SCALABLE FIXED-POINT FRAMEWORK FOR HIGH-DIMENSIONAL HAMILTONJACOBI EQUATIONS*

YESOM PARK† AND STANLEY OSHER‡

Abstract. We propose a novel, mesh–free, and gradient–free fixed–point approach for computing viscosity solutions of high-dimensional Hamilton–Jacobi (HJ) equations. By leveraging the Hopf–Lax formula, our approach iteratively solves the associated variational problem via a Picard iteration, enabling efficient evaluation of both the solution and its corresponding control without relying on grids, characteristics, or differentiation. We demonstrate the practical efficacy and scalability of the approach through numerical experiments in up to 100 dimensions, including control problems and non-smooth solutions. Our results show that the proposed scheme achieves high accuracy, is highly efficient, and exhibits computational times that are largely independent of dimensionality, highlighting its suitability for high-dimensional problems.

Key words. Hamilton–Jacobi equations, Fixed-point iteration, High-dimensional problems, Hopf–Lax formula

MSC codes. 65N99, 65K10, 65N12, 93B40

1. Introduction. Hamilton–Jacobi partial differential equations (HJ equations) play a central role in numerous areas of mathematics, physics, and engineering, including optimal control [20, 48, 5], mechanics [19, 17], and the study of dynamic systems [29, 47]. They provide a rigorous framework for modeling systems governed by physical laws and have found applications in diverse domains such as geometric optics [38, 36], computer vision [8, 26, 40], robotics [33, 31, 4], trajectory optimization [18, 44], traffic flow modeling [27, 30], and financial strategies [24, 7]. Many of these applications naturally lead to *high-dimensional* HJ equations, where the state space involves multiple degrees of freedom or coupled system variables. The inherent high dimensionality of these problems poses a severe challenge for classical numerical methods.

Traditional grid-based methods, including essentially non-oscillatory (ENO) and weighted ENO (WENO) schemes [42, 28, 6, 45], semi-Lagrangian methods [21, 12, 22], and level set approaches [41, 38, 39, 37, 2], achieve high accuracy in low-dimensional settings but rely heavily on discretizing the state space. As the dimensionality increases, the number of grid points required for accurate approximations grows exponentially, leading to prohibitive computational costs. Consequently, these methods become infeasible for many high-dimensional control problems, where the state space can easily exceed ten dimensions. To address this *curse of dimensionality*, alternative approaches have been proposed, including max-plus algebra-based methods [35, 1, 23] and Hopf–Lax formula-based methods [16, 10, 11, 9], which avoid explicit grid discretization and allow parallel computation by solving optimization problems at each point, though they are generally limited to specific problem classes and often require convexity or other structural assumptions.

Recent advances in neural network-based approaches [49, 51, 46, 34, 32, 50] have leveraged neural networks to store solution information in their parameters, offering scalable representations that can partially mitigate the dimensionality challenge inher-

†Department of Mathematics, University of California, Los Angeles (yeisom@math.ucla.edu).

‡Department of Mathematics, University of California, Los Angeles (sjo@math.ucla.edu).

1

TABLE 1

Comparison of computational paradigms for solving Hamilton–Jacobi equations. The proposed fixed-point method performs direct local evaluation, *deep learning methods represent a* global approximation *requiring prior training, and WENO is an* evolutionary solver *advancing solutions causally in time.*

Method	Computation Type	Precomputation	Evaluation Cost	Scalability	Convergence
Grid-based methods	Evolutionary	Grid generation	Exponential in grid size	Exponential in d	Guaranteed
Deep learning methods	Global approximation	Heavy training	Low (forward network pass)	Quadratic w.r.t. d	No
Fixed-point (Ours)	Local, mesh-free	None	Linear in iteration count	Nearly d-independent	Guaranteed

ent to high-dimensional HJ equations. In particular, the development of specialized neural network architectures tailored to express variational or representation formulas for specific classes of HJ equations has shown promising results [14, 15, 13]. Complementarily, a new method based on characteristics has been proposed to efficiently compute viscosity solutions via an implicit formula [43]. Despite these advancements, convergence and error guarantees of these approaches are often absent or heuristic, which limits the reliability of the computed solutions. Moreover, the repeated computation of derivatives through automatic differentiation for backpropagation incurs substantial computational and memory costs as the problem dimension increases. These factors constrain the practical scalability and efficiency of neural solvers for extremely high-dimensional HJ equations, particularly in real-world applications where both accuracy and computational tractability are critical.

In this work, we introduce a novel *fixed-point iteration framework* for solving high-dimensional HJ equations via the Hopf–Lax formula. Unlike traditional grid-based or neural network-based methods, our approach is *mesh-free*, *causality-free*, and *derivative-free*, making it particularly suitable for large state dimensions. By reformulating the Hopf–Lax formula as a fixed-point problem, we obtain a simple iterative scheme that converges to the unique global minimizer, enabling direct computation of the viscosity solution and associated optimal controls without spatial discretization or repeated differentiation. Importantly, the algorithm is straightforward to implement, requiring only basic iterative updates, which makes it highly accessible for practical use.

We also establish rigorous convergence and error bounds under standard convexity and Lipschitz continuity assumptions, covering a wide range of practical optimal control and high-dimensional Hamilton–Jacobi problems. Within this framework, the method achieves unprecedented scalability while avoiding grids, characteristic tracing, and derivative evaluations. Moreover, in scenarios where overlapping characteristics produce kinks and the fixed point is no longer unique, we propose a randomly sampled *multiple initialization* strategy to explore all fixed points and reliably identify the one corresponding to the viscosity solution. Numerical experiments in up to 100 dimensions, including non-smooth and control problems, demonstrate high accuracy, efficiency, and near dimension-independent computational cost.

Table 1 highlights the distinctions between our fixed-point approach, deep learning methods, and classical grid-based solvers. In contrast to deep learning methods, which require heavy global training, and grid-based solvers, which are evolutionary and causal from $t = 0$, the fixed-point method supports *direct local evaluation* at any point (x, t), with guaranteed convergence and nearly dimension-independent cost. Overall, our framework complements the limitations of existing classical numerical schemes and modern deep learning approaches: it combines *dimension-robust scalability* with *provable accuracy*, providing a theoretically sound and practically efficient

83 methodology for high-dimensional HJ equations.

2. Method.

2.1. Fixed-Point Representation via the Hopf–Lax Formula. Consider
86 the Hamilton–Jacobi (HJ) equation

87 (2.1)
$$\begin{cases} u_t + H(\nabla u) = 0, \\ u(x,0) = g(x), \end{cases}$$

88 where $H : \mathbb{R}^d \to \mathbb{R}$ is a convex (or concave) Hamiltonian and $g : \mathbb{R}^d \to \mathbb{R}$ is a
89 Lipschitz continuous initial condition. It is well-known that, under suitable regularity
90 assumptions, the unique viscosity solution of (2.1) can be expressed via the Hopf–Lax
91 formula:

92 (2.2)
$$u(x,t) = \inf_{y \in \mathbb{R}^n} \left\{ t\, H^* \Big(\frac{x - y}{t} \Big) + g(y) \right\},$$

93 where H^* denotes the Legendre transform of H.
94   While the Hopf–Lax formula provides a variational characterization of the solu-
95 tion, its practical evaluation requires computing the global minimizer y^*. By formally
96 differentiating the variational expression, we arrive at the following fixed-point rela-
97 tion for the minimizer:

98 (2.3)
$$y = x - t\, \nabla H(\nabla g(y)),$$

99 which naturally motivates a Picard fixed-point iteration for approximating y^*. Specif-
100 ically, the iterative procedure

101 (2.4)
$$y^{(k+1)} = x - t\, \nabla H(\nabla g(y^{(k)})), \quad k = 0, 1, 2, \dots$$

102 can be implemented in practice with a prescribed maximum number of iterations and
103 convergence tolerance. The detailed algorithmic implementation is summarized in Al-
104 gorithm 2.1. This fixed-point perspective establishes a bridge between the variational
105 formulation of HJ equations and efficient computational realization, enabling scalable
106 evaluation even in high-dimensional settings such as optimal control.

Algorithm 2.1 Fixed-Point Iteration for Hamilton–Jacobi Equation

---

Require: $x \in \mathbb{R}^d$, $t > 0$, $g : \mathbb{R}^d \to \mathbb{R}$, $H : \mathbb{R}^d \to \mathbb{R}$, $K_{\max} \in \mathbb{N}$ (maximum iterations),
    $\varepsilon > 0$ (tolerance)
Ensure: $y^* \in \mathbb{R}^d$, approximate minimizer of the Hopf–Lax formula
 1: Initialize $y^{(0)} \leftarrow x$
 2: **for** $k = 0$ **to** $K_{\max} - 1$ **do**
 3:    $y^{(k+1)} \leftarrow x - t\, \nabla H(\nabla g(y^{(k)}))$
 4:    **if** $\|y^{(k+1)} - y^{(k)}\|_2 < \varepsilon$ **then**
 5:       **break**
 6:    **end if**
 7: **end for**
 8: $y^* \leftarrow y^{(k+1)}$
 9: **return** y^*

Control-theoretic interpretation.. The fixed-point relation (2.3) admits a natural interpretation from the viewpoint of optimal control. Recall that the Hamilton–Jacobi equation can be equivalently expressed as

$$u_t + H(\nabla u) = u_t + \sup_q \{\nabla u^\top q - L(q)\} = 0,$$

where L is the Legendre transform of H. The corresponding value function of the optimal control problem is given by

$$(2.5) \qquad u(x,t) = \inf_{q(\cdot)} \left\{ \int_0^t L(q(s)) \, \mathrm{d}s + g(y(0)) : y(t) = x, \ \dot{y}(s) = q(s) \right\}.$$

Under standard convexity and differentiability assumptions, the minimizer y^* obtained from the fixed-point iteration corresponds to the optimal initial state $y(0)$ in (2.5), and the associated optimal control is given by

$$(2.6) \qquad\qquad\qquad\qquad q^* = \frac{x - y^*}{t}.$$

Hence, the proposed fixed-point scheme not only enables efficient computation of the viscosity solution $u(x,t)$, but also provides direct access to the corresponding optimal control, without requiring characteristic tracing or grid-based discretization.

Advantages.. Compared to traditional mesh-based numerical solvers (e.g., WENO schemes) or recent deep learning-based approaches, the proposed fixed-point method achieves significantly faster evaluation while maintaining high accuracy. Its key advantages include:

- **Mesh-Free and Scalable:** No spatial discretization or characteristic tracing is required, making the method well-suited for high-dimensional problems.
- **Derivative-Free:** The approach avoids explicit derivative computations, resulting in low computational overhead and excellent scalability.
- **Casuality-free and Highly Parallelizable:** The fixed-point updates do not rely on causality from neighboring grid points, resulting in a causality-free scheme that enables embarrassingly parallel computation across all spatial locations and significantly reduces wall-clock time on modern hardware.
- **Easy Implementation:** The algorithm consists of simple iterative updates, making it straightforward to implement and highly accessible for practical use.
- **Highly Fast in High Dimensions:** The pointwise fixed-point evaluation enables substantial computational speedups and maintaining efficiency even in very high-dimensional settings.
- **Control Recovery:** The optimal control associated with the Hamilton–Jacobi solution is directly obtained from the fixed-point minimizer (2.6).
- **Wide applicability:** The scheme accommodates a broad class of convex Hamiltonians and convex, Lipschitz-differentiable initial data, and can be extended to certain nonsmooth settings.

2.2. Convergence Analysis. In this section, we establish rigorous convergence, error, and complexity guarantees for the proposed fixed-point iteration scheme (2.4). We derive quantitative bounds on both the minimizer error and the approximation error in the viscosity solution, and further analyze the iteration complexity required to achieve a prescribed accuracy. Together, these results provide a comprehensive theoretical foundation for the efficiency, stability, and accuracy of the proposed method.

150　　　We begin by presenting sufficient conditions that ensure the existence and unique-
151　ness of the fixed point, as well as the global convergence of the iteration. Throughout
152　this section, $\|\cdot\|$ denotes the L^2 norm.

153　　　THEOREM 2.1 (Fixed-point convergence). *Let* $H : \mathbb{R}^n \to \mathbb{R}$ *and* $g : \mathbb{R}^n \to \mathbb{R}$ *be*
154　*continuously differentiable functions such that:*

155　　　　• H *is* strictly convex *and has* L_H-*Lipschitz continuous gradient:*

156
$$\|\nabla H(p_1) - \nabla H(p_2)\| \le L_H \|p_1 - p_2\|, \quad \forall p_1, p_2 \in \mathbb{R}^n.$$

157　　　　• g *is* convex *and has* L_g-*Lipschitz continuous gradient:*

158
$$\|\nabla g(y_1) - \nabla g(y_2)\| \le L_g \|y_1 - y_2\|, \quad \forall y_1, y_2 \in \mathbb{R}^n.$$

159　*Define the fixed-point operator*

160
$$F(y) := x - t\,\nabla H(\nabla g(y)).$$

161　*Then, for any* $x \in \mathbb{R}^n$ *and any time step* $t > 0$ *satisfying*

162
$$t\,L_H\,L_g < 1,$$

163　*the iteration*
164
$$y^{(k+1)} = F(y^{(k)}), \qquad k = 0, 1, 2, \ldots$$

165　*converges to the unique global minimizer of the Hopf–Lax formula* (2.2).

166　　　*Proof.* We aim to show that $F(y) = x - t\nabla H(\nabla g(y))$ is a contraction mapping.
167　　　Let $y_1, y_2 \in \mathbb{R}^n$. Then:

168
$$\|F(y_1) - F(y_2)\| = \|x - t\nabla H(\nabla g(y_1)) - (x - t\nabla H(\nabla g(y_2)))\|$$
169
$$= t\,\|\nabla H(\nabla g(y_1)) - \nabla H(\nabla g(y_2))\|$$
170
$$\le t\,L_H\,\|\nabla g(y_1) - \nabla g(y_2)\|$$
171
$$\le t\,L_H\,L_g\,\|y_1 - y_2\|.$$

172　The first inequality follows from the L_H-Lipschitz continuity of ∇H, and the second
173　from the L_g-Lipschitz continuity of ∇g.
174　　　Define $\lambda := t \cdot L_H \cdot L_g$. Then:

175
$$\|F(y_1) - F(y_2)\| \le \lambda \|y_1 - y_2\|.$$

176　　　Since $\lambda < 1$ by assumption, F is a contraction on the complete metric space $\mathbb{R}^n$.
177　By the Banach fixed-point theorem, there exists a unique fixed point $y^* \in \mathbb{R}^n$ such
178　that
179
$$y^* = F(y^*) = x - t\nabla H(\nabla g(y^*)).$$

180　　　We now show that this fixed point y^* is the unique minimizer of the Hopf–Lax
181　function
182
$$\phi(y) := tH^*\left(\frac{x-y}{t}\right) + g(y).$$

183　　　Since H^* is the convex conjugate of a strictly convex and differentiable function
184　H, the Hopf-Lax function is differentiable. Therefore, the critical point condition for
185　ϕ is:

186
$$0 \in \nabla\phi(y) = -\nabla H^*\left(\frac{x-y}{t}\right) + \nabla g(y).$$

187   Equivalently,

$$\nabla g(y) = \nabla H^* \left(\frac{x - y}{t} \right) \quad \Longleftrightarrow \quad y = x - t\nabla H(\nabla g(y)),$$

189   where we used the identity $\nabla H^* = (\nabla H)^{-1}$ under strict convexity. Thus, any solution
190   to the fixed-point equation is a critical point of ϕ, and vice versa.

191   Since H^* is strictly convex (as H is strictly convex), and g is convex, their sum
192   $\phi(y)$ is strictly convex. Hence, ϕ has a unique critical point, which must be the unique
193   global minimizer of ϕ.

194   Therefore, the fixed point y^* is the unique minimizer of the Hopf–Lax functional:

$$u(x,t) = \min_y \left\{ tH^* \left(\frac{x - y}{t} \right) + g(y) \right\} = \phi(y^*). \qquad \qquad \square$$

196   By establishing the convergence of the fixed-point iteration, Theorem 2.1 provides
197   a theoretical foundation for a *gradient-free, mesh-free, and causality-free* approach to
198   evaluating viscosity solutions of high-dimensional Hamilton–Jacobi equations.

199   We now proceed to quantify the relationship between the residual error and the
200   resulting bounds on both the minimizer and the Hopf–Lax functional value, thereby
201   providing explicit estimates for the approximation accuracy of the method.

202   THEOREM 2.2 (Error Analysis).   *Let H and g satisfy the assumptions of The-*
203   *orem 2.1, and let $t > 0$ satisfy $L_F := tL_H L_g < 1$. Consider the iteration $y^{(k+1)} =$*
204   *$F(y^{(k)})$ with $F(y) = x - t\,\nabla H(\nabla g(y))$, and denote the minimizer by y^*. Define the*
205   *residual $r^{(k)} := y^{(k+1)} - y^{(k)}$. Then:*
206   1. ***(Hopf–Lax Minimizer error)***

$$\|y^{(k)} - y^*\| \le \frac{1}{1 - L_F} \|r^{(k)}\|.$$

208   2. ***(Solution error)*** *If H^* is L_{H^*}-Lipschitz on the relevant domain, then*

$$|u(x,t) - u^{(k)}(x,t)| \le \frac{L_{H^*}/t + L_g}{1 - L_F} \|r^{(k)}\|,$$

210   *where $u^{(k)}(x,t) := t\,H^*\big(\frac{x-y^{(k)}}{t}\big) + g(y^{(k)})$.*
211   3. ***(Solution gradient error)***

$$\|\nabla u(x,t) - \nabla u^{(k)}(x,t)\| \le L_g \|r^{(k)}\|.$$

213   *Proof.* **1. Hopf–Lax minimizer error bound.** By Theorem 2.1, the minimizer
214   of the Hopf–Lax formula $y^\star$ is the unique fixed point of F. Since F is a contraction
215   with constant L_F, we have

$$\|F(y) - F(y^*)\| \le L_F \|y - y^*\| \quad \forall y \in \mathbb{R}^n.$$

217   By definition of the residual,

$$r^{(k)} = y^{(k+1)} - y^{(k)} = F(y^{(k)}) - y^{(k)}.$$

219   Then,

$$\begin{aligned}
\|y^{(k)} - y^*\| &= \|y^{(k)} - F(y^*)\| \\
&= \|y^{(k)} - F(y^{(k)}) + F(y^{(k)}) - F(y^*)\| \\
&\le \|r^{(k)}\| + \|F(y^{(k)}) - F(y^*)\| \\
&\le \|r^{(k)}\| + L_F \|y^{(k)} - y^*\|.
\end{aligned}$$

Rearranging gives

$$\|y^{(k)} - y^*\| \leq \frac{1}{1 - L_F}\|r^{(k)}\|.$$

2. Solution error bound. Let

$$u(x,t) = tH^*\Big(\frac{x - y^*}{t}\Big) + g(y^*), \quad u^{(k)}(x,t) = tH^*\Big(\frac{x - y^{(k)}}{t}\Big) + g(y^{(k)}).$$

Using the Lipschitz continuity of H^* and g,

$$|u(x,t) - u^{(k)}(x,t)| \leq t\left|H^*\Big(\frac{x - y^*}{t}\Big) - H^*\Big(\frac{x - y^{(k)}}{t}\Big)\right| + |g(y^*) - g(y^{(k)})|$$

$$\leq t\, L_{H^*}\frac{\|y^{(k)} - y^*\|}{t} + L_g\|y^{(k)} - y^*\|$$

$$= (L_{H^*}/t + L_g)\|y^{(k)} - y^*\|$$

$$\leq \frac{L_{H^*}/t + L_g}{1 - L_F}\|r^{(k)}\|.$$

3. Solution gradient error bound. Taking derivative of Hopf–Lax formula with respect to x gives

$$\nabla u(x,t) = \nabla H^*((x - y^*)/t).$$

By incorporating the implicit relation (2.3), it gives

$$\nabla u(x,t) = \nabla H^*(\nabla H(\nabla g(y^*))) = \nabla g(y^*),$$

where the last equality comes from the fact $\nabla H^* = \nabla H^{-1}$. Consequently, we have

$$\|\nabla u - \nabla u^{(k)}\| = \left\|\nabla g(y^{(k)}) - \nabla g(y^*)\right\|$$

$$\leq L_g\|r^{(k)}\|.$$

This completes the proof. □

Theorem 2.2 provides explicit quantitative bounds relating the residual norm to both the minimization error and the Hopf–Lax approximation error. In particular, controlling the error both in the Hopf–Lax minimizer y^* and the solution u is equivalent to controlling the error not only in the viscosity solution itself but also in its gradient. From the perspective of control theory, this ensures that the error is controlled not only for the value function but also for the associated optimal control, thereby providing rigorous guarantees for both solution and control accuracy. This equivalence highlights a key strength of the proposed methodology. In practice, this result provides a rigorous and efficient termination criterion: once the residual $\|r^{(k)}\|$ falls below a tolerance ε, the corresponding error in both the solution and its gradient (or, equivalently, in the value function and the associated optimal control) is guaranteed to remain within a controlled bound.

We now characterize the rate of convergence and the number of iterations required to reach a prescribed error threshold.

THEOREM 2.3 (Iteration Complexity). *Under the same assumption and notation in Theorem 2.2, for any initial guess $y^{(0)} \in \mathbb{R}^d$, the number of iterations required to achieve a residual norm $\|r^{(k)}\| \leq \varepsilon$ satisfies*

$$k \geq \frac{\log\Big(\frac{(1+L_F)\|y^{(0)} - y^*\|}{\varepsilon}\Big)}{-\log L_F}.$$

In particular, the fixed-point iteration converges linearly with rate L_F, and thus the iteration complexity scales as

$$k = O\big(\log(1/\varepsilon)\big).$$

Proof. Since F is a contraction with modulus L_F, it holds that

$$\|y^{(k+1)} - y^*\| \leq L_F \|y^{(k)} - y^*\|.$$

By induction,

$$\|y^{(k)} - y^*\| \leq L_F^k \|y^{(0)} - y^*\|.$$

The residual satisfies

$$r^{(k)} = y^{(k+1)} - y^{(k)} = (y^{(k+1)} - y^*) - (y^{(k)} - y^*),$$

and hence

$$\|r^{(k)}\| \leq \|y^{(k+1)} - y^*\| + \|y^{(k)} - y^*\| \leq (1 + L_F)L_F^k \|y^{(0)} - y^*\|.$$

To achieve $\|r^{(k)}\| \leq \varepsilon$, it suffices that

$$(1 + L_F)L_F^k \|y^{(0)} - y^*\| \leq \varepsilon.$$

Solving for k yields

$$k \geq \frac{\log\left(\frac{(1+L_F)\|y^{(0)} - y^*\|}{\varepsilon}\right)}{-\log L_F}.$$

Since $-\log L_F > 0$ and independent of ε, we have $k = O(\log(1/\varepsilon))$, completing the proof. $\qquad\square$

Combining the results of Theorems 2.2 and 2.3, the fixed-point iteration is guaranteed to converge linearly, with the convergence rate determined by the contraction modulus L_F. While the formal upper bound on the number of iterations is independent of the problem dimension d, in practice L_F may increase with d, leading to a moderate slowdown in convergence. Nevertheless, this dimension-dependent effect is significantly less severe than the exponential scaling of computational cost observed in classical mesh-based schemes or recent deep learning-based approaches, rendering the fixed-point method highly scalable for high-dimensional HJ problems.

2.3. Fixed Point Solver with Multiple Initializations. In general, the critical point of the Hopf–Lax formula is not necessarily unique. Even when both the Hamiltonian H and the initial condition g are smooth, the viscosity solution of the Hamilton–Jacobi equation may exhibit non-smooth behavior due to the intersection of characteristics over time, resulting in the formation of kinks. In such cases, multiple stationary points of the Hopf–Lax energy functional may exist, and equivalently, the fixed-point equation (2.3) can admit multiple solutions.

To robustly handle this situation, we adopt a multi-initialization strategy for the fixed-point iteration. The procedure is summarized as follows:

1. **Multiple Random Initializations:** Generate N independent random initial guesses $\{y_0^{(i)}\}_{i=1}^N$, each sampled uniformly from the hypercube $[-\alpha, \alpha]^d$ for a prescribed $\alpha > 0$.

2. **Fixed-Point Iteration:** For each initialization $y_0^{(i)}$, perform the iterative updates

$$y_{k+1}^{(i)} = G(y_k^{(i)}, t, x),$$

until convergence, i.e.,

$$\|y_{k+1}^{(i)} - y_k^{(i)}\|_2 < \varepsilon,$$

or until the maximum iteration count is reached.

3. **Energy Evaluation:** For each converged solution $y^{(i)}$, compute the Hopf–Lax energy

$$\mathcal{E}(y^{(i)}; x, t) = t\, H^*\left(\frac{x - y^{(i)}}{t}\right) + g\left(y^{(i)}\right),$$

where H^* denotes the Legendre transform of H.

4. **Selection of Optimal Fixed Point:** Among the converged fixed points, select

$$y^* = \arg\min_i \mathcal{E}(y^{(i)}; x, t),$$

which corresponds to the minimizer in the Hopf–Lax formula (2.2).

This multi-initialization approach enables the algorithm to explore multiple potential fixed points and ensures that the selected solution corresponds to the true viscosity solution, even in the presence of non-smooth features such as kinks.

THEOREM 2.4 (Probabilistic Completeness of Multi-Initialization). *Let $G : \mathbb{R}^d \to$ ▮ $\mathbb{R}^d$ be a continuous map, and suppose that it admits a finite set of fixed points $\{y_1^*, \ldots, y_M^*\}$ satisfying*

$$y_i^* = G(y_i^*), \quad i = 1, \ldots, M.$$

Assume the following:

- *(**Local contraction**) For each i, there exists an open neighborhood $\mathcal{B}_i \subset \mathbb{R}^d$ of y_i^* such that*

$$\|G(y) - G(z)\| \leq \kappa_i \|y - z\|, \qquad \forall y, z \in \mathcal{B}_i,$$

with some $\kappa_i \in [0, 1)$. Hence, each $\mathcal{B}_i$ is a basin of attraction under the iteration $y_{k+1} = G(y_k)$.

- *(**Nonzero measure**) Each $\mathcal{B}_i$ has positive Lebesgue measure, i.e. $\mu(\mathcal{B}_i) > 0$.*
- *(**Independent sampling**) Initial guesses $y_0^{(1)}, \ldots, y_0^{(N)}$ are drawn independently from a probability distribution ρ that is absolutely continuous with respect to the Lebesgue measure, supported on a compact set $\Omega \subset \mathbb{R}^d$ satisfying $\mu(\Omega) > 0$.*

Then, with probability one, as $N \to \infty$, the fixed-point iteration

$$y_{k+1}^{(j)} = G(y_k^{(j)}), \qquad j = 1, \ldots, N,$$

converges to all fixed points $\{y_i^*\}_{i=1}^M$.

Proof. For each i, denote the probability that a single random initialization falls within the basin $\mathcal{B}_i$ as

$$p_i = \int_{\mathcal{B}_i} \rho(y)\, dy.$$

By assumption, ρ is absolutely continuous and $\mu(\mathcal{B}_i) > 0$, hence $p_i > 0$.

329   The probability that none of the N independent initializations falls within $\mathcal{B}_i$ is
330 $(1 - p_i)^N$. Therefore, the probability that at least one initialization lies in $\mathcal{B}_i$ is

331
$$1 - (1 - p_i)^N.$$

332 Since there are finitely many basins ($M < \infty$), the probability that all basins are
333 covered is

334
$$\mathbb{P}_N = \prod_{i=1}^{M} \left(1 - (1 - p_i)^N\right).$$

335 As $N \to \infty$, each term satisfies $\lim_{N \to \infty}(1 - (1 - p_i)^N) = 1$, hence by continuity of
336 finite products,

337
$$\lim_{N \to \infty} \mathbb{P}_N = 1.$$

338 This proves that with probability approaching one, every basin $\mathcal{B}_i$ is sampled at least
339 once.
340   Within each $\mathcal{B}_i$, the local contraction property ensures that for any initialization
341 $y_0^{(j)} \in \mathcal{B}_i$, the iteration $y_{k+1}^{(j)} = G(y_k^{(j)})$ converges linearly to the corresponding fixed
342 point y_i^* by the Banach fixed point Theorem.
343   Finally, since each fixed point y_i^* is stationary for the Hopf–Lax energy $\mathcal{E}$, the
344 minimizer of $\mathcal{E}$ among $\{y_i^*\}_{i=1}^{M}$ corresponds to the global minimizer of the Hopf–Lax
345 variational formula. Because all $\{y_i^*\}$ are found almost surely as $N \to \infty$, the selected
346 minimizer y^* equals the optimal solution almost surely.   □

347   Theoretically, infinitely many initial points are required to guarantee convergence
348 to all fixed points. In practice, however, a finite number of initial guesses suffices, as
349 demonstrated in Section 3.

350   **3. Experimental Results.** In this section, we evaluate the accuracy and ef-
351 ficiency of the proposed fixed-point method through various numerical experiments.
352 We demonstrate its scalability to high-dimensional problems, compare its performance
353 with existing solvers in terms of accuracy and computational efficiency, and verify its
354 ability to recover optimal controls as well as approximate nonsmooth viscosity solu-
355 tions.
356   All experiments were conducted on a Linux server equipped with an Intel Core
357 i7-6850K CPU (6 cores, 12 threads) at 3.6 GHz, 62 GB RAM, and an NVIDIA TITAN
358 V GPU (12 GB). To ensure a fair comparison with existing deep learning approaches,
359 both the WENO solver and the proposed fixed-point method were implemented in
360 Python 3.

361   **3.1. High-dimensional problems.**

362   **3.1.1. Burgers' equation.** We consider the Hamilton–Jacobi setup with

363 (3.1)
$$H(p) = \frac{1}{2}\|p\|_2^2, \qquad g(x) = \frac{1}{2}\|x\|_2^2,$$

364 which possesses a unique critical point at the origin. The exact solution is given by
365 $u(x, t) = \frac{\|x\|_2^2}{2(1+t)}$. Using the fixed-point iteration

366
$$y^{(k+1)} = x - t\nabla H(\nabla g(y^{(k)})).$$

367   For comparison, we also consider the fifth-order WENO scheme as a represen-
368 tative grid-based method and ImplicitHJ [43], a deep-learning-based approach for

Comparison of numerical and gradient errors, computation time, and memory usage for different dimensions using WENO, ImplicitHJ, and Ours methods.

Dim	Method	L^2 Error	L^∞ Error	L^2 Grad Error	L^∞ Grad Error	Time (s)	Mem (MB)
1D	WENO	1.84×10^{-10}	1.52×10^{-4}	3.06×10^{-5}	6.56×10^{-3}	0.34	0.05
	ImplicitHJ	4.88×10^{-10}	6.91×10^{-5}	1.70×10^{-6}	4.08×10^{-3}	156081.30	0.27
	Ours	2.43×10^{-17}	1.49×10^{-8}	6.37×10^{-15}	2.98×10^{-7}	0.034	0.001
2D	WENO	2.30×10^{-6}	8.62×10^{-3}	3.72×10^{-5}	1.58×10^{-1}	8.45	4.42
	ImplicitHJ	1.57×10^{-8}	3.40×10^{-4}	1.22×10^{-5}	8.95×10^{-3}	175223.30	0.34
	Ours	2.82×10^{-16}	5.96×10^{-8}	1.03×10^{-14}	2.98×10^{-7}	0.04	0.001
3D	WENO	7.22×10^{-3}	2.54×10^{-2}	5.46×10^{-3}	8.91×10^{-1}	1768.01	642.58
	ImplicitHJ	1.52×10^{-7}	1.06×10^{-3}	7.70×10^{-5}	2.46×10^{-2}	177784.40	0.42
	Ours	5.98×10^{-16}	1.19×10^{-7}	1.81×10^{-14}	2.98×10^{-7}	0.04	0.001
10D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	5.05×10^{-6}	6.62×10^{-3}	1.45×10^{-3}	3.83×10^{-2}	195997.48	0.95
	Ours	7.90×10^{-15}	2.38×10^{-7}	6.10×10^{-14}	3.58×10^{-7}	0.04	0.001
50D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	4.71×10^{-2}	3.67×10^{-1}	2.58×10^{-1}	2.53×10^{-1}	258987.28	4.01
	Ours	2.66×10^{-13}	1.91×10^{-6}	2.97×10^{-13}	3.58×10^{-7}	0.05	0.001
100D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	1.37×10^{-1}	4.91×10^{-1}	4.20×10^{-1}	5.46×10^{-1}	324249.12	7.82
	Ours	1.23×10^{-12}	3.81×10^{-6}	6.06×10^{-13}	3.58×10^{-7}	0.05	0.001

approximating viscosity solutions of HJ equations. Performance is evaluated in 1, 2, 3, 10, 50, and 100 dimensions. Table 2 summarizes the L^2 and L^∞ errors of the solution and its gradient, along with computational time and memory usage. All errors are computed over 128 points.

WENO computations use a uniform grid of 128 points with a CFL number of 0.2. ImplicitHJ is trained following the original configuration in the reference [43], and its computational time corresponds to the total training time. The fixed-point method is applied with a maximum of 1000 iterations and a convergence tolerance of 10^{-6} across all dimensions. For fair comparison, all methods are implemented in Python.

It should be noted that these three approaches are fundamentally different in their design principles and computational characteristics. Establishing a fully "fair" comparison is challenging, particularly when considering extremely high-dimensional problems. In setting up the experiments, careful attention was given to ensure that each method is evaluated in a manner representative of its intended use, while maintaining consistency in reported metrics across dimensions.

The results in Table 2 demonstrate the superior performance of the proposed fixed-point method across all dimensions. Notably, our method consistently achieves solution and gradient errors near machine precision, whereas WENO and ImplicitHJ exhibit increasing errors as the dimension grows. In 3D and higher, WENO becomes computationally infeasible, highlighting the curse of dimensionality inherent to grid-based approaches. ImplicitHJ, while scalable, requires extensive training and exhibits non-negligible errors in high dimensions.

In contrast, the fixed-point method is *mesh-free*, *causality-free*, *gradient-free*, and *training-free*, making it inherently scalable to very high-dimensional problems. It computes both the solution and its gradient directly, with theoretically guaranteed accuracy, thereby eliminating approximation errors associated with discretization or neural-network training. Furthermore, computational time and memory usage remain nearly constant across dimensions, as the method does not rely on dense grids or

iterative solvers that scale poorly with dimension, in agreement with the bounds established in Theorem 2.2.

These results highlight the practical advantages of the fixed-point approach: it provides theoretically grounded, high-precision approximations for both the solution and its gradient, while circumventing the key limitations of conventional grid-based and deep-learning-based methods. The combination of *dimension-independent efficiency* and *provable accuracy* positions our method as a robust and scalable alternative for solving high-dimensional Hamilton–Jacobi equations.

Remarkably, even in the 100-dimensional case, the L^2-error of the computed solution was as low as 2.06×10^{-14}, demonstrating both the high accuracy and the scalability of the proposed fixed-point scheme in high-dimensional settings.

3.1.2. LQR Control Problem. We consider an Linear quadratic regulator (LQR)-type optimal control problem in which the cost functional is quadratic in both the control and the initial state:

$$(3.2) \quad u(x,t) = \min_{q(\cdot)} \left\{ \frac{1}{2} \int_0^t q(s)^\top R q(s) \, ds + \frac{1}{2} y(0)^\top Q y(0) \ \Big| \ \dot{y}(s) = q(s), \ y(t) = x \right\},$$

where R and Q are symmetric positive definite matrices. The corresponding Hamiltonian is of the anisotropic quadratic form:

$$H(p) = \frac{1}{2} p^\top R^{-1} p.$$

The problem admits a closed-form analytic solution:

$$(3.3) \qquad y^* = \left(I + tQR^{-1} \right)^{-1} x, \qquad u(x,t) = \frac{1}{2} x^\top \left(I + tQR^{-1} \right)^{-1} Qx.$$

The proposed fixed-point iteration

$$y^{(k+1)} = x - t \, y^{(k)} (R^{-1}Q)^\top$$

is guaranteed to converge if

$$t < \frac{1}{\|R^{-1}Q\|_2},$$

where $\|\cdot\|_2$ denotes the spectral norm.

For a given dimension d, we generated $Q, R \in \mathbb{R}^{d \times d}$ as symmetric positive definite matrices using random Gaussian matrices. Specifically, we drew $A, B \in \mathbb{R}^{d \times d}$ with independent standard normal entries and constructed

$$Q = s_1(A^\top A + I_d), \qquad R = s_2(B^\top B + I_d),$$

with positive scaling factors $s_1, s_2 > 0$, ensuring $Q, R \succ 0$.

We evaluated the fixed-point iteration at 100 randomly sampled points (x,t) satisfying the contraction condition $t < 1/\|R^{-1}Q\|_2$, and recorded the L^2 and L^∞ errors as well as the computational time. Table 3 summarizes the results for two independent random realizations of Q and R. Both sets of experiments demonstrate that the fixed-point iteration achieves high accuracy and low computational cost across dimensions.

TABLE 3
Fixed-point results for the LQR control problem with two independent random realizations of Q and R. Errors are measured against the analytic solution.

Dim	First Random Set			Second Random Set		
	L^2 error	L^∞ error	Time (s)	L^2 error	L^∞ error	Time (s)
1	6.70×10^{-17}	1.19×10^{-7}	0.0041	4.31×10^{-16}	5.96×10^{-8}	0.0045
10	3.26×10^{-12}	5.72×10^{-6}	0.0050	4.82×10^{-12}	7.63×10^{-6}	0.0052
50	2.21×10^{-9}	1.22×10^{-4}	0.0069	1.87×10^{-9}	1.22×10^{-4}	0.0080
100	3.84×10^{-8}	4.88×10^{-4}	0.0171	2.25×10^{-8}	3.66×10^{-4}	0.0104

TABLE 4
Fixed-point results for the cubical problem.

Dim	L^2 Error	L^∞ Error	Time (s)
1	4.53×10^{-9}	8.81×10^{-8}	0.0409
10	2.92×10^{-8}	1.25×10^{-7}	0.0458
50	1.73×10^{-7}	4.80×10^{-7}	0.0476
100	4.14×10^{-7}	1.15×10^{-6}	0.0561

3.1.3. Cubic Hamiltonian Problem. We consider the HJ equation with non-linear Hamiltonian and initial condition

$$H(p) = \frac{1}{3}\|p\|^3, \qquad g(x) = \|x\|^3,$$

for which the corresponding Legendre transform is

$$H^*(q) = \frac{2}{3}\|q\|^{3/2}.$$

We evaluated the minimization for different dimensions d and computed the L^2 and maximum errors relative to the analytic solution. The results are summarized in Table 4.

The results demonstrate that the Hopf–Lax minimization achieves high accuracy across dimensions, with errors and computational time increasing only mildly as the dimension grows.

3.2. Non-smooth solutions.

3.2.1. Absolute-value Quadratic Form. We consider Burgers' equation with a non-smooth initial condition

$$g(x) = x|x|,$$

which can be equivalently formulated as the following optimal control problem:

$$(3.4) \quad u(x,t) = \inf_{q(\cdot)} \left\{ \frac{1}{2}\int_0^t q(s)^2 \, ds + g(y(0)) \,\middle|\, y(t) = x, \ \dot{y}(s) = q(s), \ 0 \le s \le t \right\}.$$

The solution predicted using the fixed-point iteration is shown in Figure 1. As time progresses, one can observe the formation of kinks in the solution, supporting that the proposed method is capable of accurately capturing such sharp features arising from non-smooth initial data.
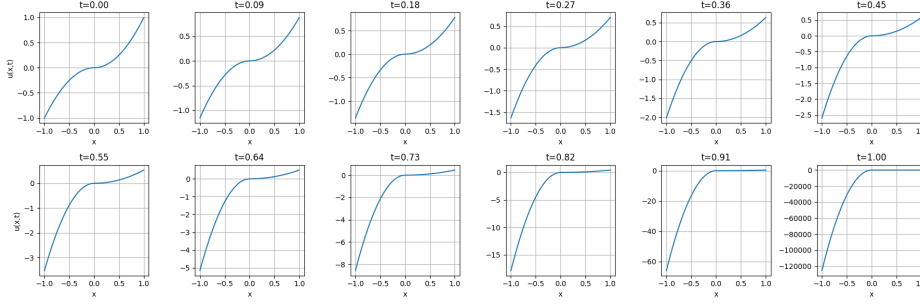
FIG. 1. *Solution of Burgers' equation with non-smooth initial condition $g(x) = x|x|$, computed via fixed-point iteration. Sharp gradients (kinks) develop as time evolves.*
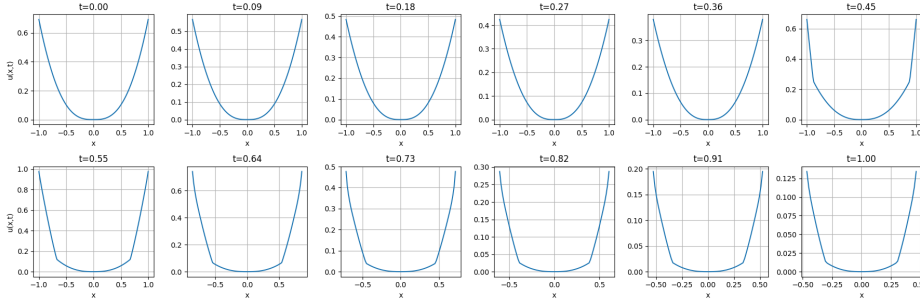

FIG. 2. *Solution of Burgers' equation with non-smooth initial condition $g(x) = x^2 \log(1 + |x|)$, computed via fixed-point iteration. Sharp gradients (kinks) develop as time evolves.*

3.2.2. Logarithmic Form. We consider Burgers' equation with a non-smooth initial condition

$$g(x) = x^2 \log(2 + |x|).$$

The solution predicted using the fixed-point iteration is shown in Figure 2. As time progresses, one can observe the formation of kinks in the solution, supporting that the proposed method is capable of accurately capturing such sharp features arising from non-smooth initial data.

3.3. Multiple Initialization. This section investigates the effectiveness of multiple initialization scheme in handling kinks arising from intersecting characteristics.

3.3.1. Steady Kink. Consider the quadratic Hamiltonian $H(p) = \frac{1}{2} \|p\|_2^2$ and initial function $g(\mathbf{x}) = -\|x\|_1$. The exact solution is

$$u(x,t) = -\|x\|_1 - \frac{dt}{2},$$

where a steady kink occurs at $x = 0$. This arises because characteristics overlap in the region $-t < x < t$, resulting in non-uniqueness of fixed points within this region. In this experiment, we examine whether the multiple initialization strategy proposed in Section 2.3 is capable of effectively identifying the fixed point that corresponds to the viscosity solution among these multiple possibilities.

The results of our methodology with multiple initialization, compared with WENO and ImplicitHJ, are summarized in Table 5. In the experiments, the number of initial-

TABLE 5
Comparison of numerical and gradient errors, computation time, and memory usage for different dimensions using WENO, ImplicitHJ, and Ours methods on the steady kink example.

Dim	Method	L^2 Error	L^∞ Error	L^2 Grad Error	L^∞ Grad Error	Time (s)	Mem (MB)
1D	WENO	1.51×10^{-7}	1.59×10^{-3}	2.17×10^{-4}	2.44×10^{-3}	0.35	0.07
	ImplicitHJ	8.59×10^{-6}	1.72×10^{-5}	2.01×10^{-5}	3.15×10^{-5}	210241.37	52.91
	Ours	1.54×10^{-14}	3.43×10^{-6}	2.15×10^{-13}	1.08×10^{-6}	25.97	3.86
2D	WENO	3.64×10^{-5}	1.02×10^{-2}	7.77×10^{-3}	7.45×10^{-1}	4.50	4.42
	ImplicitHJ	1.10×10^{-4}	2.05×10^{-4}	3.15×10^{-4}	5.10×10^{-4}	168770.98	52.99
	Ours	2.82×10^{-13}	1.13×10^{-5}	4.69×10^{-12}	5.47×10^{-5}	1.2	0.38
3D	WENO	5.38×10^{-5}	1.20×10^{-2}	9.21×10^{-2}	7.65×10^{-1}	1728.87	642.54
	ImplicitHJ	1.15×10^{-4}	2.05×10^{-4}	3.20×10^{-4}	5.15×10^{-4}	177594.47	53.06
	Ours	6.36×10^{-11}	2.41×10^{-4}	1.56×10^{-9}	4.60×10^{-4}	0.32	0.11
10D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	1.23×10^{-3}	2.45×10^{-3}	3.56×10^{-3}	6.32×10^{-3}	195088.43	53.60
	Ours	2.82×10^{-8}	1.13×10^{-3}	4.69×10^{-7}	8.41×10^{-3}	1.20	0.38
50D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	5.12×10^{-1}	1.02×10^{0}	1.08×10^{0}	1.05×10^{0}	242050.79	56.70
	Ours	1.72×10^{-5}	2.90×10^{-2}	6.25×10^{-3}	1.00×10^{0}	18.25	3.86
100D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	3.32×10^{-1}	6.65×10^{-1}	1.13×10^{0}	1.10×10^{0}	307335.52	62.61
	Ours	7.54×10^{-4}	3.43×10^{-2}	2.15×10^{-1}	1.00×10^{0}	25.97	3.86

ization points was set to $100 \times \dim$. From the results, we observe that the multiple-initialization approach effectively captures multiple fixed points and accurately approximates solutions exhibiting kinks. Moreover, it demonstrates better scalability compared to WENO, while being more accurate and stable than ImplicitHJ. Although the use of multiple initialization increases computational cost, our method remains faster than both baseline methods.

3.3.2. Unsteady Kink. We now consider a problem exhibiting an unsteady kink. Let the Hamiltonian and initial function be $H(p) = \frac{1}{2}\|p\|_2^2$ and initial function $g(\mathbf{x}) = \sum_{i=1}^{d} g_0(x_i)$, where $g_0(x) = \begin{cases} x, & \text{if } x < 0 \\ 0, & \text{if } x \geq 0. \end{cases}$. The exact solution is given by

$$u(x,t) = \sum_{x_i < t/2} (x_i - t/2),$$

which exhibits a kink moving in time at $x = t/2$. In this problem, characteristics overlap in the region $0 < x < t$, leading to non-uniqueness of fixed points. We investigate whether the multiple initialization scheme can effectively capture such moving kinks.

The results of our multiple initialization scheme, compared to WENO and ImplicitHJ, are summarized in Table 6. Similar to the steady kink case, the multiple-initialization fixed point approach effectively captures solutions exhibiting unsteady kinks. It demonstrates superior scalability and is more accurate and stable than both WENO and ImplicitHJ. Although the use of multiple initialization increases computational cost, our method remains faster than both baseline approaches.

4. Conclusion. We have presented a novel fixed-point iteration framework for solving high-dimensional Hamilton–Jacobi equations via the Hopf–Lax formula. By reformulating the variational problem as a fixed-point equation, the proposed method

TABLE 6
Comparison of numerical and gradient errors, computation time, and memory usage for different dimensions using WENO, ImplicitHJ, and Ours methods on the non-steady kink example.

Dim	Method	L^2 Error	L^∞ Error	L^2 Grad Error	L^∞ Grad Error	Time (s)	Mem (MB)
1D	WENO	1.46×10^{-7}	2.02×10^{-3}	2.19×10^{-3}	2.40×10^{-1}	0.38	0.05
	ImplicitHJ	5.16×10^{-4}	1.33×10^{-1}	5.57×10^{-2}	9.57×10^{-1}	158959.15	0.27
	Ours	1.51×10^{-16}	1.39×10^{-7}	7.81×10^{-13}	2.10×10^{-4}	0.003	0.002
2D	WENO	6.84×10^{-5}	1.68×10^{-2}	2.25×10^{-3}	4.59×10^{-1}	9.072	4.43
	ImplicitHJ	1.01×10^{-4}	1.04×10^{-1}	1.62×10^{-2}	9.13×10^{-1}	169483.23	0.34
	Ours	3.17×10^{-15}	6.37×10^{-6}	7.81×10^{-10}	7.31×10^{-3}	0.21	0.08
3D	WENO	4.16×10^{-5}	1.76×10^{-1}	2.03×10^{0}	5.66×10^{0}	1734.16	642.54
	ImplicitHJ	2.70×10^{-3}	4.08×10^{-1}	6.89×10^{-2}	1.08×10^{0}	185356.57	0.42
	Ours	6.36×10^{-15}	2.41×10^{-5}	1.56×10^{-2}	8.27×10^{-1}	0.32	0.12
10D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	1.01×10^{-4}	3.02×10^{-2}	3.14×10^{-2}	6.32×10^{-1}	192581.22	0.95
	Ours	2.82×10^{-9}	1.13×10^{-3}	4.69×10^{-2}	1.00×10^{0}	1.20	0.38
50D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	2.01×10^{-4}	5.24×10^{-2}	1.08×10^{-1}	5.48×10^{-1}	256352.28	4.01
	Ours	1.72×10^{-6}	3.90×10^{-3}	6.25×10^{-2}	1.00×10^{0}	18.25	3.86
100D	WENO	N/A	N/A	N/A	N/A	N/A	N/A
	ImplicitHJ	3.32×10^{-1}	1.57×10^{0}	1.13×10^{0}	1.57×10^{0}	321914.77	7.82
	Ours	1.54×10^{-4}	3.43×10^{-2}	2.15×10^{-1}	1.00×10^{0}	25.97	3.86

enables direct, mesh-free evaluation of viscosity solutions and associated optimal controls without requiring spatial discretization, characteristic tracing, or repeated differentiation. Our theoretical analysis establishes convergence and error bounds under standard convexity and Lipschitz continuity assumptions, ensuring both accuracy and stability. Although the convexity requirement slightly limits the class of problems to which the method can be directly applied, numerical experiments in dimensions up to 100 demonstrate that the approach achieves high precision, low computational cost, and near dimension-independent scalability, outperforming classical grid-based solvers and recent deep learning methods in both efficiency and reliability. Overall, the fixed-point framework complements the limitations of existing classical numerical schemes and modern deep learning approaches, offering a scalable, provably accurate, and computationally efficient methodology for high-dimensional Hamilton–Jacobi problems, with potential applications in optimal control, robotics, finance, and other domains involving high-dimensional PDEs.

Several directions remain for future research. Extending the method to nonconvex or non-Lipschitz Hamiltonians would broaden its applicability to a wider class of problems. Incorporating a diffusion term to extend the method to second-order Hamilton–Jacobi equations could enable applications in stochastic control, mean-field games, and related areas. Furthermore, incorporating adaptive step-size control, momentum, anchoring, Anderson acceleration, or quasi-Newton updates may further enhance convergence speed and robustness, particularly in very high-dimensional settings. Finally, combining the fixed-point approach with implicit neural network architectures [3, 25] could open new possibilities for solving inverse problems and other complex high-dimensional tasks, thereby further extending the practical impact of the method.

REFERENCES

[1] M. Akian, S. Gaubert, and A. Lakhoua, *The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis*, SIAM Journal on Control and Optimization, 47 (2008), pp. 817–848.

[2] M. Ansari, A. Khajepour, and E. Esmailzadeh, *Application of level set method to optimal vibration control of plate structures*, Journal of Sound and Vibration, 332 (2013), pp. 687–700.

[3] S. Bai, J. Z. Kolter, and V. Koltun, *Deep equilibrium models*, Advances in neural information processing systems, 32 (2019).

[4] S. Bansal, A. Bajcsy, E. Ratner, A. D. Dragan, and C. J. Tomlin, *A hamilton-jacobi reachability-based framework for predicting and analyzing human motion for safe planning*, in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 7149–7155.

[5] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, *Hamilton-jacobi reachability: A brief overview and recent advances*, in 2017 IEEE 56th Annual Conference on Decision and Control (CDC), IEEE, 2017, pp. 2242–2253.

[6] S. Bryson and D. Levy, *High-order central weno schemes for multidimensional hamilton-jacobi equations*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1339–1369.

[7] Y. Cao and N. Wan, *Optimal proportional reinsurance and investment based on hamilton–jacobi–bellman equation*, Insurance: Mathematics and Economics, 45 (2009), pp. 157–162.

[8] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, International journal of computer vision, 22 (1997), pp. 61–79.

[9] P. Chen, J. Darbon, and T. Meng, *Hopf-type representation formulas and efficient algorithms for certain high-dimensional optimal control problems*, Computers & Mathematics with Applications, 161 (2024), pp. 90–120.

[10] Y. T. Chow, J. Darbon, S. Osher, and W. Yin, *Algorithm for overcoming the curse of dimensionality for time-dependent non-convex hamilton–jacobi equations arising from optimal control and differential games problems*, Journal of Scientific Computing, 73 (2017), pp. 617–643.

[11] Y. T. Chow, J. Darbon, S. Osher, and W. Yin, *Algorithm for overcoming the curse of dimensionality for state-dependent hamilton-jacobi equations*, Journal of Computational Physics, 387 (2019), pp. 376–409.

[12] E. Cristiani and M. Falcone, *Fast semi-lagrangian schemes for the eikonal equation and applications*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1979–2011.

[13] J. Darbon, P. M. Dower, and T. Meng, *Neural network architectures using min-plus algebra for solving certain high-dimensional optimal control problems and hamilton–jacobi pdes*, Mathematics of Control, Signals, and Systems, 35 (2023), pp. 1–44.

[14] J. Darbon, G. P. Langlois, and T. Meng, *Overcoming the curse of dimensionality for some hamilton–jacobi partial differential equations via neural network architectures*, Research in the Mathematical Sciences, 7 (2020), p. 20.

[15] J. Darbon and T. Meng, *On some neural network architectures that can represent viscosity solutions of certain high dimensional hamilton–jacobi partial differential equations*, Journal of Computational Physics, 425 (2021), p. 109907.

[16] J. Darbon and S. Osher, *Algorithms for overcoming the curse of dimensionality for certain hamilton–jacobi equations arising in control theory and elsewhere*, Research in the Mathematical Sciences, 3 (2016), p. 19.

[17] M. de León, J. C. Marrero, and D. M. de Diego, *Linear almost poisson structures and hamilton-jacobi equation. applications to nonholonomic mechanics*, arXiv preprint arXiv:0801.4358, (2008).

[18] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Féron, *Mathematical models for aircraft trajectory design: A survey*, in Air Traffic Management and Systems: Selected Papers of the 3rd ENRI International Workshop on ATM/CNS (EIWAC2013), Springer, 2014, pp. 205–247.

[19] H. H. Denman and L. H. Buch, *Solution of the hamilton-jacobi equation for certain dissipative classical mechanical systems*, Journal of Mathematical Physics, 14 (1973), pp. 326–329.

[20] L. C. Evans and P. E. Souganidis, *Differential games and representation formulas for solutions of hamilton-jacobi-isaacs equations*, Indiana University mathematics journal, 33 (1984), pp. 773–797.

[21] M. Falcone and R. Ferretti, *Semi-lagrangian schemes for hamilton–jacobi equations, discrete representation formulae and godunov methods*, Journal of computational physics, 175 (2002), pp. 559–575.

[22] M. Falcone and R. Ferretti, *Semi-Lagrangian approximation schemes for linear and Hamilton—Jacobi equations*, SIAM, 2013.

[23] W. H. FLEMING AND W. M. MCENEANEY, *A max-plus-based algorithm for a hamilton–jacobi–bellman equation of nonlinear filtering*, SIAM Journal on Control and Optimization, 38 (2000), pp. 683–710.

[24] P. A. FORSYTH AND G. LABAHN, *Numerical methods for controlled hamilton-jacobi-bellman pdes in finance*, Journal of Computational Finance, 11 (2007), p. 1.

[25] S. W. FUNG, H. HEATON, Q. LI, D. MCKENZIE, S. OSHER, AND W. YIN, *Jfb: Jacobian-free backpropagation for implicit networks*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 6648–6656.

[26] G. GILBOA AND S. OSHER, *Nonlocal operators with applications to image processing*, Multiscale Modeling & Simulation, 7 (2009), pp. 1005–1028.

[27] C. IMBERT, R. MONNEAU, AND H. ZIDANI, *A hamilton-jacobi approach to junction problems and application to traffic flows*, ESAIM: Control, Optimisation and Calculus of Variations, 19 (2013), pp. 129–166.

[28] G.-S. JIANG AND D. PENG, *Weighted eno schemes for hamilton–jacobi equations*, SIAM Journal on Scientific computing, 21 (2000), pp. 2126–2143.

[29] K. KHANIN AND A. SOBOLEVSKI, *Particle dynamics inside shocks in hamilton–jacobi equations*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 368 (2010), pp. 1579–1593.

[30] J. A. LAVAL AND L. LECLERCQ, *The hamilton–jacobi partial differential equation and the three representations of traffic flow*, Transportation Research Part B: Methodological, 52 (2013), pp. 17–30.

[31] F. L. LEWIS, D. M. DAWSON, AND C. T. ABDALLAH, *Robot manipulator control: theory and practice*, CRC Press, 2003.

[32] Z. LI, N. KOVACHKI, K. AZIZZADENESHELI, B. LIU, K. BHATTACHARYA, A. STUART, AND A. ANANDKUMAR, *Fourier neural operator for parametric partial differential equations*, arXiv preprint arXiv:2010.08895, (2020).

[33] F. LIN AND R. D. BRANDT, *An optimal control approach to robust control of robot manipulators*, IEEE Transactions on robotics and automation, 14 (1998), pp. 69–77.

[34] L. LU, P. JIN, AND G. E. KARNIADAKIS, *Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*, arXiv preprint arXiv:1910.03193, (2019).

[35] W. M. MCENEANEY, *Max-plus methods for nonlinear control and estimation*, vol. 2, Springer Science & Business Media, 2006.

[36] J. C. MIÑANO, P. BENÍTEZ, AND A. SANTAMARÍA, *Hamilton-jacobi equation in momentum space*, Optics Express, 14 (2006), pp. 9083–9092.

[37] I. MITCHELL, A. BAYEN, AND C. J. TOMLIN, *Computing reachable sets for continuous dynamic games using level set methods*, Submitted January, (2004).

[38] S. OSHER, *A level set formulation for the solution of the dirichlet problem for hamilton–jacobi equations*, SIAM Journal on Mathematical Analysis, 24 (1993), pp. 1145–1152.

[39] S. OSHER, R. FEDKIW, AND K. PIECHOR, *Level set methods and dynamic implicit surfaces*, Appl. Mech. Rev., 57 (2004), pp. B15–B15.

[40] S. OSHER AND N. PARAGIOS, *Geometric level set methods in imaging, vision, and graphics*, Springer Science & Business Media, 2007.

[41] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.

[42] S. OSHER AND C.-W. SHU, *High-order essentially nonoscillatory schemes for hamilton–jacobi equations*, SIAM Journal on numerical analysis, 28 (1991), pp. 907–922.

[43] Y. PARK AND S. OSHER, *Neural implicit solution formula for efficiently solving hamilton-jacobi equations*, (2025), https://arxiv.org/abs/2501.19351, https://arxiv.org/abs/2501.19351.

[44] C. PARZANI AND S. PUECHMOREL, *On a hamilton-jacobi-bellman approach for coordinated optimal aircraft trajectories planning*, Optimal Control Applications and Methods, 39 (2018), pp. 933–948.

[45] J. QIU AND C.-W. SHU, *Hermite weno schemes for hamilton–jacobi equations*, Journal of Computational Physics, 204 (2005), pp. 82–99.

[46] 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 (2019), pp. 686–707.

[47] S. RAJEEV, *A hamilton–jacobi formalism for thermodynamics*, Annals of Physics, 323 (2008), pp. 2265–2285.

[48] A. SIDERIS AND J. E. BOBROW, *An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems*, in Proceedings of the 2005, American Control Confer-

ence, 2005., IEEE, 2005, pp. 2275–2280.

[49] J. Sirignano and K. Spiliopoulos, *Dgm: A deep learning algorithm for solving partial differential equations*, Journal of computational physics, 375 (2018), pp. 1339–1364.

[50] L. Yang, S. Liu, T. Meng, and S. J. Osher, *In-context operator learning with data prompts for differential equation problems*, Proceedings of the National Academy of Sciences, 120 (2023), p. e2310142120.

[51] B. Yu et al., *The deep ritz method: a deep learning-based numerical algorithm for solving variational problems*, Communications in Mathematics and Statistics, 6 (2018), pp. 1–12.