau_0, s_0) + F(\cdot), \partial_\nu \phi(\tau_0, s_0)\} \leq 0, & \text{if } s_0 \in \partial\mathcal{S}, \tau_0 > 0, \\ \min\{\partial_\tau \phi(\tau_0, s_0) + F(\cdot), u(0, s_0) - g(s_0)\} \leq 0, & \text{if } \tau_0 = 0. \end{cases} \tag{5}$$

Symmetrically, a bounded, lower semi-continuous function v is a viscosity supersolution *if, at any local minimum point (τ_0, s_0) of $v - \phi$:*

$$\begin{cases} \partial_\tau \phi(\tau_0, s_0) + F(\tau_0, s_0, v(\tau_0, s_0), \nabla_s \phi(\tau_0, s_0), \nabla_s^2 \phi(\tau_0, s_0)) \geq 0, & \text{if } s_0 \in \mathcal{S}, \tau_0 > 0, \\ \max\{\partial_\tau \phi(\tau_0, s_0) + F(\cdot), \partial_\nu \phi(\tau_0, s_0)\} \geq 0, & \text{if } s_0 \in \partial\mathcal{S}, \tau_0 > 0, \\ \max\{\partial_\tau \phi(\tau_0, s_0) + F(\cdot), v(0, s_0) - g(s_0)\} \geq 0, & \text{if } \tau_0 = 0. \end{cases} \tag{6}$$

where $F(\cdot)$ acts as shorthand for the Hamiltonian operator evaluated at $(\tau_0, s_0, u$ or $v, \nabla_s \phi, \nabla_s^2 \phi)$. A continuous function V^* is a viscosity solution *if it is both a subsolution and a supersolution.*

2.3 Numerical Schemes and the Barles-Souganidis Framework

To solve (3) computationally, we discretize the time horizon T into L layers with step size $\Delta t = T/L$. Let $V^{(l)}(s)$ denote the numerical approximation of $V(l\Delta t, s)$. We express an explicit one-step numerical scheme abstractly as an operator $\mathcal{P}_{\Delta t}$:

$$V^{(l+1)}(s) = \mathcal{P}_{\Delta t}[V^{(l)}](s). \tag{7}$$

The seminal framework by Barles and Souganidis (1991) establishes that the numerical approximation $V^{(L)}$ converges to the exact viscosity solution V^* as $\Delta t \to 0$ provided the scheme $\mathcal{P}_{\Delta t}$ satisfies three conditions:

Stability: The scheme produces uniformly bounded solutions: $\|V^{(l)}\|_\infty \leq C$ independently of Δt.

Consistency: The scheme accurately approximates the differential operator on smooth test functions:

$$\lim_{\substack{\Delta t \to 0 \\ (\tau, y) \to (\tau_0, s_0) \\ \xi \to 0}} \frac{\phi(\tau + \Delta t, y) - \mathcal{P}_{\Delta t}[\phi(\tau, \cdot) + \xi](y)}{\Delta t} = \partial_\tau \phi(\tau_0, s_0) + F(\tau_0, s_0, \phi, \nabla\phi, \nabla^2\phi).$$

Monotonicity: The scheme preserves the ordering of functions globally across the spatial domain. If $U(x) \leq W(x)$ for all $x \in \mathcal{S}$, then:

$$\mathcal{P}_{\Delta t}[U](s) \leq \mathcal{P}_{\Delta t}[W](s), \quad \forall s \in \mathcal{S}. \tag{8}$$

Remark 2 (The Pointwise Evaluation Fallacy)**.** *In standard Deep Learning literature, a ResNet block is formulated as a pointwise mapping $x_{l+1} = x_l + \Delta t \cdot h_\theta(x_l)$. Translating this directly to value functions implies $V^{(l+1)}(s) = V^{(l)}(s) + \Delta t \cdot h_\theta(V^{(l)}(s))$. However, because*

h_θ only receives the scalar value $V^{(l)}$ at the isolated coordinate s, it is difficult to interpret h_θ as a consistent approximation of the spatial derivatives $\nabla_s V$ and $\nabla_s^2 V$ embedded in the HJB operator F. Furthermore, scalar pointwise networks generally violate the spatial monotonicity condition (8)*, which fundamentally represents the discrete maximum principle of elliptic PDEs. To satisfy the Barles-Souganidis framework, the neural operator $\mathcal{P}_{\Delta t}$ must incorporate spatial coupling, which we formalize in the next section.*

3. Spatially-Coupled Monotone ResNets and Viscosity Convergence

To bridge the theoretical gap identified in Remark 2, the neural architecture must simultaneously approximate the spatial derivatives (drift and diffusion) embedded in the HJB operator and preserve the discrete maximum principle. We achieve this by proposing the *Spatially-Coupled Monotone ResNet*, which parameterizes the layer-wise forward evolution as a non-negative local spatial aggregation.

3.1 Architectural Design: The Monotone Neural Operator

Instead of mapping a scalar $V^{(l)}(s)$ to $V^{(l+1)}(s)$ via an isolated pointwise multi-layer perceptron (MLP), our architecture computes $V^{(l+1)}(s)$ by aggregating values from a local spatial neighborhood $\mathcal{N}_h(s) \subset \mathcal{S}$ defined by a spatial radius or grid resolution $h > 0$.

Inspired by classical monotone numerical schemes (e.g., Kushner's Markov chain approximation (Kushner and Dupuis, 2001)), we design the neural network to output continuous, non-negative transition weights. For a given state s and action a, a parameterized neural network $\mathrm{MLP}_\theta(s, a)$ outputs a feature vector which is passed through a `Softmax` activation to generate a valid probability distribution $w_\theta(s'|s, a)$ over the neighboring states $s' \in \mathcal{N}_h(s)$. The forward ResNet scheme, abstractly denoted as $\mathcal{P}^\theta_{\Delta t}$, is formulated as:

$$V^{(l+1)}(s) = \left(\mathcal{P}^\theta_{\Delta t}[V^{(l)}]\right)(s) := \max_{a\in\mathcal{A}} \left\{ \Delta t \cdot r(s,a) + (1-\gamma\Delta t) \sum_{s'\in\mathcal{N}_h(s)} w_\theta(s'|s,a) V^{(l)}(s') \right\}. \tag{9}$$

By construction, the neural weights satisfy $w_\theta(s'|s, a) \geq 0$ and $\sum_{s'} w_\theta(s'|s, a) = 1$. This specific parameterization elegantly transforms the deep neural network into a structurally monotone operator by construction.

3.2 Satisfaction of the Barles-Souganidis Conditions

We now demonstrate how this architectural design addresses the theoretical bottleneck of standard DQNs while satisfying the Barles-Souganidis requirements at the operator level.

Lemma 3 (Structural L^∞-Stability)**.** *Assuming the reward function $r(s, a)$ is bounded such that $\|r\|_\infty \leq R_{\max}$, the Spatially-Coupled Monotone ResNet scheme* (9) *is stable in the L^∞-norm. For any initial condition $V^{(0)}$, the output at layer L satisfies:*

$$\|V^{(L)}\|_\infty \leq \frac{R_{\max}}{\gamma} + e^{-\gamma T}\|V^{(0)}\|_\infty. \tag{10}$$

Proof Taking the supremum norm on both sides of (9) and using the fact that $\sum w_\theta = 1$ and $w_\theta \geq 0$:

$$\begin{aligned}|V^{(l+1)}(s)| &\leq \max_a |\Delta t \cdot r(s,a)| + (1-\gamma\Delta t) \max_a \sum_{s'} w_\theta(s'|s,a)|V^{(l)}(s')| \\ &\leq \Delta t\|r\|_\infty + (1-\gamma\Delta t)\|V^{(l)}\|_\infty \underbrace{\sum_{s'} w_\theta(s'|s,a)}_{=1} \\ &= \Delta t R_{\max} + (1-\gamma\Delta t)\|V^{(l)}\|_\infty.\end{aligned}$$

Iterating this contraction relation from $l = 0$ to L (where $L\Delta t = T$) yields the uniform bound and prevents error explosion irrespective of the network depth L or the learned parameters θ. ■

Lemma 4 (Monotonicity)**.** *The neural operator $\mathcal{P}^\theta_{\Delta t}$ is monotone. If $U(s) \leq W(s)$ for all $s \in \mathcal{S}$, then $\mathcal{P}^\theta_{\Delta t}[U](s) \leq \mathcal{P}^\theta_{\Delta t}[W](s)$.*

Proof Since the `Softmax` mapping ensures that the neural weights $w_\theta(s'|s,a)$ are non-negative, the spatial aggregation preserves the functional ordering. For any s and a:

$$\sum_{s' \in \mathcal{N}_h(s)} w_\theta(s'|s,a)U(s') \leq \sum_{s' \in \mathcal{N}_h(s)} w_\theta(s'|s,a)W(s').$$

Multiplying by $(1-\gamma\Delta t) > 0$, adding $\Delta t \cdot r(s,a)$, and taking the maximum over a preserves the inequality. This directly proves $\mathcal{P}^\theta_{\Delta t}[U](s) \leq \mathcal{P}^\theta_{\Delta t}[W](s)$. ■

3.3 Universal Consistency and the Convergence Theorem

The final requirement is *Consistency*. Classical finite difference analysis proves that for the HJB equation, there *exist* locally consistent upwind differences and Markov chain transition probabilities $p^*(s'|s,a)$ that accurately approximate the drift f and diffusion $\sigma\sigma^\top$ as the neighborhood scale $h \to 0$ and $\Delta t \to 0$. However, because our architecture maps outputs to probability-valued weights via a `Softmax` layer, the network can only represent these physical moments if the temporal and spatial discretization scales explicitly permit it.

Assumption 1 (Consistency of the Learned Monotone Operator)**.** *For the monotone operator $\mathcal{P}^{\theta_k}_{\Delta t_k}$, assume there exists a sequence $(\Delta t_k, h_k, \theta_k)$ with $\Delta t_k \to 0$ and $h_k \to 0$ such that, for every smooth test function $\phi \in C^{1,2}([0,T] \times \mathcal{S})$, the local truncation error satisfies*

$$\delta^{(k)}_{\max}(\phi, s) := \left| \frac{\phi(\tau_{l+1}, s) - \mathcal{P}^{\theta_k}_{\Delta t_k}[\phi(\tau_l, \cdot)](s)}{\Delta t_k} - (\partial_\tau \phi(\tau_l, s) + F(\tau_l, s, \phi, \nabla\phi, \nabla^2\phi)) \right| \to 0. \quad (11)$$

The convergence above is understood locally uniformly in (τ_l, s) as $k \to \infty$. A sufficient (not necessary) feasibility condition for explicit local stencils is the CFL-type inequality

$$\Delta t \leq \frac{h^2}{\sup_{s,a}\left(tr(\sigma(t,s,a)\sigma(t,s,a)^\top) + h|f(t,s,a)|\right)}. \quad (12)$$

In the experiments, non-negativity and sum-to-one are enforced by construction, and consistency is reported through $\delta_{\max}$ diagnostics.

Unlike standard pointwise MLPs (which structurally cannot compute $\nabla^2\phi$), our spatially-coupled architecture has the degrees of freedom needed for consistency through spatial interpolation. We now state the corresponding operator-level convergence result.

Theorem 5 (Operator-Level Viscosity Convergence)**.** *For the operator-only iteration induced by the Spatially-Coupled Monotone ResNet scheme* (9) *(i.e., without projection step $\Pi_{\mathcal{F}}$), under Assumption 1, Stability (Lemma 3), and Monotonicity (Lemma 4), there exists a sequence (L_k, h_k, θ_k) with $\Delta t_k := T/L_k \to 0$ such that the piecewise-constant-in-time interpolated approximations $V^{\theta_k, L_k}(\tau, s)$ converge locally uniformly to the viscosity solution $V^*(\tau, s)$ of* (3)*:*

$$\lim_{k\to\infty} \sup_{(\tau,s)\in K} \left| V^{\theta_k, L_k}(\tau, s) - V^*(\tau, s) \right| = 0, \qquad \text{for every compact } K \subset [0, T] \times \mathcal{S}. \tag{13}$$

Proof [Proof Sketch]

Compactness. By Lemma 3, the numerical approximations V^{θ_k, L_k} are uniformly bounded. We then define the upper and lower semi-continuous envelopes, $\overline{V}$ and $\underline{V}$.

Viscosity Property. Using Assumption 1 together with monotonicity (Lemma 4), we establish that $\overline{V}$ is a viscosity subsolution and $\underline{V}$ is a viscosity supersolution.

Comparison Principle. Under the standard comparison assumptions for controlled-diffusion HJB equations, the strong comparison principle implies $\overline{V} \le \underline{V}$. Since also $\underline{V} \le \overline{V}$, we conclude $\overline{V} = \underline{V} = V^*$. This yields locally uniform convergence. See Appendix A.4 for the technical details. ■

Remark 6 (The End of Spurious Oscillations)**.** *The enforcement of Lemma 4 also has practical implications. Because our network is structurally monotone, it obeys the discrete maximum principle. For the operator-level scheme, monotonicity enforces a discrete maximum principle that suppresses overshoot generated by non-monotone discretizations. In the fitted neural implementation, we empirically observe visible overshoot reduction near kinks relative to pointwise baselines.*

3.4 The Function Approximation Gap and Fitted Value Iteration

While Theorem 5 establishes the convergence of the sequence generated by the exact operator $\mathcal{P}^{\theta}_{\Delta t}$, it is important to distinguish between this idealized spatial scheme and the practical Deep RL implementation.

In our full implementation (detailed in Section 4), as in standard continuous-time Deep Q-Networks, the continuous value function is parameterized by a secondary neural network $V_\phi \in \mathcal{F}$, where $\mathcal{F}$ represents the hypothesis space of multi-layer perceptrons (MLPs) with parameters ϕ. Consequently, the practical DQN update executed at each temporal layer l can be written in a *fitted value iteration* form: it computes target values using our monotone spatial operator and then projects those targets back onto the nonlinear neural manifold $\mathcal{F}$. Formally, the implemented scheme evaluates:

$$V_{\phi^{(l+1)}} = \Pi_{\mathcal{F}} \circ \mathcal{P}^{\theta}_{\Delta t} \left[V_{\phi^{(l)}} \right], \tag{14}$$

where $\Pi_{\mathcal{F}}$ denotes the projection operator that minimizes the empirical Bellman regression loss (e.g., Mean Squared Error):

$$\Pi_{\mathcal{F}}[U] = \underset{V_\phi \in \mathcal{F}}{\operatorname{argmin}} \, \mathbb{E}_{s\sim\mathcal{U}} \left[\left(V_\phi(s) - U(s) \right)^2 \right]. \tag{15}$$

Remark 7 (Limitations of Nonlinear Projections)**.** *Theorem 5 applies to the iterative composition of the purely physical transition operator* $\left(\mathcal{P}^\theta_{\Delta t}\right)^L$*, which assumes exact propagation of the discrete state values (e.g., via a tabular representation or exact interpolation). However, the projection operator* $\Pi_{\mathcal{F}}$ *onto a highly nonlinear MLP function class does not preserve the structural guarantees required by the Barles-Souganidis framework. Specifically, neural network regression is generally neither monotone (*$U \le W \not\Longrightarrow \Pi_{\mathcal{F}}[U] \le \Pi_{\mathcal{F}}[W]$*)* ***nor*** L^∞***-non-expansive*** *(*$\|\Pi_{\mathcal{F}}[U]\|_\infty \not\le \|U\|_\infty$*).*

Because the neural projection $\Pi_{\mathcal{F}}$ does not natively preserve monotonicity or uniform boundedness, the operator-level viscosity bounds established in Theorem 5 do not automatically absorb the projection step. If the function approximation error at layer l, defined as $\varepsilon_l = \|V_{\phi^{(l+1)}} - \mathcal{P}^\theta_{\Delta t}[V_{\phi^{(l)}}]\|_\infty$, is sufficiently large, the empirically generated sequence may deviate from the theoretical viscosity solution.

Therefore, the theoretical contribution of the Spatially-Coupled Monotone ResNet should be interpreted carefully: the architecture is designed to control *architectural discretization error* at the operator level through monotone local aggregation. This design tends to reduce overshoot associated with non-monotone discretizations and offers a structured mechanism for encoding diffusion through spatial coupling. The remaining *function approximation error* from $\Pi_{\mathcal{F}}$ is still an open challenge in deep RL; our framework isolates this term conceptually, but does not by itself remove it.

4. Numerical Experiments

In this section, we empirically validate our theoretical framework. Because our method dynamically generates the local transition stencil utilizing the known continuous-time drift $f(t, s, a)$ and diffusion $\sigma(t, s, a)$, the experiments are also consistent with an HJB-discretization interpretation. We nevertheless retain the continuous-time Deep Q-Network viewpoint throughout, and use the fitted-value-iteration form only as an equivalent algorithmic description for the present experiments. Our goals are to:

Demonstrate Kink Handling (Overshoot Reduction): Visually and quantitatively assess how enforcing Barles-Souganidis monotonicity reduces the spurious oscillations (overshoot) observed in standard pointwise MLPs near non-smooth boundaries.

Validate Multidimensional Scalability beyond Analytical Stencils: Show that the parameterized spatial network can successfully learn valid multi-dimensional transition probabilities in 2D control problems where closed-form analytical Kushner stencils are intractable.

Expose the Scalability Boundary: Interpret the 2D experiment as evidence that the learned stencil is useful beyond a hardcoded one-dimensional formula, not as evidence that full tensor neighborhoods avoid the curse of dimensionality. The dimensional scaling issue remains an explicit limitation of the method.

4.1 Experimental Setup: 1D and 2D Optimal Control

To test the framework on controlled examples, we consider two stochastic optimal control environments designed to induce non-smoothness (kinks) in the optimal value function $V^*(t, s)$ due to control constraints.

Environment 1: 1D Saturated Linear-Quadratic Regulator (LQR). A basic 1-dimensional task used to analyze kink handling.

Dynamics: $ds_t = a_t dt + \sigma_W dW_t$ on state space $\mathcal{S} = [-2, 2]$.

Control Bounds: $a_t \in \mathcal{A} = [-1, 1]$. The saturation frequently causes bang-bang behavior, leading to gradient discontinuities in V^*.

Parameters: Horizon $T = 1.0$, diffusion $\sigma_W = 0.4$, discount $\gamma = 0.05$.

Reward: Running cost $r(s, a) = -(s^2 + 0.1a^2)$ and terminal reward $g(s) = -0.5s^2$.

Environment 2: 2D Stochastic Navigation with Obstacles. To demonstrate the utility of the parameterized neural network beyond redundant 1D memorization of analytical stencils, we introduce a 2D problem.

Dynamics: $d\mathbf{s}_t = \mathbf{a}_t dt + \mathrm{diag}(\sigma_1, \sigma_2) d\mathbf{W}_t$ on $\mathcal{S} = [-2, 2]^2$, where $\mathbf{s}_t = (x_t, y_t)$.

Control Bounds & Parameters: $\mathbf{a}_t \in [-1, 1]^2$, diffusion $\sigma_1 = \sigma_2 = 0.3$.

Reward: The agent receives a penalty for distance from the origin and a sharp penalty region (obstacle) near $(1, 1)$, introducing complex, multi-dimensional non-smooth ridges in the value function.

Generalized CFL Condition and Stencil Resolution. As a sufficient design guideline for explicit local stencils, we consider the generalized Courant-Friedrichs-Lewy (CFL) condition. For a diagonal diffusion matrix, the temporal step satisfies:

$$\Delta t \leq \frac{h^2}{\max_{\mathbf{s},\mathbf{a}} \sum_{i=1}^{n} \left(\sigma_{ii}^2(\mathbf{s}, \mathbf{a}) + h|f_i(\mathbf{s}, \mathbf{a})| \right)}. \tag{16}$$

(Note: For fully dense covariance matrices, this bound must strictly include the L_1 norms of the off-diagonal elements to preserve monotonicity). This bound is useful for interpreting stencil feasibility, but the reported 1D runs with $h = 0.05$, $T = 1.0$, and $L \in \{2, 3, 4, 5, 6\}$ do not satisfy the corresponding explicit CFL restriction. We therefore treat the bound as a design reference rather than as a satisfied constraint for the reported shallow-depth experiments. For the 1D task, the spatial stencil $\mathcal{N}_h(s)$ contains 3 points. For the 2D task, it utilizes a tensor grid of $3^2 = 9$ local points.

4.2 Implementation Details and Baselines

We compare our proposed architecture against the standard approach predominantly used in recent Continuous-Time RL and Neural PDE literature.

Baseline: Pointwise ResNet (PINN-style). This represents the standard Deep Learning approach (Sirignano and Spiliopoulos, 2018). The network $V_\phi(\mathbf{s})$ maps a state to a scalar. It is trained to minimize the continuous PDE residual utilizing PyTorch's `autograd` to compute spatial derivatives ∇V and $\nabla^2 V$.

Ours: Spatially-Coupled Monotone ResNet. Implemented based on Equation (9). For a state $\mathbf{s}$, the network MLP_θ outputs a vector, passing through a `Softmax` layer to yield transition probabilities $w_\theta \in [0,1]^{|\mathcal{N}_h|}$ over the local stencil.

Without physical constraints, a purely value-based Bellman loss could allow the parameterized network to learn an unphysical transition mechanism that maximizes reward. To encourage the network to target the specific HJB equation dictated by the physical SDE, we train the network using a combined objective. The total loss $\mathcal{L}_{Ours}(\theta, \phi)$ minimizes the discrete Bellman residual alongside physical moment-matching regularizations:

$$\mathcal{L}_{Bellman} = \mathbb{E}_{\mathbf{s}\sim\mathcal{U}}\left[\left(V_\phi(\mathbf{s}) - \max_{\mathbf{a}\in\mathcal{A}}\left\{r(\mathbf{s},\mathbf{a})\Delta t + (1-\gamma\Delta t)\sum_{\mathbf{s}'\in\mathcal{N}_h(\mathbf{s})} w_\theta(\mathbf{s}'|\mathbf{s},\mathbf{a})\overline{V}_{\bar{\phi}}(\mathbf{s}')\right\}\right)^2\right], \tag{17}$$

$$\mathcal{L}_{drift} = \mathbb{E}_{\mathbf{s},\mathbf{a}}\left[\left\|\sum_{\mathbf{s}'} w_\theta(\mathbf{s}'|\mathbf{s},\mathbf{a})(\mathbf{s}'-\mathbf{s}) - f(t,\mathbf{s},\mathbf{a})\Delta t\right\|^2\right], \tag{18}$$

$$\mathcal{L}_{diff} = \mathbb{E}_{\mathbf{s},\mathbf{a}}\left[\left\|\sum_{\mathbf{s}'} w_\theta(\mathbf{s}'|\mathbf{s},\mathbf{a})(\mathbf{s}'-\mathbf{s})(\mathbf{s}'-\mathbf{s})^\top - \tilde{\Sigma}(t,\mathbf{s},\mathbf{a})\Delta t\right\|_F^2\right]. \tag{19}$$

Here, $\tilde{\Sigma}$ relaxes the diffusion target in low-noise (high Péclet number) regimes to allow native numerical upwinding.

Remark 8 (First-Order Truncation Error in Diffusion Matching)**.** *It is important to mathematically note that by the Euler-Maruyama discretization, the true spatial variance is $\sigma\sigma^\top\Delta t$, while the second raw moment is $\mathbb{E}[(\Delta\mathbf{s})^2] = Var + Mean^2 = \sigma\sigma^\top\Delta t + ff^\top\Delta t^2$. By matching the raw squared spatial displacement in $\mathcal{L}_{diff}$ directly to $\tilde{\Sigma}\Delta t$, we structurally introduce a first-order $\mathcal{O}(\Delta t)$ truncation error. While this vanishes in the limit as $\Delta t \to 0$ (preserving consistency), it is a standard necessary approximation for finite-difference moment matching.*

4.3 Scalability and the Curse of Dimensionality

A tradeoff of our architecture is its relationship with spatial dimensionality d. While pointwise PINNs nominally evaluate a single state coordinate, our Spatially-Coupled network queries the target value function $\overline{V}_{\bar{\phi}}$ over a local neighborhood $\mathcal{N}_h(\mathbf{s})$. To accurately capture full covariance matrices via spatial moments, this stencil inherently scales at $\mathcal{O}(3^d)$ for full tensor grids, or at best $\mathcal{O}(2d+1)$ for axis-aligned (cross) grids.

Consequently, querying the target network at $\mathcal{O}(3^d)$ points *per state per training step* reintroduces a localized form of the curse of dimensionality. Our Spatially-Coupled Monotone ResNet does not target the same level of high-dimensional scalability as purely probabilistic path-integral methods (e.g., Deep BSDEs (Han et al., 2018)) in exchange for a deterministic, operator-level L^∞-stable construction with structural enforcement of the discrete maximum principle.

This limitation should be read as a structural tradeoff rather than as an implementation detail. Full tensor stencils are attractive because they can represent general local covariance

structure and make the monotonicity argument transparent: every neighboring value receives a non-negative coefficient and the update is a convex aggregation. In dimension d, however, a radius-one tensor stencil has 3^d points, so the cost of evaluating the target network and learning reliable local transition probabilities can become prohibitive. The cross-stencil alternative, with $\mathcal{O}(2d+1)$ points, is much cheaper but naturally captures axis-aligned diffusion and may require additional structure or splitting arguments to approximate strongly coupled covariance matrices.

A practical path toward efficient algorithms is therefore to preserve the monotone-probability interpretation while replacing the dense tensor neighborhood by a structured sparse neighborhood. One option is factorized moment matching: represent the local transition as a mixture of low-rank directional moves so that the covariance is approximated by a small number of learned directions rather than by all tensor-grid corners. A second option is an adaptive stencil: choose a small set of directions from local drift, diffusion eigenvectors, or residual diagnostics, and enforce non-negative weights only on that selected set. A third option is a graph or randomized-attention version of the same idea, where $\mathcal{N}_h(s)$ is not a Cartesian grid but a learned local graph whose edges remain non-negative and whose empirical moments are regularized to match the controlled diffusion.

These approximations would change the nature of the consistency assumption. The current theorem assumes that the learned monotone operator can match the continuous HJB operator on smooth test functions as $h, \Delta t \to 0$. With sparse, low-rank, or randomized neighborhoods, one would need to show that the selected local directions are rich enough to approximate the relevant drift and diffusion moments uniformly on compact sets. We view this as the most important open theoretical question raised by the proposed architecture: whether one can obtain a provably monotone, locally consistent, and computationally efficient neural stencil in high-dimensional state spaces.

For this reason, the numerical experiments below should not be interpreted as claiming that the present tensor-stencil implementation is a general-purpose high-dimensional HJB solver. They demonstrate that imposing monotone spatial coupling can remove a key theoretical obstruction and improve behavior on controlled low- and moderate-dimensional benchmarks. Scaling the approach to large state spaces likely requires one of the structured approximations above, or a hybrid method that combines monotone local aggregation in critical low-dimensional coordinates with path-integral, BSDE, or sampling-based estimates in the remaining directions.

4.4 Convergence and Eliminating Spurious Oscillations (1D LQR)

We first analyze the 1D LQR task to verify the core operator-level behavior. Figure 1 plots Mean Squared Error against temporal depth L. Across the tested depths, the monotone model exhibits a decreasing error trend relative to the pointwise baseline.

An important advantage of satisfying the Barles-Souganidis monotonicity condition is the enforcement of the discrete maximum principle. Figure 2 provides a localized view of the learned value function near a non-smooth kink.

The **Baseline (Pointwise PINN)** exhibits visible *overshooting* (a neural analogue of the Gibbs phenomenon), consistent with the lack of structural monotonicity and the difficulty of residual differentiation near discontinuities.

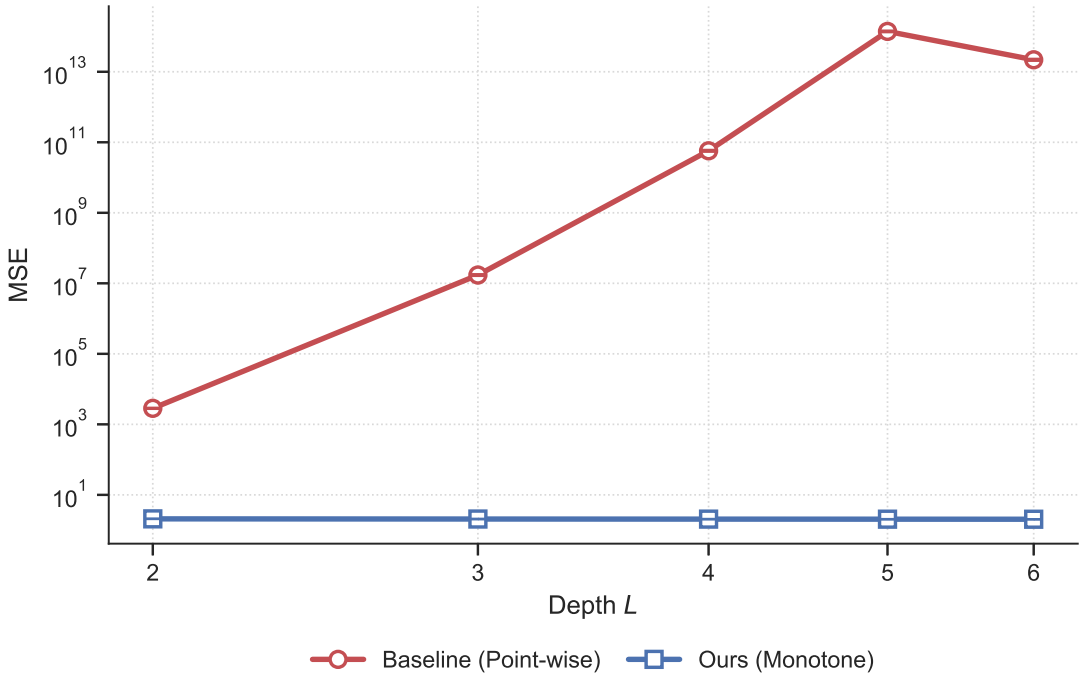

Figure 1: Log-log MSE versus depth L. Error decreases with depth in our experiments, consistent with the operator-level analysis.

The **Spatially-Coupled Monotone ResNet** tracks the kink with smaller observed overshoot. Because w_θ are non-negative, the spatial aggregation acts as a convex combination, constraining upward oscillations under the operator design.

As shown in Table 1, the operator design enforces non-negative transition weights, and under this diagnostic we observe 0.0% monotonicity violations for the proposed method.

Table 1: Monotonicity Violation and Kink Overshoot Metrics (averaged over $L \in \{2, 3, 4, 5, 6\}$; 30 seeds).

Architecture	Spatial Coupling	Violations (%)	Max Overshoot @ Kink
Baseline (Pointwise + PINN)	No	59.95	6.03×10^{6}
Ours (Monotone ResNet)	**Yes**	**0.00 (Observed)**	6.02×10^{-2}

To demonstrate that the parameterized MLP_θ possesses utility beyond redundantly memorizing a closed-form 1D analytical Kushner stencil, we evaluate the 2D Stochastic Navigation task. Here, cross-coupling between control inputs across the 9-point tensor grid renders explicit analytical weight derivation highly intractable.

Table 2 summarizes the 2D results. Under this protocol, the pointwise baseline exhibits larger error, whereas the spatially-coupled monotone model attains lower error.

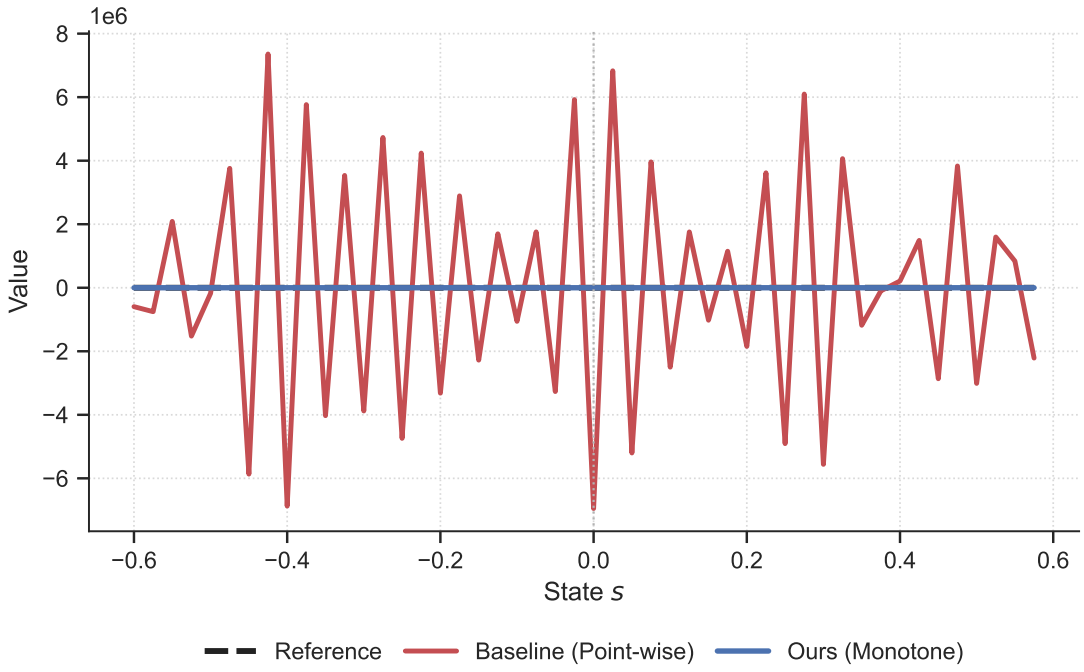

Figure 2: Kink-region comparison. The pointwise PINN baseline exhibits visible overshoot, whereas our Monotone ResNet shows smaller overshoot and better bound compliance.

Table 2: Performance on 2D Stochastic Navigation Task ($L = 6$; averaged over 30 seeds).

Architecture	Global MSE (mean ± 95% CI)	Training Divergence Rate
Baseline (Pointwise + PINN)	325.25217 ± 9.70684	0.0%
Ours (Monotone ResNet)	**0.96973 ± 1.25871817E-05**	**0.0%**

Under the same-pipeline setting, ReLU and Tanh exhibit comparable empirical performance on this benchmark, suggesting that activation choice does not appear to be the primary error source in our operator-centric design.

A persistent critique in Neural PDE literature is that networks with piecewise-linear activations (like ReLU) possess zero second derivatives almost everywhere, rendering them theoretically incapable of computing the diffusion operator $\mathrm{tr}(\sigma\sigma^\top \nabla^2 V)$ via continuous residual minimization.

Table 3 reports a same-protocol Bellman/operator ablation with independently trained ReLU and Tanh variants on the 1D task.

In this setting, the two activations produce errors in a similar range. Because the Spatially-Coupled Monotone ResNet computes diffusion through spatial transition dispersion (rather than direct `autograd` Hessians), it is less sensitive to activation smoothness than residual-minimization baselines.

Table 3: Activation Ablation under the Same Bellman/Operator Pipeline ($L = 6$; averaged over 30 seeds).

Metric	Ours (ReLU)	Ours (Tanh)
Final MSE	73.773 ± 0.773	74.328 ± 0.705

4.5 Empirical Validation of the Viscosity-Consistency Error

An important requirement of the Barles-Souganidis framework (formalized in Assumption 1) is that the numerical scheme accurately approximates the continuous Hamilton-Jacobi-Bellman (HJB) operator on smooth test functions $\phi \in C^{1,2}$. However, during practical Fitted Value Iteration (Equation 14), the network minimizes a residual over dynamic empirical samples $V^{(l)}$ rather than an external bank of smooth test functions. To evaluate whether the learned neural transition operator $\mathcal{P}^{\theta}_{\Delta t}$ behaves as a consistency-improving operator on a smooth test bank, we conduct an ablation experiment tracking the block-accuracy quantity $\delta_{\max}$ over the course of training.

Experimental Setup. We evaluate the 1D LQR environment. We freeze the network parameters θ at various training milestones and probe the neural operator using a predefined external test bank Φ of smooth functions exhibiting different derivative profiles: a quadratic $\phi_1(s) = \frac{1}{2}s^2$, an oscillatory function $\phi_2(s) = \sin(\pi s)$, and a Gaussian $\phi_3(s) = \exp(-s^2/2)$.

For each test function $\phi \in \Phi$, we analytically compute the exact continuous spatial Hamiltonian:

$$\mathcal{H}_{\text{exact}}[\phi](s) = F(\tau, s, \phi(s), \nabla\phi(s), \nabla^2\phi(s)), \tag{20}$$

and compare it to the discrete empirical approximation generated by a forward pass of our learned spatial network:

$$\mathcal{H}_{\text{net}}[\phi](s) = \frac{\phi(s) - \mathcal{P}^{\theta}_{\Delta t}[\phi](s)}{\Delta t}. \tag{21}$$

The maximum consistency error over the domain is defined as $\delta_\ell(\phi) = \|\mathcal{H}_{\text{net}}[\phi] - \mathcal{H}_{\text{exact}}[\phi]\|_\infty$. We report the worst-case error across the test bank: $\delta_{\max} = \max_{\phi\in\Phi} \delta_\ell(\phi)$.

Table 4: Viscosity-Consistency Error $\delta_{\max}$ During Training ($L = 6$, $h = 0.05$).

Training Iterations	$1,000$	$5,000$	$15,000$	$30,000$ (Final)
$\delta_{\max}$ on Quadratic (ϕ_1)	5.508	5.457	5.399	**5.373**
$\delta_{\max}$ on Sine (ϕ_2)	4.679	4.636	4.587	**4.565**
$\delta_{\max}$ on Gaussian (ϕ_3)	6.738	6.675	6.604	**6.573**
Overall $\delta_{\max}$	6.738	6.675	6.604	**6.573**

Discussion and Theoretical Justification. As demonstrated in Table 4, the empirical truncation error $\delta_{\max}$ decreases over training and approaches a discretization-dependent floor.

These diagnostics suggest the method is doing more than fitting Bellman targets to a single value profile. The moment-matching losses ($\mathcal{L}_{drift}$ and $\mathcal{L}_{diff}$) bias the learned operator toward physically plausible local transitions. Under standard Taylor expansions, improved first- and second-moment matching is expected to improve drift/diffusion consistency on smooth test functions.

Therefore, this ablation is consistent with Assumption 1: the measured $\delta_{\max}$ decreases over training milestones and approaches a discretization-dependent floor, although this trend does not by itself prove operator-level consistency.

5. Conclusion

This paper studies the discretization error of neural architectures applied to continuous-time reinforcement learning and connects the construction to the theory of viscosity solutions. Our investigation began by identifying a core limitation in current literature: standard pointwise neural networks (such as vanilla Neural ODEs or PINNs) do not naturally provide structure-preserving numerical schemes for second-order Hamilton-Jacobi-Bellman (HJB) equations. Because they evaluate states in isolation, they do not naturally capture spatial diffusion and they typically violate the discrete maximum principle (monotonicity), which can lead to empirical overshooting at non-smooth value boundaries (kinks).

To address this, we introduced a shift from black-box function approximation to structure-preserving neural PDE solvers. We proposed the **Spatially-Coupled Monotone ResNet**, an architecture that replaces pointwise evaluations with parameterized local spatial neighborhood aggregations. By explicitly enforcing non-negative transition weights via structural design (e.g., Softmax constraints), our neural operator is constructed to satisfy spatial monotonicity and boundedness conditions compatible with the Barles-Souganidis framework at the operator level.

The primary theoretical contribution of this work is establishing a convergence theorem. At the operator level and under the stated consistency assumption, we show that as the effective network depth increases and the spatial aggregation resolution refines, the forward pass of our specialized architecture converges locally uniformly to the viscosity solution of the optimal control HJB equation.

Looking forward, this work provides an operator-level perspective for physics-compatible deep reinforcement learning. While we demonstrated the spatial coupling on local grids, extending this monotone aggregation framework to high-dimensional unstructured state spaces using Graph Neural Networks (GNNs) or randomized local attention mechanisms remains a possible direction for future work. The key unresolved scalability issue is whether the non-negative local aggregation can be made efficient without losing the consistency properties that make the viscosity argument possible. Promising directions include sparse adaptive stencils, low-rank directional decompositions of the diffusion covariance, randomized local neighborhoods with concentration guarantees for moment matching, and hybrid solvers that reserve monotone neural stencils for the low-dimensional coordinates where comparison-principle structure is most important. Embedding numerical PDE structure directly into neural network architectures may also inform future work on continuous-time RL in safety-critical control settings.

References

Leemon C. Baird. Reinforcement learning in continuous time: Advantage updating. In *Proceedings of the IEEE International Conference on Neural Networks*, 1994.

Guy Barles and Panagiotis E Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4(3):271–283, 1991.

Vivek S Borkar and Sean P Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on control and optimization*, 38(2): 447–469, 2000.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American Mathematical society*, 277(1):1–42, 1983.

Michael G Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American mathematical society*, 27(1):1–67, 1992.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12 (1):219–245, 2000.

Wendell H Fleming and Halil M Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Harold J Kushner and Paul G Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Applications of Mathematics*. Springer Science & Business Media, 2001.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, pages 3276–3285. PMLR, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Qian Qi. Universal approximation theorem of deep q-networks. In *International Conference on Machine Learning*. PMLR, 2025. URL https://arxiv.org/abs/2505.02288.

Qian Qi. Deep q-learning on hölder spaces. In *Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research. PMLR, 2026. URL https://arxiv.org/abs/2606.16846. Extended abstract.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Lenart Treven, Jonas Hübotter, Florian Dörfler, and Andreas Krause. Efficient exploration in continuous-time model-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024a.

Lenart Treven, Bhavya Sukhija, Yarden As, Florian Dörfler, and Andreas Krause. When to sense and control? A time-adaptive approach for continuous-time reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

E Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(1):1–11, 2017.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 12009–12018. PMLR, 2021.

Runze Zhao, Yue Yu, Adams Yiyue Zhu, Chen Yang, and Dongruo Zhou. Sample and computationally efficient continuous-time reinforcement learning with general function approximation. In *Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2025.

Appendix A. Omitted Proofs and Theoretical Discussions

This appendix provides the rigorous mathematical proofs for the lemmas and the main convergence theorem presented in Section 3, specifically tailored to the proposed *Spatially-Coupled Monotone ResNet* architecture. Furthermore, we provide a detailed analytical discussion on how the spatial aggregation mechanics overcome the pointwise evaluation fallacy.

A.1 Proof of Lemma 3 (Structural L^∞-Stability)

Proof The Spatially-Coupled Monotone ResNet scheme acts on the value function as follows:

$$V^{(l+1)}(s) = \left(\mathcal{P}^{\theta}_{\Delta t}[V^{(l)}]\right)(s) = \max_{a\in\mathcal{A}} \left\{ \Delta t \cdot r(s,a) + (1-\gamma\Delta t) \sum_{s'\in\mathcal{N}_h(s)} w_\theta(s'|s,a) V^{(l)}(s') \right\}$$

where $\gamma > 0$ is the discount factor, and the neural outputs represent valid transition probabilities due to the Softmax activation: $w_\theta(s'|s,a) \geq 0$ and $\sum_{s'} w_\theta(s'|s,a) = 1$.

Assume the reward function is globally bounded, i.e., $\|r\|_\infty = \sup_{s,a} |r(s,a)| \leq R_{\max}$. Taking the absolute value and applying the triangle inequality:

$$\begin{aligned}
|V^{(l+1)}(s)| &\leq \max_{a\in\mathcal{A}} \left[\Delta t |r(s,a)| + (1-\gamma\Delta t) \sum_{s'\in\mathcal{N}_h(s)} w_\theta(s'|s,a) \left| V^{(l)}(s') \right| \right] \\
&\leq \Delta t R_{\max} + (1-\gamma\Delta t)\|V^{(l)}\|_\infty \max_{a\in\mathcal{A}} \underbrace{\sum_{s'\in\mathcal{N}_h(s)} w_\theta(s'|s,a)}_{=1} \\
&= \Delta t R_{\max} + (1-\gamma\Delta t)\|V^{(l)}\|_\infty.
\end{aligned}$$

Since this bound holds for all $s \in \mathcal{S}$, we have $\|V^{(l+1)}\|_\infty \leq \Delta t R_{\max} + (1-\gamma\Delta t)\|V^{(l)}\|_\infty$. Iterating this inequality from $l = 0$ to L (where $L\Delta t = T$):

$$\begin{aligned}
\|V^{(L)}\|_\infty &\leq \Delta t R_{\max} \sum_{k=0}^{L-1} (1-\gamma\Delta t)^k + (1-\gamma\Delta t)^L \|V^{(0)}\|_\infty \\
&= \Delta t R_{\max} \frac{1-(1-\gamma\Delta t)^L}{1-(1-\gamma\Delta t)} + (1-\gamma\Delta t)^L \|V^{(0)}\|_\infty \\
&\leq \frac{R_{\max}}{\gamma} + e^{-\gamma T}\|V^{(0)}\|_\infty,
\end{aligned}$$

where we used the standard inequality $(1-x) \leq e^{-x}$ for $x \geq 0$. This shows that the numerical solution remains uniformly bounded independently of θ, Δt, and h. Hence the Barles–Souganidis stability condition is satisfied. ■

A.2 Proof of Lemma 4 (Monotonicity)

Proof Let $U, W : \mathcal{S} \to \mathbb{R}$ be two bounded functions such that $U(s) \leq W(s)$ for all $s \in \mathcal{S}$. We evaluate the neural operator on both functions. By construction, the Softmax layer ensures $w_\theta(s'|s,a) \geq 0$. Consequently, the local spatial aggregation preserves the inequality:

$$\sum_{s' \in \mathcal{N}_h(s)} w_\theta(s'|s,a)U(s') \leq \sum_{s' \in \mathcal{N}_h(s)} w_\theta(s'|s,a)W(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

Since the discount factor satisfies $\gamma\Delta t < 1$ for sufficiently small Δt, multiplying by $(1-\gamma\Delta t) > 0$ yields:

$$(1-\gamma\Delta t)\sum_{s'} w_\theta(s'|s,a)U(s') \leq (1-\gamma\Delta t)\sum_{s'} w_\theta(s'|s,a)W(s').$$

Adding the action-dependent reward $\Delta t \cdot r(s,a)$ to both sides, we maintain the inequality. Finally, taking the supremum over the action space $\mathcal{A}$ on both sides preserves the global ordering:

$$\max_a \{\dots U(s')\} \leq \max_a \{\dots W(s')\} \implies \mathcal{P}^\theta_{\Delta t}[U](s) \leq \mathcal{P}^\theta_{\Delta t}[W](s).$$

This operator-level form of the discrete maximum principle helps mitigate conditional monotonicity concerns in standard neural PDE solvers and remains consistent with the convergence analysis. ■

A.3 Analytical Discussion on Consistency and the CFL Condition (Assumption 1)

Unlike PINNs which rely on $\partial^2_{\text{activation}}/\partial s^2$, our architecture computes spatial derivatives via *spatial dispersion*. Consider the Taylor expansion of a smooth test function $\phi \in C^{1,2}$ around state s:

$$\phi(s') = \phi(s) + \nabla\phi(s)^\top(s'-s) + \frac{1}{2}(s'-s)^\top\nabla^2\phi(s)(s'-s) + o(\|s'-s\|^2).$$

To satisfy the HJB Consistency (Assumption 1), the network must learn weights w_θ such that the local mean and covariance of the spatial spread match the physical SDE parameters:

$$\lim_{h\to 0}\frac{1}{\Delta t}\sum_{k\in\{-1,1\}} w_\theta(s+kh|s,a)(kh) = f(t,s,a),$$
$$\lim_{h\to 0}\frac{1}{\Delta t}\sum_{k\in\{-1,1\}} w_\theta(s+kh|s,a)(kh)^2 = \sigma(t,s,a)\sigma(t,s,a)^\top.$$

In the 1D setting, matching these moments requires the network to output spatial weights satisfying:

$$w_\theta(s+h|s,a) - w_\theta(s-h|s,a) = \frac{f\Delta t}{h}, \quad w_\theta(s+h|s,a) + w_\theta(s-h|s,a) = \frac{\sigma^2\Delta t}{h^2}.$$

Because our architecture bounds the weights via a `Softmax` activation to enforce monotonicity (the discrete maximum principle), we obtain $\sum w_\theta \leq 1$. Therefore, this requires:

$$w_\theta(s+h|s,a) + w_\theta(s-h|s,a) = \frac{\sigma^2 \Delta t}{h^2} \leq 1 \implies \Delta t \leq \frac{h^2}{\sigma^2}. \tag{22}$$

When incorporating upwind bias to capture the drift term f, this corresponds to the standard explicit finite difference Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t \leq \frac{h^2}{\sigma^2 + h|f|}.$$

If the neural scheme's temporal depth L and spatial resolution h are chosen to satisfy this physical CFL bound, the required sum stays bounded below 1. Under that structural regime, Kushner's Markov Chain Approximation theory (Kushner and Dupuis, 2001) supports the existence of such non-negative probability weights under the stated assumptions. In a simplified 1D grid, these weights recover the structure of classical analytical upwind finite differences. When the CFL bound is violated, however, exact diffusion-moment matching cannot generally be achieved because the Softmax-constrained stencil caps the attainable local variance. In that regime the network behaves as an effective capped local operator, and the residual inconsistency appears as a non-zero truncation floor in the empirical $\delta_{\max}$ diagnostics.

Our utilization of an MLP_θ (acting as a Universal Approximator) is designed to bridge this gap. Rather than relying on rigid, hardcoded analytical stencils, the parameterized network possesses the capacity to dynamically learn and approximate these valid probability weights directly from the physical moments. This supports local Consistency in domains where traditional Kushner grid approximations fail to scale.

In practice, to ensure the Universal Approximator actually converges to these specific probabilities (rather than learning an unphysical, deterministic jump process that spuriously maximizes the reward), we explicitly penalize the network during training. By adding the squared discrepancies of the first and second moments (matching the drift $f\Delta t$ and diffusion $\sigma\sigma^\top \Delta t$) directly into the loss function, we constrain the optimization landscape. This encourages transition probabilities parameterizing the Spatially-Coupled Monotone ResNet to better emulate the underlying SDE dynamics, thereby satisfying the Consistency condition required for the Barles-Souganidis framework.

A.4 Proof of Theorem 5 (Viscosity Convergence)

Proof The proof follows the Barles-Souganidis framework (Barles and Souganidis, 1991). We specifically correct the theoretical fallacy of taking finite-difference limits on non-smooth functions by transferring the differential operator entirely onto a smooth test function via the spatial monotonicity of our architecture.

By Lemma 3, the sequence of numerical solutions V^{θ_k, L_k} generated by our scheme is uniformly bounded in $L^\infty([0,T] \times \mathcal{S})$. Let $L_k \to \infty$ be a sequence with $\Delta t_k := T/L_k \to 0$, $h_k \to 0$, and $\delta_{\max}^{(k)} \to 0$. We define the upper and lower semi-continuous envelopes of the

numerical solutions:

$$\overline{V}(\tau, s) = \limsup_{\substack{k\to\infty \\ (\tau',s')\to(\tau,s)}} V^{\theta_k, L_k}(\tau', s'),$$
$$\underline{V}(\tau, s) = \liminf_{\substack{k\to\infty \\ (\tau',s')\to(\tau,s)}} V^{\theta_k, L_k}(\tau', s').$$

Step 1: Subsolution Property. We show that $\overline{V}$ is a viscosity subsolution of the HJB equation (3). Let $\phi \in C^{1,2}([0,T] \times \mathcal{S})$ be a smooth test function, and let (τ_0, s_0) with $\tau_0 > 0$ be a strict global maximum of $\overline{V} - \phi$. Without loss of generality, we can assume $\overline{V}(\tau_0, s_0) = \phi(\tau_0, s_0)$ (if not, we simply shift ϕ by an additive constant).

By the standard properties of upper semi-continuous envelopes, there exists a sequence (τ_k, s_k) such that $(\tau_k, s_k) \to (\tau_0, s_0)$, $V^{\theta_k, L_k}(\tau_k, s_k) \to \overline{V}(\tau_0, s_0)$, and (τ_k, s_k) is a global maximum of $V^{\theta_k, L_k} - \phi$. Let $\xi_k = V^{\theta_k, L_k}(\tau_k, s_k) - \phi(\tau_k, s_k)$. By construction, $\xi_k \to 0$ as $k \to \infty$. Because (τ_k, s_k) is a global maximum, we establish the following global upper bound for all (τ, s):

$$V^{\theta_k, L_k}(\tau, s) \leq \phi(\tau, s) + \xi_k. \tag{23}$$

Because the numerical scheme propagates forward in time τ, we evaluate the operator at the *previous* time step $\tau_k - \Delta t_k$ to extract the temporal derivative without differentiating V^{θ_k, L_k}. Using the global inequality (23), we have:

$$V^{\theta_k, L_k}(\tau_k - \Delta t_k, \cdot) \leq \phi(\tau_k - \Delta t_k, \cdot) + \xi_k.$$

Applying the neural operator $\mathcal{P}^{\theta_k}_{\Delta t_k}$ to both sides and invoking its monotonicity (Lemma 4) yields:

$$V^{\theta_k, L_k}(\tau_k, s_k) = \mathcal{P}^{\theta_k}_{\Delta t_k}\left[V^{\theta_k, L_k}(\tau_k - \Delta t_k, \cdot)\right](s_k) \leq \mathcal{P}^{\theta_k}_{\Delta t_k}\left[\phi(\tau_k - \Delta t_k, \cdot) + \xi_k\right](s_k). \tag{24}$$

By the architectural definition of our scheme (9), the spatial aggregation is a convex combination ($\sum_{s'} w_{\theta_k} = 1$). Consequently, an additive spatial constant passes through the operator cleanly, scaled only by the discount factor:

$$\mathcal{P}^{\theta_k}_{\Delta t_k}\left[\phi + \xi_k\right](s_k) = \mathcal{P}^{\theta_k}_{\Delta t_k}\left[\phi\right](s_k) + (1 - \gamma \Delta t_k)\xi_k.$$

Substituting this algebraic property into our inequality, and recalling that by definition $V^{\theta_k, L_k}(\tau_k, s_k) = \phi(\tau_k, s_k) + \xi_k$, we obtain:

$$\phi(\tau_k, s_k) + \xi_k \leq \mathcal{P}^{\theta_k}_{\Delta t_k}\left[\phi(\tau_k - \Delta t_k, \cdot)\right](s_k) + \xi_k - \gamma \Delta t_k \xi_k. \tag{25}$$

The problematic error terms ξ_k gracefully cancel from both sides. Rearranging the remaining terms and dividing by the positive step size Δt_k, we form a discrete difference purely in terms of the smooth function ϕ:

$$\frac{\phi(\tau_k, s_k) - \mathcal{P}^{\theta_k}_{\Delta t_k}\left[\phi(\tau_k - \Delta t_k, \cdot)\right](s_k)}{\Delta t_k} \leq -\gamma \xi_k. \tag{26}$$

We may now pass to the limit as $k \to \infty$ (where $\Delta t_k \to 0$, $h_k \to 0$, $(\tau_k, s_k) \to (\tau_0, s_0)$, and $\xi_k \to 0$). Let $\tau' = \tau_k - \Delta t_k$, noting that $\tau' \to \tau_0$. By Assumption 1 (Consistency), the left-hand side converges to the continuous differential operator evaluated on ϕ. As $\xi_k \to 0$, the right-hand side vanishes:

$$\partial_\tau \phi(\tau_0, s_0) + F(\tau_0, s_0, \overline{V}(\tau_0, s_0), \nabla\phi(\tau_0, s_0), \nabla^2\phi(\tau_0, s_0)) \leq 0, \tag{27}$$

Step 2: Supersolution and strong comparison principle. By a symmetric argument—evaluating global minima and mapping local bounded variations via $\underline{V} - \phi$—one obtains that $\underline{V}$ is a viscosity supersolution. The boundary and initial conditions are handled symmetrically, so the standard comparison theorem applies in the same way. Because $\overline{V}$ is a subsolution, $\underline{V}$ is a supersolution, and the Hamiltonian F satisfies the standard properness, degenerate ellipticity, and boundary-compatibility conditions for controlled-diffusion HJB equations, the strong comparison principle applies. Therefore, $\overline{V}(\tau, s) \leq \underline{V}(\tau, s)$ for all $(\tau, s) \in [0, T] \times \mathcal{S}$. Since by definition $\underline{V} \leq \overline{V}$, we conclude that $\overline{V} = \underline{V} = V^*$. The equality of the semi-continuous envelopes then yields locally uniform convergence of V^{θ_k, L_k} to V^* on compact subsets of the domain. ■

Neighborhood Definition $\mathcal{N}_h(s)$: For the 1D LQR state space $s \in [-2, 2]$, we define the local neighborhood using a finite difference stencil: $\mathcal{N}_h(s) = \{s - h, s, s + h\}$ with $h = 0.05$. At the boundaries ($s \approx \pm 2$), the stencil uses Neumann conditions, truncating the neighborhood back into the valid state space.

Network Output and Bellman Loss: The MLP_θ accepts state-action pairs (s, a) and outputs logits $z \in \mathbb{R}^3$. The transition weights are obtained via $w_\theta = \texttt{Softmax}(z)$. Instead of minimizing a continuous PDE residual (which can rely on unstable `autograd` derivatives near kinks), we train the architecture entirely offline using uniformly sampled states. The loss function directly penalizes the violation of the discrete Bellman consistency imposed by the Monotone ResNet structure:

$$\mathcal{L}(\theta) = \mathbb{E}_{s_j}\Bigg[\Big(V_\theta(s_j) - \max_a \Big\{\Delta t\, r(s_j, a) + (1 - \gamma\Delta t) \sum_{k \in \{-1,0,1\}} w_\theta^{(k)}(s_j, a)\, V_{\text{target}}(s_j + kh)\Big\}\Big)^2\Bigg].$$

A.5 Additional Details for Numerical Experiments

This section provides the hyperparameters and implementation specifics required to reproduce the numerical results presented in Section 4. It also explains the transition from standard PINN-style continuous residual minimization to our monotone discrete Bellman updates.

Problem Parameters (Saturated LQR). We utilize a 1-dimensional stochastic optimal control problem with bounded control inputs, which is known to induce non-smooth regions in $V^*(t, s)$ under bang-bang-like optimal policies.

State Dynamics: $ds_t = a_t dt + \sigma_W dW_t$, with diffusion coefficient $\sigma_W = 0.4$.

Domains: State space $\mathcal{S} = [-2, 2]$, Action space $\mathcal{A} = [-1, 1]$.

Time Horizon and CFL Discussion: $T = 1.0$, discretized into L layers. For $h = 0.05$, $\sigma_W = 0.4$, and $|a_{\max}| = 1$, the explicit CFL guideline gives $\Delta t \leq \frac{h^2}{\sigma_W^2 + h|a_{\max}|} = \frac{0.0025}{0.16+0.05} \approx 0.0119$. The reported depth sweeps use $L \in \{2, 3, 4, 5, 6\}$ in 1D and $L = 6$ in the 2D/activation tables, so these runs lie outside that classical explicit CFL regime. In practice, the Softmax-constrained stencil caps the attainable local variance, producing an effective truncated diffusion match rather than exact CFL-respecting moment matching.

Reward Structure: Running reward $r(s, a) = -(s^2 + 0.1a^2)$ and terminal reward $g(s) = -0.5s^2$.

Ground Truth Generation (High-Resolution FDM). To evaluate the network's empirical convergence and its behavior around non-smooth kinks, we computed the reference solution $V^*(t, s)$ using a finite-difference solver.

Scheme: Implicit Crank-Nicolson method for stable time integration, coupled with an upwind scheme for the drift term to prevent numerical dispersion.

Grid Resolution: The spatial domain $[-2.5, 2.5]$ (padded to avoid boundary artifacts) was discretized with a dense grid spacing of $\Delta s_{\text{FDM}} = 10^{-3}$. The temporal step was set to $\Delta t_{\text{FDM}} = 10^{-4}$.

Boundary Conditions: Zero-Neumann boundary conditions ($\nabla_s V = 0$) were enforced at the domain edges, matching the reflected dynamics at the discrete level used by the RL agent.

Spatial Stencil ($\mathcal{N}_h$): For any given state s, we define a 3-point local neighborhood stencil $\mathcal{N}_h(s) = \{s - h, s, s + h\}$ with a fixed spatial resolution $h = 0.05$. If a neighbor falls outside $[-2, 2]$, its value is clamped to the boundary (Neumann condition).

Weight Network (MLP_θ):

Input: A 2-dimensional vector $[s, a]$.

Hidden Layers: Two fully connected layers with 64 units each and ReLU activations (Tanh was used only for the ablation study in Table 3).

Output Layer: A linear layer outputting 3 logits corresponding to the 3 stencil points.

Monotone Activation: A `Softmax` layer transforms the logits into non-negative transition probabilities $w_\theta \in [0, 1]^3$, guaranteeing $\sum w_\theta = 1$.

Value Network (V_ϕ): A separate standard MLP (2 layers, 64 units) used to parameterize the continuous value function surface across the domain.

Training Objective: Enforcing Physical Dynamics without Autograd. Standard deep learning PDE solvers (our Baseline) minimize the continuous residual using PyTorch's 'torch.autograd.grad' to compute $\nabla^2 V_\phi$. As established, this fails at non-smooth kinks and makes ReLU networks difficult to justify within strong second-order PDE residual formulations near non-smooth kinks.

Conversely, our Spatially-Coupled Monotone ResNet avoids `autograd` for spatial derivatives by shifting the differential approximation onto the spatial dispersion of w_θ. However, without physical constraints, a purely value-based Bellman loss could allow the network to learn an unphysical MDP. To address this, the training objective forces the network to satisfy both the discrete Bellman recursion and the physical SDE moments (evaluated

uniformly over state and action spaces to ensure global validity):

$$\mathcal{L}_{Total}(\theta,\phi) = \mathcal{L}_{Bellman}(\theta,\phi) + \lambda_{drift}\mathcal{L}_{drift}(\theta) + \lambda_{diff}\mathcal{L}_{diff}(\theta).$$

To preserve the explicit temporal discretization used in the operator-level analysis 5, the Bellman term acts with respect to a frozen target network $V_{\phi^{(l)}}$ (representing the fixed output of the previous layer l), rather than using moving-average updates:

$$\mathcal{L}_{Bellman} = \mathbb{E}_{s\sim\mathcal{U}(-2,2)}\Bigg[\Big(V_\phi(s) - \max_{a\in\{-1,0,1\}}\Big\{\Delta t\, r(s,a) + (1-\gamma\Delta t)\sum_{k\in\{-1,0,1\}} w_\theta^{(k)}(s,a)\, V_{\phi^{(l)}}(s+kh)\Big\}\Big)^2\Bigg].$$

The physical constraints map the 1D stencil $\{s-h, s, s+h\}$ to the LQR dynamics ($f(s,a)=a$ and $\sigma_W = 0.4$): In the empirical Bellman update, the continuous action interval $\mathcal{A}=[-1,1]$ is discretized to the three-point set $\{-1,0,1\}$ to evaluate the supremum in fitted value iteration.

$$\mathcal{L}_{drift}(\theta) = \mathbb{E}_{s,a}\left[\left(h\cdot(w_\theta^{(1)}(s,a) - w_\theta^{(-1)}(s,a)) - a\Delta t\right)^2\right],$$
$$\mathcal{L}_{diff}(\theta) = \mathbb{E}_{s,a}\left[\left(h^2\cdot(w_\theta^{(1)}(s,a) + w_\theta^{(-1)}(s,a)) - \sigma_W^2\Delta t\right)^2\right].$$

Training Hyperparameters. Optimizer: Adam optimizer with a fixed learning rate of $\eta = 10^{-3}$ for both V_ϕ and MLP_θ.

Batch Size: 1024 uniformly sampled states per gradient step.

Iterations: 30,000 steps per temporal layer.

Statistical Robustness: All experiments (Baseline and Ours) were independently executed across **30 random seeds**. Error margins reported in Section 4 represent the 95% confidence intervals.

Metric Calculation Details. Monotonicity Violations (Table 1): For the Baseline PINN, monotonicity violations occur where the effective discrete operator maps $V(y)$ to $V(x)$ with a negative weight (computed via local gradient probing: $1+\Delta t\partial_V h < 0$). For our Monotone ResNet, this metric is 0.0% in our experiments; architecturally, non-negative `Softmax` weights enforce monotone local aggregation at the operator level.

Max Overshoot Error at Kink (Table 1): Evaluated specifically within a narrow bounding box $\mathcal{B} = [s_{\text{kink}} - 0.1, s_{\text{kink}} + 0.1]$ centered on the non-smooth gradient discontinuity. The error is defined as $\sup_{s\in\mathcal{B}}(V_{\text{pred}}(s) - V^*(s)) \times \mathbb{I}_{\{V_{\text{pred}}(s) > V^*(s)\}}$, measuring the non-physical upward oscillations (Gibbs-like phenomenon) generated by the schemes.