A Characteristic-Based Deep Learning Framework for Hamilton–Jacobi Equations with Application to Optimal Transport

Stanley Osher

joint work with Yesom Park, Shu Liu and Mo Zhou

UCLA Department of Mathematics

IPAM Workshop 2025

July 14, 2025

Outline

Hamilton-Jacobi Equations

Hamilton-Jacobi (HJ) PDE is a class of PDE of the following form:

$$\begin{cases} u_t + H(\nabla u) = 0 & \text{in } \Omega \times (0, T) \\ u = g & \text{on } \Omega \times \{t = 0\}, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ is the spatial domain, $H : \mathbb{R}^d \to \mathbb{R}$ is the Hamiltonian and $g : \Omega \to \mathbb{R}$ is the initial function.

- Non-unique
- Non-smooth, irrespective of the smoothness of the initial conditions or the Hamiltonian.

Prior Works: Classical Numerical Methods

1. Mesh-based methods, such as ENO/WENO (OS91; QS05; OS88)
   - Suffer from curse of dimensionality.

2. Hopf-formula-based causality-free method (DO16; CDOY17; CDOY19)

$$u(\mathbf{x}, t) = \inf_{\mathbf{y}} \left\{ t H^* \left(\frac{\mathbf{x} - \mathbf{y}}{t} \right) + g(\mathbf{y}) \right\}, \tag{2}$$

   where $H^*(\mathbf{z}) = \sup_{\mathbf{v} \in \mathbb{R}^d} \left\{ \mathbf{z}^{\mathrm{T}} \mathbf{v} - H(\mathbf{v}) \right\}$.
   - Suffer from computing Legendre transform.

3. PMP-based optimal control approaches (KW15; KW17)
   - Suffer from reduced practical effectiveness due to computing every single ODE trajectories.

Prior Works: Scientific ML Methods

1. Physics-Informed Neural Networks (PINNs) (DPT94; RPK19) solve the PDE by minimizing the integrated squared residual of the HJ PDE and the initial condition:

$$\mathcal{L}(u) = \int_0^T \int_\Omega \Big(u_t + H(\nabla u)\Big)^2 + \lambda \int_\Omega (u - g)^2 .$$

   - No guarantee of obtaining the viscosity solution.

2. Specialized neural network architectures that express Hopf formulas (DLM20; DDM23)

   - Limited to specific HJ PDEs.

Consider the HJ PDE

$$\begin{cases} u_t + H(\nabla u) = 0 & \text{in } \Omega \times (0, T) \\ u = g & \text{on } \Omega \times \{t = 0\}. \end{cases} \tag{3}$$

System of characteristic ODEs for (3) is given by the following:

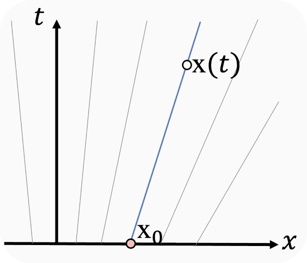
$$\begin{cases} \dot{\mathbf{x}} = \nabla H & \text{(4a)} \\ \dot{u} = q + \mathbf{p}^{\mathrm{T}} \nabla H = -H + \mathbf{p}^{\mathrm{T}} \nabla H & \text{(4b)} \\ \dot{q} = 0 & \text{(4c)} \\ \dot{\mathbf{p}} = 0, & \text{(4d)} \end{cases}$$

where the variables q and $\mathbf{p}$ are shorthand for the partial derivatives $q = u_t$ and $\mathbf{p} = \nabla u$, respectively.

Characteristic Lines

The characteristic
emanated from $\mathbf{x}(0) = \mathbf{x}_0 \in \Omega$ is a straight line

$$\mathbf{x}(t) = t\nabla H(\mathbf{p}) + \mathbf{x}_0,$$



implying that

$$\begin{aligned} u(t, \mathbf{x}(t)) &= -tH(\mathbf{p}) + t\mathbf{p}^{\mathrm{T}}\nabla H(\mathbf{p}) + u(\mathbf{x}_0, 0) \\ &= -tH(\mathbf{p}) + t\mathbf{p}^{\mathrm{T}}\nabla H(\mathbf{p}) + g(\mathbf{x}_0) \\ &= -tH(\mathbf{p}) + t\mathbf{p}^{\mathrm{T}}\nabla H(\mathbf{p}) + g(\mathbf{x} - t\nabla H(\mathbf{p})). \end{aligned}$$

Implicit Solution Formula

Substituting $\mathbf{p} = \nabla u(\mathbf{x}, t)$, we attain the following **implicit solution formula** for the HJ PDEs (3) (PO25):

$$u(\mathbf{x}, t) = -tH(\nabla u) + t\nabla u^{\mathrm{T}} \nabla H(\nabla u) + g(\mathbf{x} - t\nabla H(\nabla u)). \tag{5}$$

Theorem 1 (Convex Hamiltonian)

Assume the Hamiltonian H is differentiable and satisfies

$$\begin{cases} \mathbf{p} \mapsto H(\mathbf{p}) \ \textit{is strictly convex or concave}, \\ \lim_{|\mathbf{p}| \to \infty} \frac{H(\mathbf{p})}{|\mathbf{p}|} = +\infty, \end{cases} \tag{6}$$

and the initial function g is l.s.c. Then, the continuous function u that satisfies the implicit solution formula (5) *coincides with the **Hopf-Lax formula*** (2) *of* (3) *a.e.*

- Similarly, when g is convex (concave), the implicit solution formula coincides to the **Hopf formula**, representing the viscosity solution in this case.

Learning Implicit Solution with Neural Networks

Building upon the implicit solution formula, we propose the following minimization problem:

$$\min_u \mathcal{L}(u) \coloneqq \int_0^T \int_\Omega \Big(u + tH(\nabla u) - t\nabla u^{\mathrm{T}} \nabla H(\nabla u) - g(\mathbf{x} - t\nabla H(\nabla u))\Big)^2 \, \mathrm{d}\mathbf{x} \, \mathrm{d}t. \tag{7}$$

- **Neural representation**: Parameterize u using a standard artificial neural network $u_\theta : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$.
- **Mesh-free**: Approximate the integral of (7) using Monte Carlo methods with randomly sampled collocation points.
- **Unsupervised Learning**: No ground truth solution data is required —the network learns the viscosity solution solely from H and g.

Algorithm

Algorithm 1 Algorithm for Learning Implicit Solution of HJ PDEs

Initialize the network u_θ with an initial network parameter θ_0.
for $n = 0, \cdots, N$ **do**
    Randomly sample M collocations points $\left\{(\mathbf{x}_j, t_j)\right\}_{j=1}^{M} \sim U\left(\Omega \times [0, T]\right)$.
    Calculate the loss by Monte Carlo integration

$$\hat{\mathcal{L}}\left(\theta_n\right) = \frac{1}{M}\sum_{j=1}^{M} \mathcal{S}\left(u_{\theta_n}\left(\mathbf{x}_j, t_j\right)\right)^2 .$$

    Update θ_n by gradient descent with a step size $\alpha > 0$

$$\theta_{n+1} \leftarrow \theta_n - \alpha \nabla_\theta \hat{\mathcal{L}}\left(\theta_n\right).$$

end for
return u_{θ_N} as the predicted viscosity solution to the HJ PDE (3).

Comparison with Prior Works

Question: Does this approach effectively address the key limitations of previous works?

- The curse of dimensionality associated with mesh-based methods.
  ☺ The proposed approach is **mesh-free**.
- The computational challenges of the Legendre transform in Hopf formula-based methods.
  ☺ The proposed approach **does not require the Legendre transform**.
- The inefficiency of computing single characteristic trajectories in optimal control-based methods.
  ☺ The proposed approach **does not compute individual trajectories**.

Accuracy

Quantitative results for convex problems in dimensions $d = 1, 2, 3, 10, 40$. The mean squared errors (MSE) w.r.t. the exact solution and the memory usage (Mem, in MB) for storing the predicted solutions are reported.

Example 1 (Convex): $H(\mathbf{p}) = \frac{1}{2}\|\mathbf{p}\|_2^2$ and $g(\mathbf{x}) = \|\mathbf{x}\|_1$.

	$d=1$		$d=2$		$d=3$		$d=10$		$d=40$	
Method	MSE	Mem	MSE	Mem	MSE	Mem	MSE	Mem	MSE	Mem
Ours	1.14E-7	0.06	1.91E-7	0.06	3.21E-6	0.06	2.56E-5	0.06	1.30E-3	0.07
PINNs	2.39E-6	0.06	2.14E-5	0.06	1.98E-4	0.06	5.78E-3	0.06	8.00	0.07
WENO (same Mem)	3.84E-5	0.06	1.3E-3	0.06	3.68E-3	0.06	N/A	N/A	N/A	N/A
WENO (same MSE)	1.14E-7	6.17	1.83E-7	51498.41	N/A	N/A	N/A	N/A	N/A	N/A

Example 2 (Concave): $H(\mathbf{p}) = -\frac{1}{2}\|\mathbf{p}\|_2^2$ and $g(\mathbf{x}) = \|\mathbf{x}\|_1$.

	$d=1$		$d=2$		$d=3$		$d=10$		$d=40$	
Method	MSE	Mem	MSE	Mem	MSE	Mem	MSE	Mem	MSE	Mem
Ours	8.59E-6	0.06	1.10E-4	0.06	1.15E-4	0.06	1.63E-4	0.06	1.23E-3	0.07
PINNs	1.40E-5	0.06	2.98E-4	0.06	5.53E-4	0.06	2.20E-2	0.06	11.90	0.07
WENO (same Mem)	1.01E-6	0.06	1.30E-3	0.06	1.35E-1	0.06	N/A	N/A	N/A	N/A
WENO (same MSE)	–	–	1.11E-4	3.81	1.53E-4	686.33	N/A	N/A	N/A	N/A

Effect of Network Size

- **Example 3:** $H(\mathbf{p}) = \|\mathbf{p}\|_2$ and the g is the signed distance function from two disjoint $(d-1)$-spheres.
- **Example 4:** $H(\mathbf{p}) = \|\mathbf{p}\|_\infty$ and $g(\mathbf{x}) = \|\mathbf{x}\|_1$.

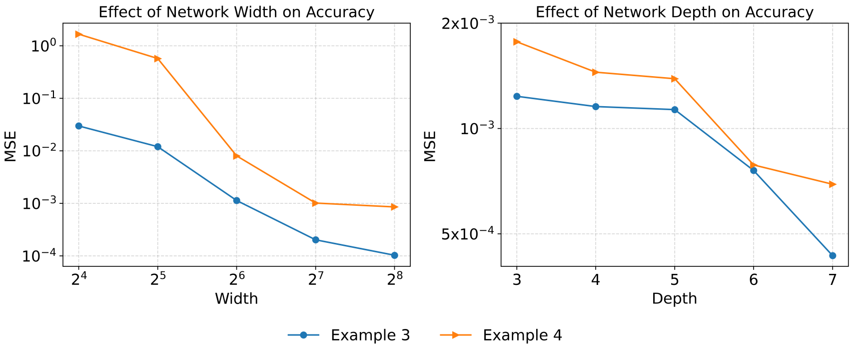


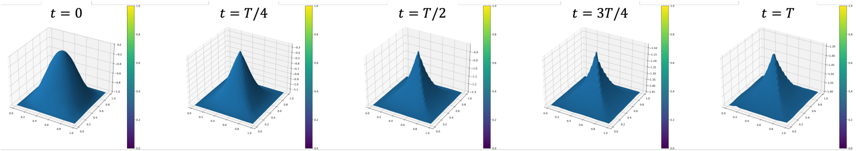
Figure 1: Effects of network size on the solution accuracy (MSE) of **40-dimensional** case. (Left) the depth is fixed at 5 while varying the width; (Right) the width is fixed at 64 while varying the depth. Results demonstrate that increasing the network size enhances accuracy.

Numerical Results

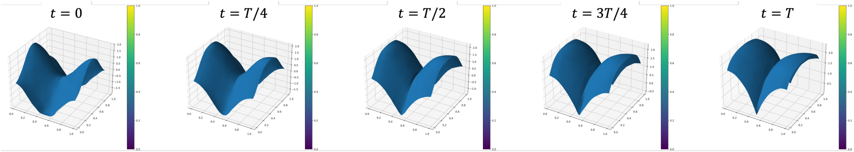
Table 1: Nonconvex examples

Problem	Hamiltonian H	Initial function g
Example 5	$-\cos\left(\sum_{i=1}^{d} u_{x_i} + 1\right)$	$-\cos\left(\frac{\pi}{d}\sum_{i=1}^{d} x_i\right)$
Example 6	$\sin\left(u_x + u_y\right)$	$\pi\left(\lvert y\rvert - \lvert x\rvert\right)$
Example 7	$u_x u_y$	$\sin(x) + \cos(y)$
Example 8	$\left(u_x + u_y + 1\right)^{1/2}$	$\frac{1}{4}\left(\cos\left(2\pi x\right) - 1\right)\left(\cos\left(2\pi y\right) - 1\right) - 1$
Example 9	$-\left(u_x + u_y + 1\right)^{1/2}$	$\cos\left(2\pi x\right) - \cos\left(2\pi y\right)$
Example 10	$u_x^3 - u_x$	$-\frac{1}{10}\cos\left(5x\right)$

Numerical Results



(a) Example 8 (Eikonal)



(b) Example 9 (Combustion)

Optimal Transport

For two distributions $\mu, \nu \in \mathcal{P}(\Omega)$ supported on $\Omega \subset \mathbb{R}^d$, optimal transport (OT) problem seeks a map T that transforms μ to ν whilst minimizing the cost ℓ.

Monge Formulation

$$W_c(\mu, \nu) \coloneqq \inf_{T_\sharp \mu = \nu} \int_\Omega \ell(\mathbf{x} - T(\mathbf{x}))\, \mathrm{d}\mu(\mathbf{x}). \tag{8}$$

Benamou-Brenier fluid dynamical formulation

$$\inf_v \mathbb{E}_\mu \left[\int_0^{t_f} \ell(v(\mathbf{x}(t), t))\, \mathrm{d}t \right] \tag{9}$$

$$\text{s.t. } \dot{\mathbf{x}} = v \tag{10}$$

$$\mathbf{x}(0) \sim \mu, \ \mathbf{x}(t_f) \sim \nu, \tag{11}$$

HJ equation

$$\begin{cases} \frac{\partial u}{\partial t} - h(\nabla u) = 0 & \text{in } \Omega \times (0, t_f) \\ u = g & \text{on } \Omega \times \{t = 0\}, \end{cases} \tag{12}$$

Method of Characteristics

The viscosity solution is theoretically characterized by the **characteristic ODEs**

$$\begin{cases} \dot{\mathbf{x}} = \nabla h & (13a) \\ \dot{u} = -h + \mathbf{p}^{\mathrm{T}} \nabla h & (13b) \\ \dot{\mathbf{p}} = 0, & (13c) \end{cases}$$

Bidirectional OT Map

A bidirectional formulation of the OT map arises from the forward and backward characteristic flows of the associated HJ equation:

- Forward Map : $T^{\star}(\mathbf{x}) = \mathbf{x} - t_f \nabla h \left(\nabla u \left(\mathbf{x}, 0\right)\right), \quad \mathbf{x} \sim \mu, \quad (14)$
- Backward Map: $\left(T^{\star}\right)^{-1}(\mathbf{y}) = \mathbf{y} + t_f \nabla h \left(\nabla u \left(\mathbf{y}, t_f\right)\right), \quad \mathbf{y} \sim \nu. \quad (15)$

Characteristic-based OT Learning

We propose a deep learning framework for OT based on HJ characteristics, consisting of two key steps:

1. Learning the Viscosity Solution: Train a neural network u_θ to approximate the viscosity solution using the implicit solution formula of the HJ equation.
2. Recovering the OT Map: Obtain the bidirectional OT map from the learned solution by the characteristic-based formulation

$$
\begin{aligned}
T_\theta(\mathbf{x}) &= \mathbf{x} - t_f \nabla h\left(\nabla u_\theta\left(\mathbf{x}, 0\right)\right), \quad \mathbf{x} \sim \mu, \\
\left(T_\theta\right)^{-1}\left(\mathbf{y}\right) &= \mathbf{y} + t_f \nabla h\left(\nabla u_\theta\left(\mathbf{y}, t_f\right)\right), \quad \mathbf{y} \sim \nu,
\end{aligned}
$$

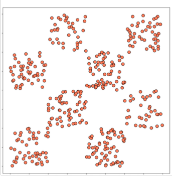
   without numerical integration of ODEs.

Method	**Optimization**	**# Networks**	**OT direction**	**Sampling**	**Optimality of T**
Dual Formulation	Min-Max	Two	One-way	Direct	No
Dynamical Models	Min	Single	Bidirectional	Iterative	No
HJ-based (Proposed)	Min	Single	Bidirectional	Direct	Yes

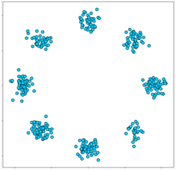
Table 2: Comparison of key features across different OT model approaches.

2D Examples

Data



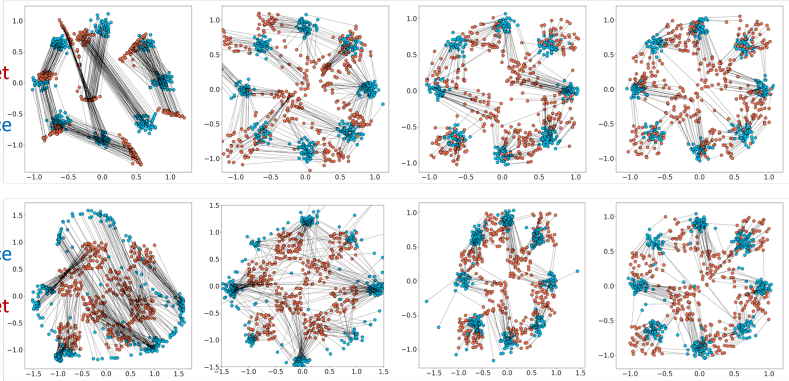
Target Distribution



Source Distribution

Target ↓ Source

Source ↓ Target



NOT (strong) | NOT (weak) | HJ-PINN | **Ours**

Accuracy of Learned OT Map

Evaluate the learned transport map in the Gaussian-to-Gaussian setting, $\mu = \mathcal{N}\left(\mathbf{0}, \Sigma_\mu\right)$ and $\nu = \mathcal{N}\left(\mathbf{0}, \Sigma_\nu\right)$, where a closed-form solution to the OT is available.

Table 3: Quantitative comparison of L^2 error (↓) across OT methods in increasing dimensions.

Model	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$
NOT	77.248	125.419	114.056	176.086	182.287	196.831
WGAN-QC	1.596	5.897	31.0367	59.314	113.237	141.407
LS	5.806	9.781	15.963	25.232	41.445	55.360
MM-v1	0.161	0.172	0.173	0.210	**0.374**	**0.415**
HJ-PINN	0.080	0.069	0.163	0.458	0.576	1.683
Ours	**0.010**	**0.021**	**0.086**	**0.146**	0.436	0.858

Computational Efficiency

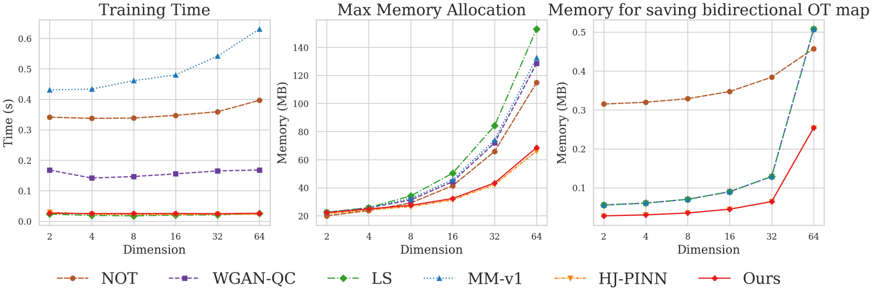


Figure 2: Comparison of training time (s/epoch), maximum memory usage (MB) during training, and memory consumption (MB) for saving bidirectional OT maps across models and dimensions.

Color Transfer

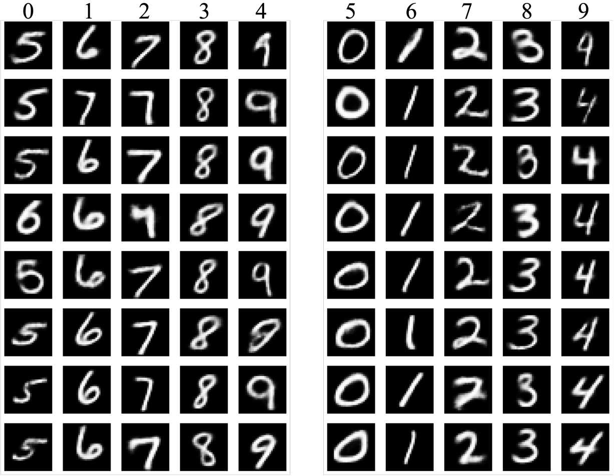
Learn an OT map between the color distributions of source and target images, where RGB values are interpreted as samples from probability measures in $\mathbb{R}^3$.

Table 4: Quantitative comparison of methods on three datasets: Earth mover distance (EMD) and histogram intersection (HI) between color distributions of target and transported images across three datasets.

Method	Winter-Summer		Monet-Image		Gogh-Image	
	EMD (↓)	HI (↑)	EMD (↓)	HI(↑)	EMD (↓)	HI(↑)
HisMatching	0.0012	0.7296	0.0013	0.7532	0.0010	0.7668
Reinhard	0.0013	0.6255	0.0012	0.7255	0.0009	0.7406
NOT	0.0008	0.8002	0.0008	0.8210	0.0008	0.8247
MM-v1	0.0014	0.7295	0.0011	0.7722	0.0007	0.8265
Ours	**0.0005**	**0.8914**	**0.0004**	**0.9174**	**0.0003**	**0.9117**

Class-Conditional OT

Class-wise OT on the MNIST dataset (28 × 28), transporting digits from $\{0, 1, 2, 3, 4\}$ to its corresponding digit in $\{5, 6, 7, 8, 9\}$ (i.e., $0 \to 5$, $1 \to 6$, ..., $4 \to 9$).



(a) Forward (b) Backward

Conclusion

- Proposed a novel implicit solution formula for HJ PDEs derived from the characteristics.
- Recovered the classical Hopf formula in convex settings, while simplifying it by eliminating the need for Legendre transforms.
- Developed a simple and effective deep learning-based method for solving high-dimensional HJ PDEs, mitigating the curse of dimensionality.
- Demonstrated the scalability and effectiveness of the proposed method across various high-dimensional and nonconvex benchmark problems.
- Showed that the implicit formula, together with characteristic flows, enables an efficient and principled approach to solving optimal transport problems.

References I

[CDOY17] Yat Tin Chow, Jérôme Darbon, Stanley Osher, and Wotao 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), 617–643.

[CDOY19] ______, *Algorithm for overcoming the curse of dimensionality for state–dependent hamilton–jacobi equations*, Journal of Computational Physics **387** (2019), 376–409.

[DDM23] Jérôme Darbon, Peter M Dower, and Tingwei 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), no. 1, 1–44.

[DLM20] Jérôme Darbon, Gabriel P Langlois, and Tingwei Meng, *Overcoming the curse of dimensionality for some hamilton–jacobi partial differential equations via neural network architectures*, Research in the Mathematical Sciences **7** (2020), no. 3, 20.

[DO16] Jérôme Darbon and Stanley 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), no. 1, 19.

[DPT94] MWMG Dissanayake and Nhan Phan-Thien, *Neural-network-based approximations for solving partial differential equations*, communications in Numerical Methods in Engineering **10** (1994), no. 3, 195–201.

[KW15] Wei Kang and Lucas Wilcox, *A causality free computational method for hjb equations with application to rigid body satellites*, AIAA Guidance, Navigation, and Control Conference, 2015, p. 2009.

[KW17] Wei Kang and Lucas C Wilcox, *Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and hjb equations*, Computational Optimization and Applications **68** (2017), 289–315.

References III

[OS88] Stanley Osher and James A Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations*, Journal of computational physics **79** (1988), no. 1, 12–49.

[OS91] Stanley Osher and Chi-Wang Shu, *High-order essentially nonoscillatory schemes for hamilton–jacobi equations*, SIAM Journal on numerical analysis **28** (1991), no. 4, 907–922.

[PO25] Yesom Park and Stanley Osher, *Neural implicit solution formula for efficiently solving hamilton-jacobi equations*, arXiv preprint, arxiv2501.19351 (2025).

[QS05] Jianxian Qiu and Chi-Wang Shu, *Hermite weno schemes for hamilton–jacobi equations*, Journal of Computational Physics **204** (2005), no. 1, 82–99.

References IV

[RPK19] 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** (2019), 686–707.

Thank you!