OVERCOMING THE CURSE OF DIMENSIONALITY FOR SOME HAMILTON–JACOBI PARTIAL DIFFERENTIAL EQUATIONS VIA NEURAL NETWORK ARCHITECTURES

JÉRÔME DARBON, GABRIEL P. LANGLOIS, AND TINGWEI MENG

Abstract. We propose new and original mathematical connections between Hamilton-Jacobi (HJ) partial differential equations (PDEs) with initial data and neural network architectures. Specifically, we prove that some classes of neural networks correspond to representation formulas of HJ PDE solutions whose Hamiltonians and initial data are obtained from the parameters of the neural networks. These results do not rely on universal approximation properties of neural networks; rather, our results show that some classes of neural network architectures naturally encode the physics contained in some HJ PDEs. Our results naturally yield efficient neural network-based methods for evaluating solutions of some HJ PDEs in high dimension without using grids or numerical approximations. We also present some numerical results for solving some inverse problems involving HJ PDEs using our proposed architectures.

1. Introduction

The Hamilton–Jacobi (HJ) equations are an important class of partial differential equation (PDE) models that arise in many scientific disciplines, e.g., physics [6, 22, 23, 29, 87], imaging science [34, 35, 36], game theory [12, 21, 46, 70], and optimal control [8, 43, 49, 50, 95]. Exact or approximate solutions to these equations then give practical insight about the models in consideration. We consider here HJ equations specified by a Hamiltonian function $H : \mathbb{R}^n \to \mathbb{R}$ and initial data $J : \mathbb{R}^n \to \mathbb{R}$

\begin{equation}
\begin{cases}
\frac{\partial S}{\partial t}(x, t) + H(\nabla_x S(x, t)) = \frac{\epsilon}{2} \Delta_x S(x, t) & \text{in } \mathbb{R}^n \times (0, +\infty), \\
S(x, 0) = J(x) & \text{in } \mathbb{R}^n \times \{0\},
\end{cases}
\end{equation}

where $\epsilon \geq 0$, $\frac{\partial S}{\partial t}(x, t)$ and $\nabla_x S(x, t) = \left( \frac{\partial S}{\partial x_1}(x, t), \ldots, \frac{\partial S}{\partial x_n}(x, t) \right)$ denote the partial derivative with respect to $t$ and the gradient vector with respect to $x$ of the function $(x, t) \mapsto S(x, t)$, and $\Delta_x S(x, t) = \sum_{i=1}^{n} \frac{\partial^2 S}{\partial x_i^2}(x, t)$.

We wish to compute the viscosity solution of (1) for a given $x \in \mathbb{R}^n$ and $t > 0$, $J(x, 0)$. The viscosity solution rarely admits a closed-form expression, and in general it must be computed with numerical algorithms or other methods tailored for the Hamiltonian $H$, initial data $J$, and dimension $n$.

The dimensionality, in particular, matters significantly because in many applications involving HJ PDE models, the dimension $n$ is extremely large. In imaging problems, for example, the vector $x$ typically corresponds to a noisy image whose entries are its pixel values, and the associated Hamilton–Jacobi equations describe the solution to an image denoising convex optimization problem [34, 35]. Denoising a 1080 x 1920 standard full HD image on a smartphone, for example, corresponds to solving a HJ PDE in dimension $n = 1080 \times 1920 = 2,073,600$.

Unfortunately, standard grid-based numerical algorithms for PDEs are impractical when $n > 5$. Such algorithms employ grids to discretize the spatial and time domain, and the number of grid points required to evaluate accurately solutions of PDEs grows exponentially with the dimension $n$. It is therefore essentially impossible in practice to numerically solve PDEs in high dimension using grid-based algorithms, even with sophisticated high-order accuracy methods for HJ PDEs such as ENO [106], WENO [72], and DG [67]. This problem severely limits the usefulness of PDE models and is known as the curse of dimensionality [16].

Overcoming the curse of dimensionality in general remains an open problem, but for HJ PDEs several methods have been proposed to solve it. These include, but are not limited to, max-plus algebra methods [2, 3, 44, 51, 54, 95, 97, 98], dynamic programming and reinforcement learning [4, 18], tensor decomposition techniques [40, 65, 126], sparse grids [19, 53, 77], model order reduction [5, 89], polynomial approximation.
Among these methods, neural networks have become increasingly popular tools to solve PDEs [7, 14, 13, 15, 17, 26, 27, 38, 34, 39, 42, 47, 52, 56, 57, 58, 66, 68, 69, 73, 79, 80, 84, 85, 86, 89, 94, 99, 100, 103, 105, 108, 116, 119, 120, 122, 123, 124, 128, 130, 131, 132] and inverse problems involving PDEs [93, 92, 101, 102, 107, 111, 112, 114, 115, 113, 127, 131, 134, 135]. Their popularity is due to universal approximation theorems that state that neural networks can approximate broad classes of (high-dimensional, nonlinear) functions on compact sets [31, 63, 64, 109]. These properties, in particular, have been recently leveraged to approximate solutions to high-dimensional nonlinear HJ PDEs [57, 122] and for the development of physics-informed neural networks that aim to solve supervised learning problems while respecting any given laws of physics described by a set of nonlinear PDEs [113].

In this paper, we propose some neural network architectures that exactly represent viscosity solutions to HJ PDEs of the form of (1), where the Hamiltonians $H$ and initial data $J$ are obtained from the parameters of the neural network architectures. In other words, we show that some neural networks correspond to representation formulas of HJ PDE solutions.

Contributions of this paper. In this paper, we prove that some classes of shallow neural networks are, under certain conditions, viscosity solutions to Hamilton–Jacobi equations. Specifically, we propose a first neural network architecture (depicted in Fig. 1). We show in Thm. 3.1 that under certain conditions this neural network architecture represents the viscosity solution of a set of first-order HJ PDEs of the form of (1) (with $\epsilon = 0$), where the Hamiltonians and the initial data are obtained from the parameters. As a corollary of this result for the one-dimensional case, we propose a second neural network architecture (depicted in Fig. 2) that represents the spatial gradient of the viscosity solution of the HJ PDE above in 1D and show in Proposition 3.1 that under appropriate conditions this neural network corresponds to entropy solutions of some conservation laws in 1D. Finally, we propose a third neural network architecture (depicted in Fig. 3) and show in Proposition 3.2 that it represents the solution to a second-order HJ PDE (1) (with $\epsilon > 0$ and Hamiltonian $H = -\frac{1}{2} \| \cdot \|_2^2$), where the initial data is obtained from the parameters.

Let us emphasize that the two proposed architectures for representing solution of HJ PDEs allow us to numerically evaluate solutions of these HJ PDEs in high dimension without using grids.

We stress that our results do not rely on universal approximation properties of neural networks. Our results show that the physics contained in some HJ PDEs can naturally be encoded by some classes of neural network architectures. Our results also suggest interpretations of some neural network architectures in terms of solutions to PDEs.

We also test the proposed neural network architectures on some inverse problems. To do so, we consider the following problem. Given training data sampled from the solution $S$ of a first-order HJ PDE (1) (with $\epsilon = 0$) with unknown initial function $J$ and Hamiltonian $H$, we aim to recover the unknown initial function. After the training process using the Adam optimizer, the trained neural network with input time variable $t = 0$ gives an approximation to the initial function $J$. Moreover, the parameters in the trained neural network also provide partial information on the Hamiltonian $H$. The parameters only approximate the Hamiltonian at certain points, however, and therefore do not give complete information about the function.

We show the experimental results on several examples. Our numerical results show that this problem cannot generally be solved using Adam optimizer with high accuracy. In other words, while the theoretical results show that the neural network representation to some HJ PDEs are exact, the Adam optimizer for training the proposed networks in this paper sometimes give large errors in some of our inverse problems, and as such there is no guarantee that the algorithm works well for the proposed networks.

Organization of this paper. In Sect. 2 we briefly review shallow neural networks and concepts of convex analysis that will be used throughout this paper. In Sect. 3 we establish connections between several classes of neural network architectures and viscosity solutions to HJ equations and one-dimensional conservation laws. The mathematical set-up for establishing these connections is described in Sect. 3.1 our main results, which concern first-order HJ equations, are described in Sect. 3.2 and extensions of these results to one-dimensional conservation laws and a subclass of second-order HJ equations are presented in Sects. 3.3 and 3.4, respectively. In Sect. 4 we perform numerical experiments to test the effectiveness of our proposed
architectures for solving some inverse problems. Finally, we draw some conclusions and directions for future work in Sect. 5.

2. Background

In this section, we introduce mathematical concepts that we will use in this paper. We review the standard structure of shallow neural networks from a mathematical point of view in Sect. 2.1 and present some fundamental definitions and results in convex analysis in Sect. 2.2. For the notation, we use $\mathbb{R}^n$ to denote the $n$-dimensional Euclidean space. The Euclidean scalar product and Euclidean norm on $\mathbb{R}^n$ are denoted by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|_2$. The set containing all $m \times n$ matrices with real entries is denoted by $M_{m,n}(\mathbb{R})$.

2.1. Shallow neural networks. Neural networks provide architectures for constructing complicated non-linear functions from simple building blocks. Common neural network architectures in applications include, for example, feedforward neural networks in statistical learning, recurrent neural networks (RNNs) in natural language processing, and convolutional neural networks (CNNs) in imaging science. In this paper, we focus on shallow neural networks, a subclass of feedforward neural networks that typically consist of one hidden layer and one output layer. We give here a brief mathematical introduction to shallow neural networks. For more details, we refer the readers to [55, 88, 121] and the references listed therein.

A shallow neural network with one hidden layer and one output layer is a composition of affine functions with a nonlinear function. A hidden layer with $m \in \mathbb{N}$ neurons comprises $m$ affine functions of an input $x \in \mathbb{R}^n$ with weights $w_i \in \mathbb{R}^n$ and biases $b_i \in \mathbb{R}$:

$$\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \ni (x, w_i, b_i) \mapsto \langle w_i, x \rangle + b_i.$$ 

These $m$ affine functions can be succinctly written in vector form as $Wx + b$, where the matrix $W \in M_{m,n}(\mathbb{R})$ has for rows the weights $w_i$, and the vector $b \in \mathbb{R}^m$ has for entries the biases $b_i$. The output layer comprises a nonlinear function $\sigma : \mathbb{R}^m \to \mathbb{R}$ that takes for input the vector $Wx + b$ of affine functions and gives the number

$$\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \ni (x, w_i, b_i) \mapsto \sigma(Wx + b).$$

The nonlinear function $\sigma$ is called the activation function of the output layer.

In Sect. 3, we will consider the following problem: Given data points $\{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^n \times \mathbb{R}$, infer the relationship between the input $x_i$’s and the output $y_i$’s. To infer this relation, we assume that the output takes the form (or can be approximated by) $y_i = \sigma(Wx_i + b)$ for some known activation function $\sigma$, unknown matrix of weights $W \in M_{m,n}(\mathbb{R})$, and unknown vector of bias $b$. A standard approach to solve such a problem is to estimate the weights $w_i$ and biases $b_i$ so as to minimize the mean square error

$$\{\langle \bar{w}_i, b_i \rangle\}_{i=1}^m \in \arg \min_{w_i \in \mathbb{R}^n, b_i \in \mathbb{R}} \left\{ \frac{1}{N} \sum_{i=1}^N \left( \sigma(Wx_i + b) - y_i \right)^2 \right\}.$$ 

In the field of machine learning, solving this minimization problem is called the learning or training process. The data $\{(x_i, y_i)\}_{i=1}^N$ used in the training process is called training data. Finding a global minimizer is generally difficult due to the complexity of the minimization problem and that the objective function is not convex with respect to the weights and biases. State-of-the-art algorithms for solving these problems are stochastic gradient descent based methods with momentum acceleration, such as the Adam optimization algorithm for neural networks [81]. This algorithm will be used in our numerical experiments.

2.2. Convex analysis. We introduce here several definitions and results of convex analysis that will be used in this paper. Readers should refer to Hiriart–Urruty and Lemaréchal [60, 61] and Rockafellar [118] for comprehensive references on finite-dimensional convex analysis.

Definition 1. (Convex sets, relative interiors, and convex hulls) A set $C \subset \mathbb{R}^n$ is called convex if for any $\lambda \in [0,1]$ and any $x, y \in C$, the element $\lambda x + (1-\lambda)y$ is also in $C$. The relative interior of a convex set $C \subset \mathbb{R}^n$, denoted by $\text{ri } C$, consists of the points in the interior of the unique smallest affine set containing $C$. Every convex set $C \subset \mathbb{R}^n$ with non-empty interior is $n$-dimensional with $\text{ri } C = \text{int } C$. The convex hull
of a set $C$, denoted by $\text{conv } C$, consists of all the convex combinations of the elements of $C$. An important example of convex hull is the unit simplex in $\mathbb{R}^n$, $n \in \mathbb{N}$, denoted by
\[(3) \quad \Lambda_n := \left\{ (\alpha_1, \ldots, \alpha_n) \in [0,1]^n : \sum_{i=1}^n \alpha_i = 1 \right\}.
\]

**Definition 2.** (Domains and proper functions) The domain of a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is the set
$$\text{dom } f = \{ x \in \mathbb{R}^n : f(x) < +\infty \}.$$ A function $f$ is called proper if its domain is non-empty and $f(x) > -\infty$ for every $x \in \mathbb{R}^n$.

**Definition 3.** (Convex functions, lower semi-continuity, and convex envelopes) A proper function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called convex if the set $\text{dom } f$ is convex and if for any $x, y \in \text{dom } f$ and all $\lambda \in [0,1]$, there holds
\[(4) \quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \]
A proper function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called lower semi-continuous if for every sequence $\{x_k\}_{k=1}^{+\infty} \subseteq \mathbb{R}^n$ with $\lim_{k \to +\infty} x_k = x \in \mathbb{R}^n$, we have $\liminf_{k \to +\infty} f(x_k) \geq f(x)$.

The class of proper, lower semi-continuous convex functions is denoted by $\Gamma_0(\mathbb{R}^n)$. Given a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, we define its convex envelope $\text{co } f$ as the largest convex function such that $\text{co } f(x) \leq f(x)$ for every $x \in \mathbb{R}^n$. We define the convex lower-semicontinuous envelope $\overline{\text{co }} f$ as the largest convex and lower semi-continuous function such that $\overline{\text{co }} f(x) \leq f(x)$ for every $x \in \mathbb{R}^n$.

**Definition 4.** (Subdifferentials and subgradients) The subdifferential $\partial f(x)$ of $f \in \Gamma_0(\mathbb{R}^n)$ at $x \in \text{dom } f$ is the set (possibly empty) of vectors $s \in \mathbb{R}^n$ satisfying
\[(5) \quad \forall y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \langle s, y-x \rangle.
\]
The subdifferential $\partial f(x)$ is a closed convex set whenever it is non–empty, and any vector $s \in \partial f(x)$ is called a subgradient of $f$ at $x$. If $f$ is a proper convex function, then $\partial f(x) \neq \emptyset$ whenever $x \in \text{ri } (\text{dom } f)$, and $\partial f(x) = \emptyset$ whenever $x \notin \text{dom } f$.

If a convex function $f$ is differentiable at $x_0 \in \mathbb{R}^n$, then its gradient $\nabla_x f(x_0)$ is the unique subgradient of $f$ at $x_0$, and conversely if $f$ has a unique subgradient at $x_0$, then $f$ is differentiable at that point [118, Thm. 23.4].

**Definition 5.** (Legendre transforms) Let $f \in \Gamma_0(\mathbb{R}^n)$. The Legendre transform $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ of $f$ is defined as
\[(6) \quad f^*(s) = \sup_{x \in \mathbb{R}^n} \{ \langle s, x \rangle - f(x) \}.
\]
For any $f \in \Gamma_0(\mathbb{R}^n)$, the mapping $f \mapsto f^*$ is one-to-one, $f^* \in \Gamma_0(\mathbb{R}^n)$, and $(f^*)^* = f$. Moreover, for any $(x, s) \in \mathbb{R}^n \times \mathbb{R}^n$, the so-called Fenchel’s inequality holds:
\[(7) \quad f(x) + f^*(s) \geq \langle x, s \rangle,
\]
with equality attained if and only if $s \in \partial f(x)$, if and only if $x \in \partial f^*(s)$ [61, Cor. X.1.4.4].

We summarize some notation and definitions in Tab. 1.

3. Connections between neural networks and Hamilton–Jacobi equations

3.1. Set-up. In this section, we consider the function $f : \mathbb{R}^n \times [0, +\infty) \to \mathbb{R}$ given by the neural network in Fig. 1. Mathematically, the function $f$ can be expressed using the following formula
\[(8) \quad f(x, t; \{(p_i, \theta_i, \gamma_i)\}_{i=1}^m) = \max_{i \in \{1, \ldots, m\}} \{ (p_i, x) - t\theta_i - \gamma_i \}.
\]
Our goal is to show that the function $f$ in (8) is the unique uniformly continuous viscosity solution to a suitable Hamilton–Jacobi equation. In what follows we denote $f(x, t; \{(p_i, \theta_i, \gamma_i)\}_{i=1}^m)$ by $f(x, t)$ when there is no ambiguity in the parameters.

We adopt the following assumptions on the parameters:
(A1) The parameters $\{p_i\}_{i=1}^m$ are pairwise distinct, i.e., $p_i \neq p_j$ if $i \neq j$.
(A2) There exists a convex function $g : \mathbb{R}^n \to \mathbb{R}$ such that $g(p_i) = \gamma_i$. 

Table 1. Notation used in this paper. Here, we use $C$ to denote a set in $\mathbb{R}^n$, $f$ to denote a function from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ and $x$ to denote a vector in $\mathbb{R}^n$.

| Notation | Meaning | Definition |
|----------|---------|-----------|
| $\langle \cdot, \cdot \rangle$ | Euclidean scalar product in $\mathbb{R}^n$ | $\langle x, y \rangle := \sum_{i=1}^n x_i y_i$ |
| $\|\cdot\|_2$ | Euclidean norm in $\mathbb{R}^n$ | $\|x\|_2 := \sqrt{x^T x}$ |
| $\text{ri } C$ | Relative interior of $C$ | The interior of $C$ with respect to the minimal hyperplane containing $C$ in $\mathbb{R}^n$ |
| $\text{conv } C$ | Convex hull of $C$ | The set containing all convex combinations of the elements of $C$ |
| $\Lambda_n$ | Unit simplex in $\mathbb{R}^n$ | $\{(\alpha_1, \ldots, \alpha_n) \in [0,1]^n : \sum_{i=1}^n \alpha_i = 1\}$ |
| $\text{dom } f$ | Domain of $f$ | $\{x \in \mathbb{R}^n : f(x) < +\infty\}$ |
| $\Gamma_0(\mathbb{R}^n)$ | A useful and standard class of convex functions | The set containing all proper, convex, lower semi-continuous functions from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ |
| $\text{co } f$ | Convex envelope of $f$ | The largest convex function such that $f(x) \leq f(\bar{x})$ for every $\bar{x} \in \mathbb{R}^n$ |
| $\text{co } f$ | Convex and lower semi-continuous envelope of $f$ | The largest convex and lower semi-continuous function such that $\text{co } f(x) \leq f(x)$ for every $x \in \mathbb{R}^n$ |
| $\partial f(x)$ | Subdifferential of $f$ at $x$ | $\{p \in \mathbb{R}^n : f(y) \geq f(x) + \langle p, y-x \rangle \forall y \in \mathbb{R}^n\}$ |
| $f^*$ | Legendre transform of $f$ | $f^*(p) := \sup_{x \in \mathbb{R}^n} \{\langle p, x \rangle - f(x)\}$ |

(A3) For any $j \in \{1, \ldots, m\}$ and any $(\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ that satisfy

$$
\begin{cases}
(\alpha_1, \ldots, \alpha_m) \in \Lambda_m \text{ with } \alpha_j = 0, \\
\sum_{i \neq j} \alpha_i p_i = p_j, \\
\sum_{i \neq j} \alpha_i \gamma_i = \gamma_j,
\end{cases}
$$

there holds $\sum_{i \neq j} \alpha_i \theta_i > \theta_j$.

Note that (A3) is not a strong assumption. Indeed, if there exist $j \in \{1, \ldots, m\}$ and $(\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ satisfying Eq. (9) and $\sum_{i \neq j} \alpha_i \theta_i \leq \theta_j$, then there holds

$$
\langle p_j, x \rangle - t \theta_j - \gamma_j \leq \sum_{i \neq j} \alpha_i (\langle p_i, x \rangle - t \theta_i - \gamma_i) \leq \max_{i \neq j} \{\langle p_i, x \rangle - t \theta_i - \gamma_i\}.
$$

As a result, the $j$th neuron in the network can be removed without changing the value of $f(x, t)$ for any $x \in \mathbb{R}^n$ and $t \geq 0$. Removing all such neurons in the network, we can therefore assume (A3) holds.

Our aim is to identify the HJ equations whose viscosity solutions correspond to the neural network $(x, t) \mapsto f(x, t)$. Here, $x$ and $t$ play the role of the spatial and time variables, and $f(\cdot, 0)$ corresponds to the initial data. To simplify the notation, we define the function $J : \mathbb{R}^n \rightarrow \mathbb{R}$ as

$$
f(x, 0) = J(x) := \max_{i \in \{1, \ldots, m\}} \{\langle p_i, x \rangle - \gamma_i\}
$$

and the set $I_x$ as the collection of maximizers in Eq. (10) at $x$, that is,

$$
I_x := \arg \max_{i \in \{1, \ldots, m\}} \{\langle p_i, x \rangle - \gamma_i\}.
$$

The function $J$ satisfies several properties that we describe in the following lemma.

**Lemma 3.1.** Suppose $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$ and $\{\gamma_i\}_{i=1}^m \subset \mathbb{R}$ satisfy assumptions (A1) and (A2). Then the following statements hold.

(i) The Legendre transform of $J$ is given by the convex and lower-semicontinuous function

$$
J^*(p) = \begin{cases} 
\min_{(\alpha_1, \ldots, \alpha_m) \in \Lambda_m} \left\{ \sum_{i=1}^m \alpha_i \gamma_i \right\}, & \text{if } p \in \text{co } \{p_i\}_{i=1}^m, \\
+\infty, & \text{otherwise.}
\end{cases}
$$
Moreover, its restriction to \( \text{dom} \ J^* \) is continuous, and the subdifferential \( \partial J^*(p) \) is non-empty for every \( p \in \text{dom} \ J^* \).

(ii) Let \( p \in \text{dom} \ J^* \) and \( x \in \partial J^*(p) \). Then \((\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m\) is a minimizer in Eq. (12) if and only if it satisfies the constraints

(a) \((\alpha_1, \ldots, \alpha_m) \in \Lambda_m\),
(b) \[ \sum_{i=1}^m \alpha_i p_i = p, \]
(c) \( \alpha_i = 0 \) for any \( i \notin I_x \).

(iii) For any \( i, k \in \{1, \ldots, m\} \), let

\[ \alpha_i = \delta_{ik} := \begin{cases} 1, & \text{if } i = k, \\ 0, & \text{if } i \neq k. \end{cases} \]

Then \((\alpha_1, \ldots, \alpha_m)\) is a minimizer in Eq. (12) at the point \( p = p_k \). Hence, we have \( J^*(p_k) = \gamma_k \).

**Proof.** See Appendix A.1 for the proof. \( \square \)

Having defined the initial condition \( J \), the next step is to define a Hamiltonian \( H \). To do so, first denote by \( \mathcal{A}(p) \) the set of minimizers in Eq. (12) evaluated at \( p \in \text{dom} \ J^* \), i.e.,

\[
\mathcal{A}(p) := \arg \min_{(\alpha_1, \ldots, \alpha_m) \in \Lambda_m} \left\{ \sum_{i=1}^m \alpha_i \gamma_i \right\}.
\]
Note that the set $A(p)$ is non-empty for every $p \in \text{dom } J^*$ by Lem. 3.1(i). Now, we define the Hamiltonian function $H: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ by
\[
H(p) := \begin{cases} 
\inf_{\alpha \in A(p)} \{\sum_{i=1}^m \alpha_i \theta_i\}, & \text{if } p \in \text{dom } J^*, \\
+\infty, & \text{otherwise}.
\end{cases}
\]
The properties of $H$ are stated in the following lemma.

**Lemma 3.2.** Suppose $\{p_i\}_{i=1}^m \subset \mathbb{R}^n$, $\{\gamma_i\}_{i=1}^m \subset \mathbb{R}$, and $\{\theta_i\}_{i=1}^m \subset \mathbb{R}$ satisfy assumptions (A1)-(A3). Then the following statements hold.

(i) For every $p \in \text{dom } J^*$, the set $A(p)$ is compact and Eq. (14) has at least one minimizer.

(ii) The restriction of $H$ to $\text{dom } J^*$ is a bounded and continuous function.

(iii) There holds $H(p_i) = \theta_i$ for each $i = 1, \ldots, m$.

Proof. See Appendix A.2 for the proof. \qed

### 3.2. Main results: First-order Hamilton–Jacobi equations

Let $f$ be the function represented by the neural network architecture in Fig. 1 whose mathematical definition is given in Eq. (8). In the following theorem, we identify the HJ equations whose solution is given by $f$. Specifically, $f$ solves the HJ equation with Hamiltonian $H$ and initial function $J$ that were defined previously in Eqs. (14) and (10), respectively. Furthermore, we have a stronger statement. In fact, we provide the necessary and sufficient conditions for the HJ equation with solution $f$.

**Theorem 3.1.** Assume (A1)-(A3) hold. Let $f$ be the neural network defined by Eq. (8) with parameters $(\{p_i, \theta_i, \gamma_i\})_{i=1}^m$. Let $J$ and $H$ be the functions defined in Eqs. (10) and (14), respectively, and let $\bar{H}: \mathbb{R}^n \to \mathbb{R}$ be a continuous function. Then the following two statements hold.

(i) The neural network $f$ is the unique uniformly continuous viscosity solution to the first-order Hamilton–Jacobi equation
\[
\begin{cases} 
\frac{\partial f}{\partial t}(x, t) + H(x, t) = 0, & x \in \mathbb{R}^n, t > 0, \\
f(x, 0) = J(x), & x \in \mathbb{R}^n.
\end{cases}
\]

Moreover, $f$ is jointly convex in $(x, t)$.

(ii) The neural network $f$ is the unique uniformly continuous viscosity solution to the first-order Hamilton–Jacobi equation
\[
\begin{cases} 
\frac{\partial f}{\partial t}(x, t) + \bar{H}(x, t) = 0, & x \in \mathbb{R}^n, t > 0, \\
f(x, 0) = J(x), & x \in \mathbb{R}^n.
\end{cases}
\]

if and only if $\bar{H}(p_i) = H(p_i)$ for every $i = 1, \ldots, m$ and $\bar{H}(p) \geq H(p)$ for every $p \in \text{dom } J^*$.

**Remark.** This theorem identifies the set of HJ equations with initial data $J$ whose solution is given by the neural network $f$. To each such HJ equation, there corresponds a continuous Hamiltonian $\bar{H}$ satisfying $\bar{H}(p_i) = H(p_i)$ for every $i = 1, \ldots, m$ and $\bar{H}(p) \geq H(p)$ for every $p \in \text{dom } J^*$. The smallest possible Hamiltonian satisfying these constraints is the function $H$ defined in (14), and its corresponding HJ equation is given by (15).

Proof. To prove this theorem, we use the following three lemmas whose proofs are given in Appendix A.3, A.4, and A.5, respectively.

**Lemma 3.3.** Assume (A1)-(A3) hold. Let $J$ and $H$ be the functions defined in Eqs. (14) and (10), respectively. Let $\bar{H}: \mathbb{R}^n \to \mathbb{R}$ be a continuous function satisfying $\bar{H}(p_i) = H(p_i)$ for any $i = 1, \ldots, m$ and $\bar{H}(p) \geq H(p)$ for any $p \in \text{dom } J^*$. Then the neural network $f$ defined in Eq. (8) satisfies
\[
f(x, t) := \max_{i \in \{1, \ldots, m\}} \{\langle p_i, x \rangle - t\theta_i - \gamma_i\} = \sup_{p \in \text{dom } J^*} \left\{ \langle p, x \rangle - t\bar{H}(p) - J^*(p) \right\}.
\]

**Lemma 3.4.** Assume (A1)-(A3) hold. For any $k = 1, \ldots, m$, there exist $x \in \mathbb{R}^n$ and $t > 0$ such that $f(\cdot, t)$ is differentiable at $x$ and $\nabla_x f(x, t) = p_k$. 

\[\text{(17)}\]
Lemma 3.5. Assume (A1)-(A3) hold. Define a function $F: \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ by
\begin{equation}
F(p, E^-) := \begin{cases} J^*(p), & \text{if } E^- + H(p) \leq 0, \\
+\infty, & \text{otherwise.}
\end{cases}
\end{equation}
for any $p \in \mathbb{R}^n$ and $E^- \in \mathbb{R}$. Then the convex envelope of $F$ is given by
\begin{equation}
\co F(p, E^-) = \inf_{(c_1, \ldots, c_m) \in C(p, E^-)} \sum_{i=1}^{m} c_i \gamma_i,
\end{equation}
where the constraint set $C(p, E^-)$ is defined by $C(p, E^-) := \left\{(c_1, \ldots, c_m) \in \Lambda_m : \sum_{i=1}^{m} c_i p_i = p, \sum_{i=1}^{m} c_i \theta_i \leq -E^-\right\}$.

Proof of (i) in Thm. 3.1. First, the neural network $f$ is the pointwise maximum of $m$ affine functions in $(x, t)$ and therefore is jointly convex in these variables. Second, as the function $H$ is continuous and bounded in dom $J^*$ by Lem. 3.2(ii), there exists a continuous and bounded function defined in $\mathbb{R}^n$ whose restriction to dom $J^*$ coincides with $H$ [51 Thm. 4.16]. Then statement (i) follows by substituting this function for $\tilde{H}$ in statement (ii), and so it suffices to prove the latter.

Proof of (ii) in Thm. 3.1 (sufficiency): Suppose $\tilde{H}(p_i) = H(p_i)$ for every $i = 1, \ldots, m$ and $\tilde{H}(p) \geq H(p)$ for every $p \in \text{dom } J^*$. Since $H$ is continuous on $\mathbb{R}^n$ and $J$ is convex and Lipschitz continuous with Lipschitz constant $L = \max_i \{\|p_i\|\}$, [9 Thm. 3.1] implies that $(x, t) \mapsto \sup_{p \in \text{dom } J^*} \{\langle p, x \rangle - t \tilde{H}(p) - J^*(p)\}$ is the unique uniformly continuous viscosity solution to the HJ equation [16]. But this function is equivalent to the neural network $f$ by Lem. 3.3 and therefore both sufficiency and statement (i) follow.

Proof of (ii) in Thm. 3.1 (necessity): Suppose the neural network $f$ is the unique uniformly continuous viscosity solution to (16). First, we prove that $\tilde{H}(p_i) = H(p_i)$ for every $i = 1, \ldots, m$. Fix $k \in \{1, \ldots, m\}$. By Lem. 3.4 there exist $x \in \mathbb{R}^n$ and $t > 0$ satisfying $\partial_x f(x, t) = \{p_k\}$. Use Lems. 3.1(iii) and 3.2(iii) to write the maximization problem in Eq. (8) as
\begin{equation}
f(x, t) = \max_{p \in \{p_1, \ldots, p_m\}} \langle p, x \rangle - t H(p) - J^*(p),
\end{equation}
where $(p, t) \mapsto \langle p, x \rangle - t H(p) - J^*(p)$ is continuous in $(p, t)$ and differentiable in $t$. As the feasible set $\{p_1, \ldots, p_m\}$ is compact, $f$ is also differentiable with respect to $t$ [20 Prop. 4.12], and its derivative equals
\begin{equation}
\frac{\partial f}{\partial t}(x, t) = \min \{-H(p) : p \text{ is a maximizer in Eq. (20)}\}.
\end{equation}
Since $x$ and $t$ satisfy $\partial_x f(x, t) = \{p_k\}$, [60 Thm. VI.4.4.2] implies that the only maximizer in Eq. (20) is $p_k$. As a result, there holds
\begin{equation}
\frac{\partial f}{\partial t}(x, t) = -H(p_k).
\end{equation}
Since $f$ is convex on $\mathbb{R}^n$, its subdifferential $\partial f(x, t)$ is non-empty and satisfies
\begin{equation}
\partial f(x, t) \subseteq \partial_x f(x, t) \times \partial_t f(x, t) = \{(p_k, -H(p_k))\}.
\end{equation}
In other words, the subdifferential $\partial f(x, t)$ contains only one element, and therefore $f$ is differentiable at $(x, t)$ and its gradient equals $(p_k, -H(p_k))$ [118 Thm. 21.5]. Using (16) and (21), we obtain
\begin{equation}
0 = \frac{\partial f}{\partial t}(x, t) + \tilde{H}(\nabla_x f(x, t)) = -H(p_k) + \tilde{H}(p_k).
\end{equation}
As $k \in \{1, \ldots, m\}$ is arbitrary, we find that $H(p_k) = \tilde{H}(p_k)$ for every $k = 1, \ldots, m$.

Next, we prove by contradiction that $\tilde{H}(p) \geq H(p)$ for every $p \in \text{dom } J^*$. It is enough to prove the property only for every $p \in \text{ri dom } J^*$ by continuity of both $H$ and $\tilde{H}$ (where continuity of $H$ is proved in Lem. 3.2(ii)). Assume $\tilde{H}(p) < H(p)$ for some $p \in \text{ri dom } J^*$. Define two functions $F$ and $\tilde{F}$ from $\mathbb{R}^n \times \mathbb{R}$ to $\mathbb{R} \cup \{+\infty\}$ by
\begin{equation}
F(q, E^-) := \begin{cases} J^*(q), & \text{if } E^- + H(q) \leq 0, \\
+\infty, & \text{otherwise.}
\end{cases}
\end{equation}
and
\begin{equation}
\tilde{F}(q, E^-) := \begin{cases} J^*(q), & \text{if } E^- + \tilde{H}(q) \leq 0, \\
+\infty, & \text{otherwise.}
\end{cases}
\end{equation}
for any \( q \in \mathbb{R}^n \) and \( E^- \in \mathbb{R} \). Denoting the convex envelope of \( F \) by \( \text{co } F \), Lem. 3.5 implies

\[
\text{co } F(q, E^-) = \inf_{c_1, ..., c_m \in C(q, E^-)} \sum_{i=1}^m c_i \gamma_i, \quad \text{where } C \text{ is defined by}
\]

\[
C(q, E^-) := \left\{ (c_1, ..., c_m) \in \Lambda_m : \sum_{i=1}^m c_i p_i = q, \sum_{i=1}^m c_i \theta_i \leq -E^- \right\}.
\]

Let \( E^-_1 \in \left(-H(p), -\tilde{H}(p)\right) \). Now, we want to prove that \( \text{co } F(p, E^-_1) \leq J^*(p) \); this inequality will lead to a contradiction with the definition of \( H \).

Using statement (i) of this theorem and the supposition that \( f \) is the unique viscosity solution to the HJ equation [16], we have that

\[
f(x, t) = \sup_{q \in \mathbb{R}^n} \{ \langle q, x \rangle - tH(q) - J^*(q) \} = \sup_{q \in \mathbb{R}^n} \{ \langle q, x \rangle - t\tilde{H}(q) - J^*(q) \}.
\]

Furthermore, a similar calculation as in the proof of [35, Prop. 3.1] yields

\[
f = F^* = \tilde{F}^*, \quad \text{which implies } f^* = \text{co } F = \text{co } \tilde{F}.
\]

where \( \text{co } F \) and \( \text{co } \tilde{F} \) denotes the convex lower semi-continuous envelopes of \( F \) and \( \tilde{F} \), respectively. On the one hand, since \( f^* = \text{co } \tilde{F} \), the definition of \( \tilde{F} \) in Eq. (22) implies

\[
f^* \left(p, -\tilde{H}(p)\right) \leq \tilde{F} \left(p, -\tilde{H}(p)\right) = J^*(p) \quad \text{and} \quad \left(p, -\tilde{H}(p)\right) \subseteq \text{dom } \tilde{F} \subseteq \text{dom } f^*.
\]

Recall that \( p \in \text{ri dom } J^* \) and \( E^-_1 \leq -\tilde{H}(p) \), so that \( (p, E^-_1) \in \text{ri dom } f^* \). As a result, we get

\[
(p, \alpha E^-_1 + (1 - \alpha)(-\tilde{H}(p))) \in \text{ri dom } f^* \text{ for all } \alpha \in (0, 1).
\]

On the other hand, since \( f^* = \text{co } F \), we have \( \text{ri dom } f^* = \text{ri dom } (\text{co } F) \) and \( f^* = \text{co } F \) in \( \text{ri dom } f^* \). Taken together with Eq. (25) and the continuity of \( f^* \), there holds

\[
f^* \left(p, -\tilde{H}(p)\right) = \lim_{\alpha \to 0 \atop 0 < \alpha < 1} f^* \left(p, \alpha E^-_1 + (1 - \alpha)(-\tilde{H}(p))\right)
\]

\[
= \lim_{\alpha \to 0 \atop 0 < \alpha < 1} \text{co } F \left(p, \alpha E^-_1 + (1 - \alpha)(-\tilde{H}(p))\right).
\]

Note that \( \text{co } F(p, \cdot) \) is monotone non-decreasing. Indeed, if \( E^-_2 \) is a real number such that \( E^-_2 > E^-_1 \), by the definition of the set \( C \) in Eq. (23) there holds \( C(p, E^-_2) \subseteq C(p, E^-_1) \), which implies \( \text{co } F(p, E^-_2) \geq \text{co } F(p, E^-_1) \). Recalling that \( E^-_1 \leq -\tilde{H}(p) \), monotonicity of \( \text{co } F(p, \cdot) \) and Eq. (26) imply

\[
f^* \left(p, -\tilde{H}(p)\right) \geq \lim_{\alpha \to 0 \atop 0 < \alpha < 1} \text{co } F \left(p, \alpha E^-_1 + (1 - \alpha)E^-_1\right) = \text{co } F(p, E^-_1).
\]

Combining Eqs. (24) and (27), we get

\[
\text{co } F(p, E^-_1) \leq J^*(p) < +\infty.
\]

As a result, the set \( C(p, E^-_1) \) is non-empty. Since it is also compact, there exists a minimizer in Eq. (23) evaluated at the point \( (p, E^-_1) \). Let \( (c_1, ..., c_m) \) be such a minimizer. By Eqs. (23) and (28) and the assumption that \( E^-_1 \in \left(-H(p), -\tilde{H}(p)\right) \), there holds

\[
\begin{cases}
(c_1, ..., c_m) \in \Lambda_m, \\
\sum_{i=1}^m c_i p_i = p, \\
\sum_{i=1}^m c_i \gamma_i = \text{co } F(p, E^-_1) \leq J^*(p), \\
\sum_{i=1}^m c_i \theta_i \leq -E^-_1 < H(p).
\end{cases}
\]

Comparing the first three statements in Eq. (29) and the formula of \( J^* \) in Eq. (12), we deduce that \( (c_1, ..., c_m) \) is a minimizer in Eq. (12), i.e., \( (c_1, ..., c_m) \in A(p) \). By definition of \( H \) in Eq. (14), we have

\[
H(p) = \inf_{\alpha \in A(p)} \sum_{i=1}^m \alpha_i \theta_i \leq \sum_{i=1}^m c_i \theta_i,
\]
which contradicts the last inequality in Eq. (29). Therefore, we conclude that $\tilde{H}(p) \geq H(p)$ for any $p \in \text{ri dom } J^*$ and the proof is finished.

Figure 2. Illustration of the structure of the neural network (31) that can represent the entropy solution to one-dimensional conservation laws.

3.3. First-order one-dimensional conservation laws. It is well-known that one-dimensional conservation laws are related to HJ equations (see, e.g., [1, 25, 28, 74, 78, 82, 91], and also [53] for a comprehensive introduction to conservation laws and entropy solutions). Formally, by taking spatial gradient of the HJ equation (1) (with $\epsilon = 0$) and identifying the gradient $\nabla_x f \equiv u$, we obtain the conservation law

$$\begin{aligned}
\frac{\partial u}{\partial t}(x,t) + \nabla_x H(u(x,t)) &= 0, \quad x \in \mathbb{R}, t > 0, \\
u(x,0) &= u_0(x) := \nabla J(x), \quad x \in \mathbb{R},
\end{aligned}$$

(30)

where the flux function corresponds to the Hamiltonian $H$ in the HJ equation. Here, we assume that the initial data $J$ is convex and globally Lipschitz continuous, and the symbols $\nabla$ and $\nabla_x$ in this section correspond to derivatives in the sense of distribution if the classical derivatives do not exist.

In this section, we show that the conservation law derived from the HJ equation (1) (with $\epsilon = 0$) can be represented by a neural network architecture. Specifically, the corresponding entropy solution $u(x,t) \equiv \nabla_x f(x,t)$ to the one-dimensional conservation law (30) can be represented using a neural network architecture with an argmax based activation function, i.e.,

$$\nabla_x f(x,t) = p_j, \quad \text{where } j \in \arg\max_{i \in \{1, \ldots, m\}} \{p_i, x - t\theta_i - \gamma_i\},$$

(31)
The structure of this network is shown in Fig. 2. When more than one maximizer exist in the optimization problem above, one can choose any maximizer \( j \) and define the value to be \( p_j \). We now prove that the function \( \nabla_x f \) given by the neural network (31) is indeed the entropy solution to the one-dimensional conservation law (30) with flux function \( H \) and initial data \( \nabla J \), where \( H \) and \( J \) defined as per Eqs. (14) and (10), respectively.

**Proposition 3.1.** Consider the one-dimensional case, i.e., \( n = 1 \), and assume (A1)-(A3) hold. Let \( u := \nabla_x f \) be the neural network defined in Eq. (31) with parameters \( \{(p_i, \theta_i, \gamma_i)\}_{i=1}^{m} \). Let \( J \) and \( H \) be the functions defined in Eqs. (10) and (14), respectively, and let \( \tilde{H} : \mathbb{R} \to \mathbb{R} \) be a locally Lipschitz continuous function. Then the following two statements hold.

(i) The neural network \( u \) is the entropy solution to the conservation law

\[
\begin{cases}
\frac{\partial u}{\partial t} + \nabla_x H(u(x,t)) = 0, & x \in \mathbb{R}, t > 0, \\
u(x,0) = \nabla J(x), & x \in \mathbb{R}.
\end{cases}
\]

(ii) The neural network \( u \) is the entropy solution to the conservation law

\[
\begin{cases}
\frac{\partial u}{\partial t} + \nabla_x \tilde{H}(u(x,t)) = 0, & x \in \mathbb{R}, t > 0, \\
u(x,0) = \nabla J(x), & x \in \mathbb{R},
\end{cases}
\]

if and only if there exists a constant \( C \in \mathbb{R} \) such that \( \tilde{H}(p_i) = H(p_i) + C \) for every \( i \in \{1, \ldots, m\} \) and \( \tilde{H}(p) \geq H(p) + C \) for any \( p \in \text{conv} \{p_i\}_{i=1}^{m} \).

**Proof.** See Appendix B for the proof. \( \square \)

### 3.4. Second-order Hamilton–Jacobi equations

In this section, we show that when the maximum activation function of the neural network (8) is replaced by a smooth log-exponential function, then under certain conditions the resulting neural network solves a second-order HJ PDE. Specifically, let \( \epsilon > 0 \) and define the function \( f_\epsilon : \mathbb{R}^n \times [0, +\infty) \to \mathbb{R} \) by

\[
f_\epsilon(x,t) := \epsilon \log \left( \sum_{i=1}^{m} e^{\langle (p_i, x) - \theta_i, \gamma_i \rangle / \epsilon} \right).
\]

This neural network \( f_\epsilon \), which we illustrate in Fig. 3 approximates the neural network \( f \) in that it satisfies the bounds

\[
f(x,t) \leq f_\epsilon(x,t) \leq f(x,t) + \epsilon \log(m)
\]

and the limit \( \lim_{\epsilon \to 0} f_\epsilon(x,t) = f(x,t) \). We now show that under the assumption that the parameter \( \theta_i = -\frac{1}{2} \| p_i \|_2^2 \), the neural network (34) solves a second-order HJ PDE.

**Proposition 3.2.** Let \( f_\epsilon \) be the neural network defined by Eq. (34) with parameters \( \{(p_i, \theta_i, \gamma_i)\}_{i=1}^{m} \) and \( \epsilon > 0 \). In addition, let \( \theta_i = -\frac{1}{2} \| p_i \|_2^2 \) for \( i \in \{1, \ldots, m\} \). Then the neural network \( f_\epsilon \) is the unique smooth solution to the second-order Hamilton–Jacobi equation

\[
\begin{cases}
\frac{\partial f_\epsilon(x,t)}{\partial t} - \frac{1}{2} \| \nabla_x f_\epsilon(x,t) \|_2^2 = \frac{1}{2} \Delta_x f_\epsilon(x,t) & \text{in } \mathbb{R}^n \times (0, +\infty), \\
f_\epsilon(x,0) = \epsilon \log \left( \sum_{i=1}^{m} e^{\langle (p_i, x) - \gamma_i \rangle / \epsilon} \right) & \forall x \in \mathbb{R}^n.
\end{cases}
\]

Moreover, \( f_\epsilon \) is jointly convex in \((x,t)\), \( f_\epsilon \) satisfies the bounds (35), and \( f_\epsilon \) satisfies the limit

\[
\lim_{\epsilon \to 0} f_\epsilon(x,t) = \max_{i \in \{1, \ldots, m\}} \{ (p_i, x) + \left( \frac{1}{2} \| p_i \|_2^2 - \gamma_i \right) \}
\]

for every \( x \in \mathbb{R}^n \) and \( t \geq 0 \). Finally, if assumptions (A1)-(A3) hold, the right hand side of (37) solves the first-order Hamilton–Jacobi equation (16) with \( \tilde{H} := -\frac{1}{2} \| \cdot \|_2^2 \).

**Proof.** See Appendix C for the proof. \( \square \)

### 4. Numerical Experiments

In this section, we present some numerical experiments that show the representability of our three proposed neural network architectures each corresponding to first-order HJ equations, one-dimensional conservation laws, and a subclass of second-order HJ equations.
4.1. **First-order Hamilton–Jacobi equations.** Here, we present several numerical experiments for recovering information on the initial data of first-order HJ equations using the neural network architecture in Fig. 1 and machine learning techniques. We focus on the following inverse problem: We are given data samples from a function $S: \mathbb{R}^n \times [0, +\infty) \rightarrow \mathbb{R}$ that is the viscosity solution to an HJ equation (1) where $\epsilon = 0$ with unknown initial data $J$ and Hamiltonian $H$. Our aim is to recover the initial data $J$. We propose to learn the neural network using machine learning techniques to recover the initial data $J$. We shall see that this approach also provides partial information on the Hamiltonian $H$.

Specifically, given data samples $\{(x_j, t_j, S(x_j, t_j))\}_{j=1}^N$, where $\{(x_j, t_j)\}_{j=1}^N \subset \mathbb{R}^n \times [0, +\infty)$, we train the neural network $f$ with structure in Fig. 1 using the mean square loss function defined by

$$l(\{(p_i, \theta_i, \gamma_i)\}_{i=1}^m) = \frac{1}{N} \sum_{j=1}^N |f(x_j, t_j; \{(p_i, \theta_i, \gamma_i)\}_{i=1}^m) - S(x_j, t_j)|^2.$$

The training problem is formulated as

$$\arg\min_{\{(p_i, \theta_i, \gamma_i)\}_{i=1}^m \subset \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}} l(\{(p_i, \theta_i, \gamma_i)\}_{i=1}^m).$$

After training, we approximate the initial condition in the HJ equation, denoted by $\tilde{J}$, by evaluating the trained neural network at $t = 0$. That is, we approximate the initial condition by

$$\tilde{J} := f(\cdot, 0).$$
### Table 2. Relative mean square errors of the parameters in the neural network \( f \) with 2 neurons in different cases and different dimensions averaged over 100 repeated experiments.

| # Case | Case 1     | Case 2     | Case 3     | Case 4     |
|--------|------------|------------|------------|------------|
|        | 2D         | 4D         | 8D         | 16D        |
| Averaged | 4.10E-03   | 1.41E-09   | 1.14E-09   | 1.14E-09   |
| Relative | 2.10E-03   | 1.20E-09   | 1.03E-09   | 6.68E-03   |
| Errors of \( \{ p_i \} \) | 3.84E-03   | 1.38E-09   | 1.09E-09   | 1.23E-09   |
|        | 2.82E-03   | 1.29E-09   | 1.20E-09   | 7.74E-03   |
| Averaged | 4.82E-02   | 3.47E-10   | 1.47E-10   | 5.44E-11   |
| Relative | 7.31E-02   | 2.82E-10   | 1.08E-10   | 1.09E-10   |
| Errors of \( \{ \theta_i \} \) | 1.17E-01   | 1.15E-09   | 2.10E-09   | 4.75E-11   |
|        | 1.79E-01   | 1.15E-09   | 1.29E-09   | 4.12E-03   |
| Averaged | 1.35E-02   | 3.71E-10   | 2.91E-10   | 2.80E-10   |
| Relative | 1.01E-01   | 1.24E-09   | 1.74E-10   | 3.27E-01   |
| Errors of \( \{ \gamma_i \} \) | 1.33E-02   | 3.67E-10   | 2.82E-10   | 1.56E-01   |
|        | 9.24E-02   | 1.10E-09   | 2.25E-03   | 3.62E-02   |

### Table 3. Relative mean square errors of the parameters in the neural network \( f \) with 4 neurons in different cases and different dimensions averaged over 100 repeated experiments.

| # Case | Case 1     | Case 2     | Case 3     | Case 4     |
|--------|------------|------------|------------|------------|
|        | 2D         | 4D         | 8D         | 16D        |
| Averaged | 3.12E-01   | 7.82E-02   | 2.62E-02   | 2.88E-02   |
| Relative | 2.21E-01   | 6.12E-02   | 4.31E-03   | 3.64E-02   |
| Errors of \( \{ p_i \} \) | 2.85E-01   | 7.92E-02   | 4.02E-02   | 4.35E-02   |
|        | 2.14E-01   | 4.30E-02   | 7.82E-03   | 1.73E-02   |
| Averaged | 2.59E-01   | 6.07E-02   | 1.04E-02   | 2.66E-03   |
| Relative | 3.68E-01   | 8.37E-02   | 8.48E-03   | 7.80E-03   |
| Errors of \( \{ \theta_i \} \) | 4.82E-01   | 9.47E-02   | 1.41E-02   | 1.90E-02   |
|        | 2.59E-00   | 1.23E-01   | 1.31E-02   | 3.66E-01   |
| Averaged | 1.01E-02   | 6.07E-02   | 1.04E-02   | 8.09E-04   |
| Relative | 3.19E-01   | 9.47E-02   | 8.48E-03   | 1.81E-02   |
| Errors of \( \{ \gamma_i \} \) | 1.51E-02   | 1.23E-01   | 1.90E-02   | 3.66E-01   |
|        | 2.65E-01   | 1.30E-02   | 1.94E-03   | 1.17E-01   |

In addition, we obtain partial information of the Hamiltonian \( H \) using the parameters in the trained neural network via the following procedure. We first detect the effective neurons of the network, which we define to be the affine functions \( \{ \langle p_i, x \rangle - t \theta_i - \gamma_i \} \) that contribute to the pointwise maximum in the neural network \( f \) (see Eq. (8)). We then denote by \( L \) the set of indices that correspond to the parameters of the effective neurons, i.e.,

\[
L := \bigcup_{x \in \mathbb{R}^n, t \geq 0} \arg \max_{i \in \{1, \ldots, m\}} \{ \langle p_i, x \rangle - t \theta_i - \gamma_i \},
\]

and we finally use each effective parameter \( (p_l, \theta_l) \) for \( l \in L \) to approximate the point \( (p_l, H(p_l)) \) on the graph of the Hamiltonian. In practice, we approximate the set \( L \) using a large number of points \((x, t)\) sampled in the domain \( \mathbb{R}^n \times [0, +\infty) \).
4.1.1. Randomly generalized piecewise affine $H$ and $J$. In this subsection, we randomly select $m$ parameters $p^i_{true}$ in $[-1,1]^n$ for $i \in \{1, \ldots, m\}$, and define $\theta^i_{true}$ and $\gamma^i_{true}$ as follows

Case 1. $\theta^i_{true} = -\frac{1}{2}||p^i_{true}||_2$ and $\gamma^i_{true} = 0$, for $i \in \{1, \ldots, m\}$.
Case 2. $\theta^i_{true} = -\frac{1}{2}||p^i_{true}||_2$ and $\gamma^i_{true} = \frac{1}{2}||p^i_{true}||_2$, for $i \in \{1, \ldots, m\}$.
Case 3. $\theta^i_{true} = -\frac{1}{2}||p^i_{true}||_2$ and $\gamma^i_{true} = 0$, for $i \in \{1, \ldots, m\}$.
Case 4. $\theta^i_{true} = -\frac{1}{2}||p^i_{true}||_2$ and $\gamma^i_{true} = \frac{1}{2}||p^i_{true}||_2$, for $i \in \{1, \ldots, m\}$.

Define the function $S$ as

$$S(x,t) := \max_{i \in \{1, \ldots, m\}} \{\langle p^i_{true}, x \rangle - t\theta^i_{true} - \gamma^i_{true}\}.$$ 

By Thm. 3.1, this function $S$ is a viscosity solution to the HJ equations whose Hamiltonian and initial function are the piecewise affine functions defined in Eqs. (14) and (10), respectively. In other words, $S$ solves the HJ equation with initial data $J$ satisfying

$$J(x) := \max_{i \in \{1, \ldots, m\}} \{\langle p^i_{true}, x \rangle\}, \quad \text{for Case 1 and 3;}$$

$$J(x) := \max_{i \in \{1, \ldots, m\}} \{\langle p^i_{true}, x \rangle - \frac{1}{2}||p^i_{true}||^2\}, \quad \text{for Case 2 and 4},$$

and Hamiltonian $H$ satisfying

$$H(p) := \begin{cases} -\max_{\alpha \in \mathcal{A}(p)} \{\sum_{i=1}^m \alpha_i ||p^i_{true}||_2\}, & \text{if } p \in \text{dom } J^*, \text{ for Case 1 and 2;} \\ +\infty, & \text{otherwise}, \end{cases}$$

$$H(p) := \begin{cases} -\frac{1}{2} \max_{\alpha \in \mathcal{A}(p)} \{\sum_{i=1}^m \alpha_i ||p^i_{true}||_2^2\}, & \text{if } p \in \text{dom } J^*, \text{ for Case 3 and 4;} \\ +\infty, & \text{otherwise}, \end{cases}$$

where $\mathcal{A}(p)$ is the set of maximizers of the corresponding maximization problem in Eq. (10). Specifically, if we construct a neural network $f$ as shown in Fig. 1 with the underlying parameters $\{(p^i_{true}, \theta^i_{true}, \gamma^i_{true})\}_{i=1}^m$, then the function given by the neural network is exactly the same as the function $S$. In other words, $\{(p^i_{true}, \theta^i_{true}, \gamma^i_{true})\}_{i=1}^m$ is a global minimizer for the training problem (38) with the global minimal loss value equal to zero.

Now, we train the neural network $f$ with training data $\{(x_j, t_j, S(x_j, t_j))\}_{j=1}^N$, where the points $\{(x_j, t_j)\}_{j=1}^N$ are randomly sampled in $\mathbb{R}^n \times [0, +\infty)$ with respect to the standard normal distribution for each $j \in \{1, \ldots, N\}$ (we take the absolute value for $t$ to make sure it is non-negative). Here and after, the number of training data points is $N = 20,000$. We run 60,000 descent steps using the Adam optimizer to train the neural network $f$. The parameters for the Adam algorithm are chosen to be $\beta_1 = 0.5$, $\beta_2 = 0.9$, the learning rate is $10^{-4}$ and the batch size is 500.

To measure the performance of the training process, we compute the relative mean square errors of the sorted parameters in the trained neural network, denoted by $\{(p_i, \theta_i, \gamma_i)\}_{i=1}^m$, and the sorted underlying true parameters $\{(p^i_{true}, \theta^i_{true}, \gamma^i_{true})\}_{i=1}^m$. To be specific, the errors are computed as follows

$$\text{relative mean square error of } \{p_i\} = \frac{\sum_{i=1}^m ||p_i - p^i_{true}||^2_2}{\sum_{i=1}^m ||p^i_{true}||^2_2},$$

$$\text{relative mean square error of } \{\theta_i\} = \frac{\sum_{i=1}^m ||\theta_i - \theta^i_{true}||^2}{\sum_{i=1}^m ||p^i_{true}||^2_2},$$

$$\text{relative mean square error of } \{\gamma_i\} = \frac{\sum_{i=1}^m ||\gamma_i - \gamma^i_{true}||^2}{\sum_{i=1}^m ||p^i_{true}||^2_2}.$$ 

For the cases when the denominator $\sum_{i=1}^m ||\gamma^i_{true}||^2$ is zero, such as Case 1 and Case 3, we measure the absolute mean square error $\frac{1}{m} \sum_{i=1}^m ||\gamma_i - \gamma^i_{true}||^2$ instead.

We test Cases 1–4 on the neural networks with 2 and 4 neurons, i.e., we set $m = 2, 4$ and repeat the experiments 100 times. We then compute the relative mean square errors in each experiments and take the average. The averaged relative mean square errors are shown in Tabs. 2 and 3 respectively. From the error tables, we observe that the training process performs pretty well and gives errors below $10^{-8}$ in some cases when $m = 2$. However, for the case when $m = 4$, we do not obtain the global minimizers and the error is above $10^{-3}$. Therefore, there is no guarantee for the performance of Adam in this training problem and it may be related to the complexity of the solution $S$ to the underlying HJ equation.
4.1.2. Quadratic Hamiltonians. In this subsection, we consider two inverse problems of first-order HJ equations whose Hamiltonians and initial data are defined as follows

1. \( H(p) = -\frac{1}{2} \| p \|_2^2 \) and \( J(x) = \| x \|_1 \) for \( p, x \in \mathbb{R}^n \).
2. \( H(p) = \frac{1}{2} \| p \|_2^2 \) and \( J(x) = \| x \|_1 \) for \( p, x \in \mathbb{R}^n \).

The solution to each of the two corresponding HJ equations can be represented using the Hopf formula and reads

1. \( S(x, t) = \| x \|_1 + \frac{at}{2} \) for \( x \in \mathbb{R}^n \) and \( t \geq 0 \).
2. \( S(x, t) = \sum_{i: |x_i| \geq \frac{t}{2}} (|x_i| - \frac{t}{2}) + \sum_{i: |x_i| < \frac{t}{2}} \frac{x_i^2}{2t^2} \), where \( x = (x_1, \ldots, x_n) \in \mathbb{R}^n \) and \( t \geq 0 \).

We train the neural network \( f \) using the same procedure as in the previous subsection and obtain the function \( J \) (see Eq. (39)) and the parameters \( \{(p_i, \theta_i)\}_{i \in L} \) associated to the effective neurons. We compute the relative mean square error of \( J \) and \( \{(p_i, \theta_i)\}_{i \in L} \) as follows

\[
\text{relative error of } J := \frac{\sum_{j=1}^{N_{test}} |\tilde{J}(x_{i_{test}}^j) - J(x_{i_{test}}^j)|^2}{\sum_{j=1}^{N_{test}} |J(x_{i_{test}}^j)|^2},
\]

\[
\text{relative error of } \{(p_i, \theta_i)\} := \frac{\sum_{l \in L} |\theta_l - H(p_l)|^2}{\sum_{l \in L} |H(p_l)|^2},
\]

where \( \{x_{i_{test}}\} \) are randomly sampled with respect to the standard normal distribution in \( \mathbb{R}^n \) and there are in total \( N_{test} = 2,000 \) testing data points. We repeat the experiments 100 times. The corresponding averaged errors in the two examples are listed in Tabs. 4 and 5 respectively.

In the first example, we have \( H(p) = -\frac{1}{2} \| p \|_2^2 \) and \( J(x) = \| x \|_1 \). According to Thm. 3.1 the solution \( S \) can be represented without error by the neural network in Fig. 4 with parameters

\[
\left\{ (p, \theta, \gamma) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} : p(i) \in \{\pm 1\}, \text{ for } i \in \{1, \ldots, n\}, \theta = \frac{n}{2}, \gamma = 0 \right\},
\]

where \( p(i) \) denotes the \( i \)th entry of the vector \( p \). In other words, the global minimal loss value in the training problem is theoretically guaranteed to be zero. From the numerical errors in Tab. 4 we observe that in low dimension such as 1D and 2D, the errors of the initial function are small. However, in most cases, the errors of the parameters are pretty large. In the case of \( n \) dimension, the viscosity solution can be represented using the \( 2^n \) parameters in Eq. (41). However, the number of effective neurons are larger than \( 2^n \) in all cases, which also implies that the Adam optimizer does not find the global minimizers in this example.
Table 5. Relative mean square errors of \( \tilde{J} \) and \( \{ (p_l, \theta_l) \} \) for the inverse problems of the first-order HJ equations in different dimensions with \( J = \| \cdot \|_1 \) and \( H = \| \cdot \|_2^2/2 \), averaged over 100 repeated experiments.

| # Neurons | 64    | 128    | 256    | 512    | 1024   |
|-----------|-------|--------|--------|--------|--------|
| Averaged  |       |        |        |        |        |
| 1D        | 5.23E-08 | 2.45E-08 | 1.96E-08 | 1.77E-08 | 1.77E-08 |
| 2D        | 1.75E-05 | 1.67E-05 | 1.77E-05 | 1.85E-05 | 1.91E-05 |
| 4D        | 5.82E-04 | 4.94E-04 | 5.28E-04 | 5.76E-04 | 6.16E-04 |
| 8D        | 1.54E-02 | 1.40E-02 | 1.35E-02 | 1.33E-02 | 1.32E-02 |
| 16D       | 4.19E-02 | 4.33E-02 | 4.43E-02 | 4.46E-02 | 4.49E-02 |

| Averaged  |       |        |        |        |        |
| 1D        | 3.25E-02 | 1.93E-02 | 1.24E-02 | 5.62E-03 | 2.92E-03 |
| 2D        | 8.30E-03 | 7.08E-03 | 5.78E-03 | 4.25E-03 | 3.47E-03 |
| 4D        | 2.41E-02 | 2.41E-02 | 2.51E-02 | 2.65E-02 | 2.82E-02 |
| 8D        | 7.33E-02 | 7.32E-02 | 7.25E-02 | 7.15E-02 | 7.08E-02 |
| 16D       | 3.85E-01 | 3.90E-01 | 3.92E-01 | 3.92E-01 | 3.91E-01 |

| Averaged  |       |        |        |        |        |
| 1D        | 20.26 | 26.94 | 32.26 | 36.02 | 38.61 |
| 2D        | 32.74 | 48.05 | 65.7  | 84.87 | 99.83 |
| 4D        | 46.69 | 72.3  | 103.71 | 147.41 | 198.27 |
| 8D        | 55.55 | 82.04 | 95.46 | 90.82 | 82.5  |
| 16D       | 61.51 | 99.63 | 119.95 | 118.89 | 109.1 |

In the second example, the solution \( S \) cannot be represented using our proposed neural network without error. Hence the results describes the approximation of the solution \( S \) by the neural network. From Tab. 5, we observe that the errors become larger when the dimension increases. For this example, the number of effective neurons should be \( m \) where \( m \) is the number of neurons used in the architecture. Tab. 5 shows that the average number of effective neurons is below this optimal number. Therefore, this implies that the Adam optimizer does not find the global minimizers in this example either.

In conclusion, these numerical experiments suggest that recovering initial data from data samples using our proposed neural network architecture with the Adam optimizer is unsatisfactory for solving these inverse problems. In particular, Adam optimizer is not always able to find a global minimizer when the solution can be represented without error using our network architecture.

4.2. One-dimensional conservation laws. In this part, we show the representability of the neural network \( \nabla_x f \) given in Fig. 2 and Eq. (31). Since the number of neurons is finite, the function \( \nabla_x f \) only takes values in the finite set \( \{ p_i \}_{i=1}^m \). In other words, it can represent the entropy solution \( u \) to the PDE (30) without error only if \( u \) takes values in a finite set.

Here, we consider the following two examples

1. \( H(p) = -\frac{1}{2} p^2 \) and \( J(x) = |x| \) for \( p, x \in \mathbb{R} \). The initial condition \( u_0 \) is then given by
   \[
   u_0(x) = \begin{cases} 
   -1, & x \leq 0, \\
   1, & x > 0.
   \end{cases}
   \]

2. \( H(p) = \frac{1}{2} p^2 \) and \( J(x) = |x| \) for \( p, x \in \mathbb{R} \). Hence, the initial function \( u_0 \) is the same as in example 1.

In the first example, the entropy solution \( u \) only takes values in the finite set \( \{ \pm 1 \} \), and it can be represented by the neural network \( \nabla_x f \) without error by Prop. 3.1. However, in the second example, the solution \( u \) takes values in the infinite set \( [-1, 1] \), hence the neural network \( \nabla_x f \) is only an approximation of the corresponding solution \( u \).

To show the representability of the neural network, in each example, we choose the parameters \( \{ p_i \}_{i=1}^m \) to be the uniform grid points in \( [-1, 1] \), i.e.,
\[
p_i = -1 + \frac{2(i - 1)}{m - 1}, \quad \text{for } i \in \{1, \ldots, m\}.
\]
Figure 4. Plot of the function represented by the neural network $\nabla_x f$ at time $t = 1$ with 64 neurons whose parameters are defined using $H$ and $J^*$ in example 1. The function given by the neural network is plotted in orange and the true solution is plotted in blue.

Figure 5. Plot of the function represented by the neural network $\nabla_x f$ at time $t = 1$ with 32 and 128 neurons whose parameters are defined using $H$ and $J^*$ in example 2. The function given by the neural network is plotted in orange and the true solution is plotted in blue. The neural network with 32 neurons is shown on the left, while the neural network with 128 neurons is shown on the right.

We set $\theta_i = H(p_i)$ and $\gamma_i = J^*(p_i)$ for each $i \in \{1, \ldots, m\}$, where $J^*$ is the Legendre transform of the anti-derivative of the initial function $u_0$. Hence, in these two examples, $\gamma_i$ equals for each $i$. Figs. 4 and 5 show the neural network $\nabla_x f$ and the true entropy solution $u$ in these two examples at time $t = 1$. As expected, the error in Fig. 4 for example 1 is negligible. For example 2, we consider neural networks with 32 and 128 neurons whose graphs are plotted in Figs. 5a and 5b, respectively. We observe in these figures that the error of the neural networks with the specific parameters decreases as the number of neurons increases.

In conclusion, the neural network $\nabla_x f$ with the architecture in Fig. 2 can represent the solution to the one-dimensional conservation laws given in Eq. (30) pretty well. In fact, because of the discontinuity of the
activation function, the proposed neural network $\nabla_x f$ has advantages in representing the discontinuity in solution such as shocks, but it requires more neurons when approximating non-constant smooth parts of the solution.

4.3. Second-order Hamilton–Jacobi equations. In this part, we consider the inverse problem involving the second-order HJ equations (36) using the neural network architecture in Fig. 3 with $\theta_i = -\|p_i\|^2/2$ for each $i \in \{1, \ldots, m\}$. To be specific, the true parameters $\{(p_i^{\text{true}}, \gamma_i^{\text{true}})\}_{i=1}^m$ are randomly selected in $[-1, 1]^n \times [-1, 1]$, and the corresponding true solution $S_\epsilon$ to the PDE (36) is analytically computed using Eq. (34). Our target is to learn the initial function from the true solution evaluated on some sample points.

First, we construct a neural network with the architecture shown in Fig. 3 with $\theta_i = -\|p_i\|^2/2$ for each $i \in \{1, \ldots, m\}$. Similar to the case of the first-order HJ equations, the neural network is trained using the training data $\{(x_j, t_j, S_\epsilon(x_j, t_j))\}_{j=1}^N$, where $\{(x_j, t_j)\}_{j=1}^N$ are randomly selected with respect to the standard normal distribution in $\mathbb{R}^n \times [0, +\infty)$ for each $k \in \{1, \ldots, K\}$ (we take the absolute value for $t$ to make sure it is non-negative). The parameters in Adam optimizer are the same as in the case of first-order HJ equations in Sect. 4.1.1. After training, we compute the learned initial function $\tilde{J}_\epsilon$ by

$$\tilde{J}_\epsilon(x) := f_\epsilon(x, 0) = \epsilon \log \left( \sum_{i=1}^m e^{(p_i(x) - \gamma_i)/\epsilon} \right),$$

where $\{(p_i, \gamma_i)\}_{i=1}^m$ are the parameters in the trained neural network.

To compare the performance in different dimensions, we measure the relative mean square errors of the learned parameters $\{(p_i, \gamma_i)\}_{i=1}^m$ and the learned initial function $\tilde{J}_\epsilon$. Here, the parameter $\epsilon$ in the PDE is chosen to be 0.001. We test on the neural network with 2 and 4 neurons, i.e., we choose $m = 2$ and 4, and repeat the experiments 100 times. We show the averaged relative errors of the initial function and the parameters in Table 6 and 7 for $m = 2$ and 4, respectively. We observe small errors for the initial function $\tilde{J}_\epsilon$ and the parameters $\{(p_i, \gamma_i)\}_{i=1}^m$ in some cases when $m = 2$. The errors when $m = 4$ are much larger. However, we also observe that the errors of 16D and 32D are generally smaller than the low dimensional cases. In conclusion, there is no guarantee for the Adam optimizer to obtain the global minimizers for these problems.

Table 6. Relative mean square errors of the initial function $\tilde{J}_\epsilon$ and the parameters $\{p_i\}$ and $\{\gamma_i\}$ in the neural network $f_\epsilon$ with 2 neurons and $\epsilon = 10^{-3}$ in different dimensions, averaged over 100 repeated experiments.

|                | 1D  | 2D  | 4D  | 8D  | 16D | 32D |
|----------------|-----|-----|-----|-----|-----|-----|
| Relative Errors of $\tilde{J}_\epsilon$ | 2.59E-04 | 4.65E-03 | 4.23E-05 | 1.52E-09 | 1.47E-09 | 1.59E-09 |
| Relative Errors of $\{p_i\}$        | 3.53E-01 | 9.28E-02 | 5.71E-03 | 1.08E-09 | 1.23E-09 | 1.45E-09 |
| Relative Errors of $\{\gamma_i\}$   | 4.97E-01 | 7.48E-01 | 1.32E-03 | 3.38E-09 | 8.09E-09 | 2.25E-09 |

Table 7. Relative mean square errors of the initial function $\tilde{J}_\epsilon$ and the parameters $\{p_i\}$ and $\{\gamma_i\}$ in the neural network $f_\epsilon$ with 4 neurons and $\epsilon = 10^{-3}$ in different dimensions, averaged over 100 repeated experiments.

|                | 1D  | 2D  | 4D  | 8D  | 16D | 32D |
|----------------|-----|-----|-----|-----|-----|-----|
| Relative Errors of $\tilde{J}_\epsilon$ | 3.27E-04 | 1.08E-03 | 9.45E-04 | 1.07E-03 | 1.92E-04 | 6.88E-04 |
| Relative Errors of $\{p_i\}$        | 5.23E-01 | 3.54E-01 | 1.55E-01 | 6.14E-02 | 2.18E-03 | 1.09E-02 |
| Relative Errors of $\{\gamma_i\}$   | 6.11E-01 | 6.40E-01 | 2.73E-01 | 1.14E-01 | 2.71E-03 | 3.75E-02 |
5. Conclusion

Summary of the proposed work. In this paper, we have established novel mathematical connections between some classes of HJ PDEs with initial data and neural network architectures. Our results give conditions under which some neural network architectures represent viscosity solutions to HJ PDEs of the form \((1)\). These results do not rely on universal approximation properties of neural networks; rather, our results show that some neural networks correspond to representation formulas of HJ PDE solutions whose Hamiltonians \(H\) and initial data \(J\) are obtained from the parameters of these neural networks. This means that some neural network architectures naturally encode the physics contained in some HJ PDEs.

The first neural network architecture that we have proposed is depicted in Fig. 1. We have shown in Thm. 3.1 that under certain conditions on the parameters this neural network architecture represents the viscosity solution of the HJ PDEs \((10)\). The corresponding Hamiltonian and initial data can be recovered from the parameters of this neural network. As a corollary of this result for the one-dimensional case, we have proposed a second neural network architecture (depicted in Fig. 2) that represents the spatial gradient of the viscosity solution of the HJ PDEs \((1)\) (in one dimension and with \(\epsilon = 0\)) and have showed in Prop. 3.1 that under appropriate conditions on the parameters this neural network corresponds to entropy solutions of the conservation laws \((33)\). Finally, we have proposed a third neural network architecture (depicted in Fig. 3) and have shown in Prop. 3.2 that it represents the solution to a second-order HJ PDE \((36)\), where the initial data is obtained from the parameters of the neural network.

Let us emphasize that the two proposed neural network architectures shown in Fig. 1 and Fig. 2 that represent solutions to the HJ PDEs \((10)\) and \((36)\), respectively, allow us to numerically evaluate their solutions in high dimension without using grids or numerical approximations.

We have also tested the proposed neural network architectures on some inverse problems. Our numerical experiments in Sect. 4 show that these problems cannot generally be solved with the state-of-the-art Adam optimizer algorithm with high accuracy. These numerical results suggest further developments of efficient neural network training algorithms for solving inverse problems with our proposed neural network architectures.

Perspectives on other neural network architectures and HJ PDEs. We now present extensions of the proposed architectures that are viable candidates for representing solutions of HJ PDEs.

First consider the following multi-time HJ PDE \([11, 24, 33, 90, 104, 110, 117, 125]\) which reads

\[
\begin{align*}
\frac{\partial s_j}{\partial t_j} + H_j(\nabla x S) = 0 & \quad \text{for any } j \in \{1, \ldots, N\}, \quad x \in \mathbb{R}^n, t_1, \ldots, t_N > 0, \\
S(x, 0, \ldots, 0) = J(x), & \quad x \in \mathbb{R}^n.
\end{align*}
\]  

A generalized Hopf formula \([35, 90, 117]\) for this multi-time HJ equation is given by

\[
S(x, t_1, \ldots, t_N) = \left( \sum_{i=1}^{N} t_i H_i + J^* \right)^* (x) = \sup_{p \in \mathbb{R}^n} \left\{ \langle p, x \rangle - \sum_{j=1}^{N} t_j H_j(p) - J^*(p) \right\},
\]

for any \(x \in \mathbb{R}^n\) and \(t_1, \ldots, t_N \geq 0\). Based on this formula, we propose a neural network architecture, depicted in Fig. 6 whose mathematical definition is given by

\[
f(x, t_1, \ldots, t_N) = \max_{i \in \{1, \ldots, m\}} \left\{ \langle p_i, x \rangle - \sum_{j=1}^{N} t_j \theta_{ij} - \gamma_i \right\},
\]

where \(\{(p_i, \theta_{ij}, \ldots, \theta_{iN}, \gamma_i)\}_{i=1}^{m} \subset \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}\) is the set of parameters. The generalized Hopf formula \((43)\) suggests that the neural network architecture depicted in Fig. 6 is a good candidate for representing the solution to \((42)\) under appropriate conditions on the parameters of the network.

As mentioned in \([90]\), the multi-time HJ equation \((42)\) may not have viscosity solutions. However, under suitable assumptions \([11, 24, 33, 104]\), the generalized Hopf formula \((43)\) is a viscosity solution of the multi-time HJ equation. We intend to clarify the connections between the generalized Hopf formula, multi-time HJ PDEs, viscosity solutions and general solutions in a future work.
Figure 6. Illustration of the structure of the neural network (44) that can represent solutions to some first-order multi-time HJ equations.

Figure 7. Illustration of the structure of the ResNet-type neural network (45) that can represent the minimizer $u$ in the Lax-Oleinik formula. Note that the activation function is defined using the gradient of the Hamiltonian $H$, i.e., $\nabla H$. 
In [34, 35], it is shown that when the Hamiltonian $H$ and the initial data $J$ are both convex, and under appropriate assumptions, the solution $S$ to the following HJ PDE

\[
\begin{cases}
\frac{\partial S}{\partial t}(x, t) + H(\nabla_x S(x, t)) = 0 & \text{in } \mathbb{R}^n \times (0, +\infty), \\
S(x, 0) = J(x) & \text{in } \mathbb{R}^n \times \{0\},
\end{cases}
\]

is represented by the Hopf [62] and Lax-Oleinik formulas [44, Sect. 10.3.4]. These formulas read

\[
S(x, t) = \max_{p \in \mathbb{R}^n} \{ \langle p, x \rangle - J^*(p) - tH(p) \} \quad \text{(Hopf formula)}
\]

\[
= \min_{u \in \mathbb{R}^n} \left\{ J(u) + tH^* \left( \frac{x - u}{t} \right) \right\}. \quad \text{(Lax-Oleinik formula)}
\]

Let $p(x, t)$ be the maximizer in the Hopf formula and $u(x, t)$ be the minimizer in the Lax-Oleinik formula. Then, they satisfy the following relation [34, 35]

\[
u(x, t) = x - t\nabla H(p(x, t)).
\]

Fig. 7 depicts an architecture of a neural network that implements the formula above for the minimizer $u(x, t)$. In other words, we consider the ResNet-type neural network defined by

\[
u_k = \nu_{k-1} - t_k \nabla H(p_{j_k}^k), \quad \text{for each } k \in \{1, \ldots, N \},
\]

where $u_0 = x$ and $p_{j_k}^k$ is the output of the argmax based activation function in the $k^{th}$ layer. For the case when $N = 2$, an illustration of this deep ResNet architecture with two layers is shown in Fig. 8. In fact, this deep ResNet architecture can be formulated as follows

\[
u_N = x - \sum_{k=1}^N t_k \nabla H(p_{j_k}^k).\]
This formulation suggests that this architecture should also provide the minimizers of the generalized Lax-Oleinik formula for the multi-time HJ PDEs [55]. These results will be presented in detail in a forthcoming paper.

**Appendix A. Proofs of lemmas in Section 3**

A.1. **Proof of Lemma 3.1.** Proof of (i): The convex and lower-semicontinuous function \( J \) satisfies Eq. (12) by [61] Prop. X.3.4.1. It is also finite and continuous over its polytopal domain \( \text{dom} J = \text{conv} \{p_i\}_{i=1}^m \) [118] Thms. 10.2 and 20.5, and moreover the subdifferential \( \partial J^* \) is non-empty by [118] Thm. 23.10.

Proof of (ii): First, suppose the vector \((\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m\) satisfies the constraints (a)–(c). Since \( x \in \partial J^*(p) \), there holds \( J^*(p) = \langle p, x \rangle - J(x) \) [61] Cor. X.1.4.4, and using the definition of the set \( I_x \) [11] and constraints (a)–(c) we deduce that

\[
J^*(p) = \langle p, x \rangle - J(x) = \langle p, x \rangle - \sum_{i \in I_x} \alpha_i J(x)
\]

\[
= \langle p, x \rangle - \sum_{i \in I_x} \alpha_i (\langle p_i, x \rangle - \gamma_i)
\]

\[
= \left( p - \sum_{i \in I_x} \alpha_i p_i, x \right) + \sum_{i \in I_x} \alpha_i \gamma_i = \sum_{i=1}^m \alpha_i \gamma_i.
\]

Therefore, \((\alpha_1, \ldots, \alpha_m)\) is a minimizer in Eq. (12). Second, let \((\alpha_1, \ldots, \alpha_m)\) be a minimizer in Eq. (12). Then (a)–(b) follows directly from the constraints in Eq. (12). A similar argument as above yields

\[
J(x) = \langle p, x \rangle - J^*(p) = \left( \sum_{i=1}^m \alpha_i p_i, x \right) - \sum_{i=1}^m \alpha_i \gamma_i = \sum_{i=1}^m (\langle p_i, x \rangle - \gamma_i).
\]

But \( J(x) = \max_{i \in \{1, \ldots, m\}} \{\langle p_i, x \rangle - \gamma_i\} \) by definition, and so there holds \( \alpha_i = 0 \) whenever \( J(x) \neq \langle p_i, x \rangle - \gamma_i \). In other words, \( \alpha_i = 0 \) whenever \( i \notin I_x \).

Proof of (iii): Let \((\beta_1, \ldots, \beta_m) \in \Lambda_m\) satisfy \( \sum_{i=1}^m \beta_i p_i = p_k \). By assumption (A2), we have \( \gamma_k = g(p_k) \) with \( g \) convex, and hence Jensen’s inequality yields

\[
\sum_{i=1}^m \delta_k \gamma_i = \gamma_k = g(p_k) = g \left( \sum_{i=1}^m \beta_i p_i \right) \leq \sum_{i=1}^m \beta_i g(p_i) = \sum_{i=1}^m \beta_i \gamma_i.
\]

Therefore, the vector \((\delta_{1k}, \ldots, \delta_{mk})\) is a minimizer in Eq. (12) at the point \( p_k \), and \( J^*(p_k) = \gamma_k \) follows.

A.2. **Proof of Lemma 3.2.**Proof of (i): Let \( p \in \text{dom} J^* \). The set \( A(p) \subseteq \Lambda_m \) is non-empty and bounded by Lem. 3.1(i), and it is closed since \( A(p) \) is the solution set to the linear programming problem (12). Hence, \( A(p) \) is compact. As a result, we immediately have that \( H(p) < +\infty \). Moreover, for each \((\alpha_1, \ldots, \alpha_m) \in A(p)\) there holds

\[
-\infty < \min_{i=1}^m \theta_i \leq \sum_{i=1}^m \alpha_i \theta_i \leq \max_{i=1}^m \theta_i < +\infty,
\]

from which we conclude that \( H \) is a bounded function on \( \text{dom} J^* \). Since the target function in the minimization problem (14) is continuous, existence of a minimizer follows by compactness of \( A(p) \).

Proof of (ii): We have already shown in the proof of (i) that the restriction of \( H \) to \( \text{dom} J^* \) is bounded, and so it remains to prove its continuity. For any \( p \in \text{dom} J^* \), we have that \((\alpha_1, \ldots, \alpha_m) \in A(p)\) if and only if \((\alpha_1, \ldots, \alpha_m) \in \Lambda_m\), \( \sum_{i=1}^m \alpha_i p_i = p \), and \( \sum_{i=1}^m \alpha_i \gamma_i = J^*(p) \). As a result, we have

\[
H(p) = \min \left\{ \sum_{i=1}^m \alpha_i \theta_i : (\alpha_1, \ldots, \alpha_m) \in \Lambda_m, \sum_{i=1}^m \alpha_i p_i = p, \sum_{i=1}^m \alpha_i \gamma_i = J^*(p) \right\}.
\]

Define the function \( h : \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\} \) by

\[
h(p, r) := \min \left\{ \sum_{i=1}^m \alpha_i \theta_i : (\alpha_1, \ldots, \alpha_m) \in \Lambda_m, \sum_{i=1}^m \alpha_i p_i = p, \sum_{i=1}^m \alpha_i \gamma_i = r \right\},
\]

for any \( p \in \mathbb{R}^n \) and \( r \in \mathbb{R} \). Using the same argument as in the proof of Lem. 3.1(i), we conclude that \( h \) is a convex lower semi-continuous function, and in fact continuous over its domain \( h = \text{conv} \{ \langle p_i, \gamma_i \rangle \}_{i=1}^m \).
Comparing Eq. (47) and the definition of \( h \) in (48), we deduce that \( H(p) = h(p, J^*(p)) \) for any \( p \in \text{dom } J^* \). Continuity of \( H \) in \( \text{dom } J^* \) then follows from the continuity of \( h \) and \( J^* \) in their own domains.

Proof of (iii): Let \( k \in \{1, \ldots, m\} \). On the one hand, Lem. 3.1(iii) implies \((\delta_{1k}, \ldots, \delta_{mk}) \in A(p_k)\), so that

\[
H(p_k) \leq \sum_{i=1}^{m} \delta_{ik} \theta_i = \theta_k.
\]  

(49)

On the other hand, let \((\alpha_1, \ldots, \alpha_m) \in A(p_k)\) be a vector different from \((\delta_{1k}, \ldots, \delta_{mk})\). Then \((\alpha_1, \ldots, \alpha_m) \in \Lambda_m\) satisfies \(\sum_{i=1}^{m} \alpha_i \theta_i = p_k\), \(\sum_{i=1}^{m} \alpha_i \gamma_i = J^*(p_k)\), and \(\alpha_k < 1\). Define \((\beta_1, \ldots, \beta_m) \in \Lambda_m\) by

\[
\beta_j := \begin{cases} 
\frac{\alpha_j}{1 - \alpha_k}, & \text{if } j \neq k, \\
0, & \text{if } j = k.
\end{cases}
\]

A straightforward computation using the properties of \((\alpha_1, \ldots, \alpha_m)\), Lem. 3.1(iii), and the definition of \((\beta_1, \ldots, \beta_m)\) yields

\[
\begin{cases}
\sum_{i \neq k} \beta_i p_i = \sum_{i \neq k} \frac{\alpha_i p_i}{1 - \alpha_k} = \frac{p_k - \alpha_k p_k}{1 - \alpha_k} = p_k, \\
\sum_{i \neq k} \beta_i \gamma_i = \sum_{i \neq k} \frac{\alpha_i \gamma_i}{1 - \alpha_k} = \frac{J^*(p_k) - \alpha_k \gamma_k}{1 - \alpha_k} = \gamma_k - \frac{\alpha_k \gamma_k}{1 - \alpha_k} = \gamma_k.
\end{cases}
\]

In other words, Eq. (9) holds at index \( k \), which, by assumption (A3), implies that \(\sum_{i \neq k} \beta_i \theta_i > \theta_k\). As a result, we have

\[
\sum_{i=1}^{m} \alpha_i \theta_i = \alpha_k \theta_k + (1 - \alpha_k) \sum_{i \neq k} \beta_i \theta_i > \alpha_k \theta_k + (1 - \alpha_k) \theta_k = \theta_k = \sum_{i=1}^{m} \delta_{ik} \theta_i.
\]

Taken together with Eq. (49), we conclude that \((\delta_{1k}, \ldots, \delta_{mk})\) is the unique minimizer in (14), and hence we obtain \(H(p_k) = \theta_k\).

A.3. Proof of Lemma 3.3. Let \( x \in \mathbb{R}^n \) and \( t \geq 0 \). Since \( \tilde{H}(p) \geq H(p) \) for every \( p \in \text{dom } J^* \), we get

\[
\langle p, x \rangle - t\tilde{H}(p) - J^*(p) \leq \langle p, x \rangle - tH(p) - J^*(p).
\]

(50)

Let \((\alpha_1, \ldots, \alpha_m)\) be a minimizer in (14). By Eqs. (12), (13), and (14), we have

\[
p = \sum_{i=1}^{m} \alpha_i p_i, \quad H(p) = \sum_{i=1}^{m} \alpha_i \theta_i, \quad \text{and } J^*(p) = \sum_{i=1}^{m} \alpha_i \gamma_i.
\]

(51)

Combining Eqs. (50), (51), and (8), we get

\[
\langle p, x \rangle - t\tilde{H}(p) - J^*(p) \leq \sum_{i=1}^{m} \alpha_i (\langle p_i, x \rangle - t\theta_i - \gamma_i)
\]

\[
\leq \max_{i \in \{1, \ldots, m\}} \{\langle p_i, x \rangle - t\theta_i - \gamma_i\} = f(x, t),
\]

where the second inequality follows from the constraint \((\alpha_1, \ldots, \alpha_m) \in \Lambda_m\). Since \( p \in \text{dom } J^* \) is arbitrary, we obtain

\[
\sup_{p \in \text{dom } J^*} \{\langle p, x \rangle - t\tilde{H}(p) - J^*(p)\} \leq f(x, t).
\]

(52)

Now, by Lem. 3.1(iii), Lem. 3.2(iii), and the assumptions on \( \tilde{H} \), we have

\[\tilde{H}(p_k) = H(p_k) = \theta_k, \quad \text{and } J^*(p_k) = \gamma_k,\]
for each \( k = 1, \ldots, m \). A straightforward computation yields
\[
f(x, t) = \max_{i=1,\ldots,m} \{ \langle p_i, x \rangle - t \theta_i - \gamma_i \}
\]
\[
= \max_{i=1,\ldots,m} \left\{ \langle p_i, x \rangle - t \tilde{H}(p_i) - J^*(p_i) \right\}
\leq \sup_{p \in \text{dom } J^*} \left\{ \langle p, x \rangle - t \tilde{H}(p) - J^*(p) \right\},
\]
where the inequality holds since \( p_i \in \text{dom } J^* \) for every \( i \in \{1, \ldots, m\} \). The conclusion then follows from Eqs. (52) and (53).

A.4. Proof of Lemma 3.4. Since \( f \) is the supremum of a finite number of affine functions by definition (8), it is finite-valued and convex for \( t \geq 0 \). As a result, \( \nabla_x f(x, t) = p_k \) is equivalent to \( \partial(f(\cdot, t))(x) = \{ p_k \} \), and so it suffices to prove that \( \partial(f(\cdot, t))(x) = \{ p_k \} \) for some \( x \in \mathbb{R}^n \) and \( t > 0 \). To simplify the notation, we use \( \partial_x f(x, t) \) to denote the subdifferential of \( f(\cdot, t) \) at \( x \).

By [60, Thm. VI.4.4.2], the subdifferential of \( f(\cdot, t) \) at \( x \) is the convex hull of the \( p_i \)'s whose indices \( i \)'s are maximizers in (8), that is,
\[
\partial_x f(x, t) = \text{co } \{ p_i : i \text{ is a maximizer in } (8) \}.
\]
It suffices then to prove the existence of \( x \in \mathbb{R}^n \) and \( t > 0 \) such that
\[
\langle p_k, x \rangle - t \theta_k - \gamma_k > \langle p_i, x \rangle - t \theta_i - \gamma_i, \quad \text{for every } i \neq k.
\]

First, consider the case when there exists \( x \in \mathbb{R}^n \) such that \( \langle p_k, x \rangle - \gamma_k > \langle p_i, x \rangle - \gamma_i \) for every \( i \neq k \). In that case, by continuity, there exists small \( t > 0 \) such that \( \langle p_k, x \rangle - t \theta_k - \gamma_k > \langle p_i, x \rangle - t \theta_i - \gamma_i \) for every \( i \neq k \). Then (54) holds.

Now, consider the case when there does not exist \( x \in \mathbb{R}^n \) such that \( \langle p_k, x \rangle - \gamma_k > \max_{i \neq k} \{ \langle p_i, x \rangle - \gamma_i \} \). In other words, we assume
\[
J(x) = \max_{i \neq k} \{ \langle p_i, x \rangle - \gamma_i \} \quad \text{for every } x \in \mathbb{R}^n.
\]
We apply Lem. 3.1(i) to the formula above and obtain
\[
J^*(p_k) = \min \left\{ \sum_{i=1}^m \alpha_i \gamma_i : (\alpha_1, \ldots, \alpha_m) \in \Lambda_m, \sum_{i=1}^m \alpha_i p_i = p_k, \alpha_k = 0 \right\}.
\]
Let \( x_0 \in \partial J^*(p_k) \). Denote by \( I_{x_0} \) the set of maximizers in Eq. (55) at the point \( x_0 \), i.e.,
\[
I_{x_0} := \text{arg } \max_{i \neq k} \{ \langle p_i, x \rangle - \gamma_i \}.
\]
Note that we have \( k \notin I_{x_0} \) by definition of \( I_{x_0} \). Define a function \( h: \mathbb{R}^n \to \mathbb{R} \cup \{ +\infty \} \) by
\[
h(p) := \begin{cases} 
\theta_i, & \text{if } p = p_i \text{ and } i \in I_{x_0}, \\
+\infty, & \text{otherwise}.
\end{cases}
\]
Denote the convex lower semi-continuous envelope of \( h \) by \( \overline{h} \). Since \( x_0 \in \partial J^*(p_k) \), we can use [60, Thm. VI.4.4.2] and the definition of \( I_{x_0} \) and \( h \) in Eqs. (57) and (58) to deduce
\[
p_k \in \partial J(x_0) = \text{co } \{ p_i : i \in I_{x_0} \} = \text{dom } \overline{h}.
\]
Hence the point \( p_k \) is in the domain of the polytopal convex function \( \overline{h} \). Then [118, Thm. 23.10] implies \( \partial(\overline{h})(p_k) \neq \emptyset \). Let \( v_0 \in \partial(\overline{h})(p_k) \) and \( x = x_0 + t v_0 \). It remains to choose suitable positive \( t \) such that (54) holds. Letting \( x = x_0 + t v_0 \) in (54) yields
\[
\langle p_k, x \rangle - t \theta_k - \gamma_k - (\langle p_i, x \rangle - t \theta_i - \gamma_i)
= \langle p_k, x_0 + t v_0 \rangle - t \theta_k - \gamma_k - (\langle p_i, x_0 + t v_0 \rangle - t \theta_i - \gamma_i)
= \langle p_k, x_0 \rangle - \gamma_k - (\langle p_i, x_0 \rangle - \gamma_i) + t(\theta_i - \theta_k - (p_i - p_k, v_0)).
\]
Now, we consider two situations, the first when \( i \notin I_{x_0} \cup \{ k \} \) and the second when \( i \in I_{x_0} \). It suffices to prove (54) hold in each case for small enough positive \( t \).
Thus the inequalities become equalities in the equation above. As a result, we have
\[
J(x_0) \geq \langle p_k, x_0 \rangle - \gamma_k = \langle p_k, x_0 \rangle - g(p_k) = \left( \sum_{j \in I_{x_0}} c_j p_j, x_0 \right) - g \left( \sum_{j \in I_{x_0}} c_j p_j \right) \]
\[
\geq \sum_{j \in I_{x_0}} c_j \left( \langle p_j, x_0 \rangle - g(p_j) \right) = \sum_{j \in I_{x_0}} c_j J(x_0) = J(x_0).
\]

Thus the inequalities become equalities in the equation above. As a result, we have
\[
\langle p_k, x_0 \rangle - \gamma_k = J(x_0) > \langle p_i, x_0 \rangle - \gamma_i,
\]
where the inequality holds because \( i \notin I_{x_0} \cup \{k\} \) by assumption. This inequality implies that the constant \( \langle p_k, x_0 \rangle - \gamma_k - \langle p_i, x_0 \rangle - \gamma_i \) is positive, and taken together with \( \text{Eq. \ref{Jx0}} \), we conclude that the inequality in \( \text{Eq. \ref{Jx0}} \) holds for \( i \notin I_{x_0} \cup \{k\} \) when \( t \) is small enough.

If \( i \in I_{x_0} \), then both \( i \) and \( k \) are maximizers in \( \text{Eq. \ref{Jx0}} \) at \( x_0 \), and hence we have
\[
\langle p_k, x_0 \rangle - \gamma_k = J(x_0) = \langle p_i, x_0 \rangle - \gamma_i.
\]

Together with \( \text{Eq. \ref{Jx0}} \) and the definition of \( h \) in \( \text{Eq. \ref{Jx0}} \), we obtain
\[
\langle p_k, x \rangle - t \theta_k - \gamma_k - (\langle p_i, x \rangle - t \theta_i - \gamma_i) = 0 + t(h(p_i) - \theta_k - \langle p_i - p_k, v_0 \rangle)
\]
\[
\geq t(h_0(p_i) - \theta_k - \langle p_i - p_k, v_0 \rangle).
\]

In addition, since \( v_0 \in \partial(h(p_k)) \), we have
\[
h_0(p_i) \geq h(p_k) + \langle p_i - p_k, v_0 \rangle.
\]

Combining Eqs. \( \text{Eq. \ref{Jx0}} \) and \( \text{Eq. \ref{Jx0}} \), we obtain
\[
\langle p_k, x \rangle - t \theta_k - \gamma_k - (\langle p_i, x \rangle - t \theta_i - \gamma_i) \geq t(h(p_k) - \theta_k).
\]

To prove the result, it suffices to show \( h(p_k) > \theta_k \). As \( p_k \in h \) (as shown before in \( \text{Eq. \ref{Jx0}} \)), then according to \( \text{Prop. X.1.5.4} \) we have
\[
h(p_k) = \sum_{j \in I_{x_0}} \alpha_j h(p_j) = \sum_{j \in I_{x_0}} \alpha_j \theta_j,
\]
for some \( \alpha_1, \ldots, \alpha_m \in \Lambda_m \) satisfying \( p_k = \sum_{j=1}^{m} \alpha_j p_j \) and \( \alpha_j = 0 \) whenever \( j \notin I_{x_0} \). Then, by Lem. \( \text{3.1 \( \text{ii} \)} \) \((\alpha_1, \ldots, \alpha_m) \) is a minimizer in \( \text{Eq. \ref{Jx0}} \), that is,
\[
\gamma_k = J^*(p_k) = \sum_{j=1}^{m} \alpha_j \gamma_j = \sum_{j \in I_{x_0}} \alpha_j \gamma_i = \sum_{i \notin I_{x_0}} \alpha_i \gamma_i.
\]

Hence Eq. \( \text{Eq. \ref{Jx0}} \) holds for the index \( k \). By assumption \( \text{A3} \), we have \( \theta_k < \sum_{j \neq k} \alpha_j \theta_j \). Taken together with the fact that \( \alpha_j = 0 \) whenever \( j \notin I_{x_0} \) and Eq. \( \text{Eq. \ref{Jx0}} \), we find
\[
\theta_k < \sum_{j \neq k} \alpha_j \theta_j = \sum_{j \in I_{x_0}} \alpha_j \theta_j = h(p_k).
\]

Hence, the right-hand-side of Eq. \( \text{Eq. \ref{Jx0}} \) is strictly positive, and we conclude that \( \langle p_k, x \rangle - t \theta_k - \gamma_k > \langle p_i, x \rangle - t \theta_i - \gamma_i \) for \( t > 0 \) if \( i \in I_{x_0} \).

Therefore, in this case, when \( t > 0 \) is small enough and \( x \) is chosen as above, we have \( \langle p_k, x \rangle - t \theta_k - \gamma_k > \langle p_i, x \rangle - t \theta_i - \gamma_i \) for every \( i \neq k \), and the proof is complete.
A.5. **Proof of Lemma 3.5.** First, we compute the convex hull of epi $F$, which we denote by $\text{co (epi } F\text{)}$. Let $(p, E^-, r) \in \text{co (epi } F\text{)}$, where $p \in \mathbb{R}^n$ and $E^-, r \in \mathbb{R}$. Then there exist $k \in \mathbb{N}$, $(\beta_1, \ldots, \beta_k) \in \Lambda_k$ and $(q_i, E_i^-, r_i) \in \text{epi } F$ for each $i = 1, \ldots, k$ such that $(p, E^-, r) = \sum_{i=1}^k \beta_i (q_i, E_i^-, r_i)$. By definition of $F$ in Eq. (18), $(q_i, E_i^-, r_i) \in \text{epi } F$ holds if and only if $q_i \in \text{dom } J^*$, $E_i^- + H(q_i) \leq 0$ and $r_i \geq J^*(q_i)$. In conclusion, we have

$$
\begin{aligned}
(\beta_1, \ldots, \beta_k) &\in \Lambda_k, \\
(p, E^-, r) &\in \sum_{i=1}^k \beta_i (q_i, E_i^-, r_i), \\
q_1, \ldots, q_k &\in \text{dom } J^*, \\
E_i^- + H(q_i) &\leq 0 \text{ for any } i = 1, \ldots, k, \\
r_i &\geq J^*(q_i) \text{ for any } i = 1, \ldots, k.
\end{aligned}
\tag{67}
$$

For each $i$, since we have $q_i \in \text{dom } J^*$, by Lem. 3.2(i) the minimization problem in (14) evaluated at $q_i$ has at least one minimizer. Let $(\alpha_{i1}, \ldots, \alpha_{im})$ be such a minimizer. Using Eqs. (12), (14), and $(\alpha_{i1}, \ldots, \alpha_{im}) \in \Lambda_m$, we have

$$
\sum_{j=1}^m \alpha_{ij} (1, p_j, \theta_j, \gamma_j) = (1, q_i, H(q_i), J^*(q_i)).
\tag{68}
$$

Define the real number $c_j \equiv \sum_{i=1}^k \beta_i \alpha_{ij}$ for any $j = 1, \ldots, m$. Combining Eqs. (67) and (68), we get that $c_j \geq 0$ for any $j$ and

$$
\sum_{j=1}^m c_j (1, p_j, \theta_j, \gamma_j) = \sum_{j=1}^m \sum_{i=1}^k \beta_i \alpha_{ij} (1, p_j, \theta_j, \gamma_j) = \sum_{i=1}^k \beta_i (1, q_i, H(q_i), J^*(q_i)).
$$

We continue the computation using Eq. (67) and get

$$
\begin{aligned}
\sum_{j=1}^m c_j (1, p_j) &= \sum_{i=1}^k \beta_i (1, q_i) = (1, p), \\
\sum_{j=1}^m c_j \theta_j &= \sum_{i=1}^k \beta_i H(q_i) \leq - \sum_{i=1}^k \beta_i E_i^- = -E^-, \\
\sum_{j=1}^m c_j \gamma_j &= \sum_{i=1}^k \beta_i J^*(q_i) \leq \sum_{i=1}^k \beta_i r_i = r.
\end{aligned}
$$

Therefore, we conclude that $(c_1, \ldots, c_m) \in \Lambda_m$ and

$$
\begin{aligned}
p &= \sum_{j=1}^m c_j p_j, \\
E^- &\leq - \sum_{j=1}^m c_j \theta_j, \\
r &\geq \sum_{j=1}^m c_j \gamma_j.
\end{aligned}
$$

As a consequence, $\text{co (epi } F\text{)} \subseteq \text{co } \left( \bigcup_{j=1}^m \{(p_j) \times (-\infty, -\theta_j] \times [\gamma_j, +\infty)\} \right)$. Now, Lem. 3.1(iii) and 3.2(iii) imply $\{p_j\} \times (-\infty, -\theta_j] \times [\gamma_j, +\infty) \subseteq \text{epi } F$ for each $j = 1, \ldots, m$. Therefore, we have

$$
\text{co (epi } F\text{)} = \left\{ (p, E^-, r) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} : \text{there exists } (c_1, \ldots, c_m) \in \Lambda_m \text{ s.t.} \right\}
\tag{69}
$$

$$
\begin{aligned}
p &= \sum_{j=1}^m c_j p_j, \\
E^- &\leq - \sum_{j=1}^m c_j \theta_j, \\
r &\geq \sum_{j=1}^m c_j \gamma_j.
\end{aligned}
$$

By Def. IV.2.5.3 and Prop. IV.2.5.1, we have

$$
\text{co } F(p, E^-) = \inf \{r \in \mathbb{R} : (p, E^-, r) \in \text{co (epi } F\text{)}\}.
\tag{70}
$$
The conclusion then follows from Eqs. (69) and (70).

### Appendix B. Proof of Proposition 3.1

To prove this proposition, we will use three lemmas whose statements and proofs are given in Sects. B.1, B.2 and B.3 respectively. The proof of Prop. 3.1 is given in Sect. B.4.

#### B.1. Statement and proof of Lemma B.1

**Lemma B.1.** Consider the one-dimensional case, i.e., $n = 1$. Let $p_1, \ldots, p_m \in \mathbb{R}$ satisfy $p_1 < \cdots < p_m$, and define the function $J$ using Eq. (17). Suppose assumptions (A1)-(A2) hold. Let $x \in \mathbb{R}$, $p \in \partial J(x)$, and suppose $p \neq p_i$ for any $i \in \{1, \ldots, m\}$. Then there exists $k \in \{1, \ldots, m\}$ such that $p_k < p < p_{k+1}$ and

$$k, k+1 \in \arg \max_{i \in \{1, \ldots, m\}} \{x \gamma_i\}.$$  

**Proof.** Let $I_x$ denotes the set of maximizers in Eq. (11) at $x$. Since $p \in \partial J(x)$, $p \neq p_i$ for $i \in \{1, \ldots, m\}$, and $\partial J(x) = \co \{p_i : i \in I_x\}$ by [60], Thm. VI.4.4.2], there exist $j, l \in I_x$ such that $p_j < p < p_l$. Moreover, there exists $k$ with $j \leq k < k+1 \leq l$ such that $p_j \leq p < p_{k+1} \leq p_l$. We will show that $k, k+1 \in I_x$.

We only prove $k \in I_x$; the case for $k+1$ is similar.

If $p_j = p_k$, then $j = k \in I_x$ and the conclusion follows directly. Hence suppose $p_j < p_k < p_l$. Then there exists $\alpha \in (0, 1)$ such that $p_k = \alpha p_j + (1-\alpha)p_l$. Using that $j, l \in I_x$, assumption (A2), and Jensen inequality, we get

$$xp_k - \gamma_k = xp_k - g(p_k) = (\alpha p_j + (1-\alpha)p_l) x - g(\alpha p_j + (1-\alpha)p_l)$$

$$\geq \alpha xp_j + (1-\alpha)xp_l - \alpha g(p_j) - (1-\alpha)g(p_l)$$

$$= \alpha(xp_j - \gamma_j) + (1-\alpha)(xp_l - \gamma_l)$$

$$= \max_{i \in \{1, \ldots, m\}} \{xp_i - \gamma_i\},$$

which implies that $k \in I_x$. A similar argument shows that $k+1 \in I_x$, which completes the proof. \qed

#### B.2. Statement and proof of Lemma B.2

**Lemma B.2.** Consider the one-dimensional case, i.e., $n = 1$. Let $p_1, \ldots, p_m \in \mathbb{R}$ satisfy $p_1 < \cdots < p_m$, and define the function $H$ using Eq. (14). Suppose assumptions (A1)-(A2) hold. Let $u_0 \in \mathbb{R}$ and $p_k < u_0 < p_{k+1}$ for some index $k$. Then there holds

$$H(u_0) = \beta_k \theta_k + \beta_{k+1} \theta_{k+1},$$

where

$$\beta_k := \frac{p_{k+1} - u_0}{p_{k+1} - p_k}$$

and

$$\beta_{k+1} := \frac{u_0 - p_k}{p_{k+1} - p_k}.$$  

**Proof.** Let $\beta := (\beta_1, \ldots, \beta_m) \in \Lambda_m$ satisfy

$$\beta_k := \frac{p_{k+1} - u_0}{p_{k+1} - p_k}$$

and

$$\beta_{k+1} := \frac{u_0 - p_k}{p_{k+1} - p_k},$$

and $\beta_i = 0$ for every $i \in \{1, \ldots, m\} \setminus \{k, k+1\}$. We will prove that $\beta$ is a minimizer in Eq. (14) evaluated at $u_0$, that is,

$$\beta \in \arg \min_{\alpha \in \mathcal{A}(u_0)} \left\{ \sum_{i=1}^{m} \alpha_i \theta_i \right\},$$

where

$$\mathcal{A}(u_0) := \arg \min_{\{\alpha_1, \ldots, \alpha_m\} \in \Lambda_m, \sum_{i=1}^{m} \alpha_i \theta_i = 0} \left\{ \sum_{i=1}^{m} \alpha_i \gamma_i \right\}.$$  

First, we show that $\beta \in \mathcal{A}(u_0)$. By definition of $\beta$ and Lem. 3.1(ii) with $p = u_0$, the statement holds provided $k, k+1 \in I_x$, where the set $I_x$ contains the maximizers in Eq. (10) evaluated at $x \in \partial J^\star(u_0)$. But if $x \in \partial J^\star(u_0)$, we have $u_0 \in \partial J(x)$, and Lem. B.1 implies $k, k+1 \in I_x$. Hence $\beta \in \mathcal{A}(u_0)$.
Now, suppose that \( \beta \) is not a minimizer in Eq. (14) evaluated at \( u_0 \). By Lem. 3.2(i), there exists a minimizer in Eq. (14) evaluated at the point \( u_0 \), which we denote by \( (\alpha_1, \ldots, \alpha_m) \). Then there holds

\[
\begin{aligned}
\sum_{i=1}^{m} \alpha_i &= \sum_{i=1}^{m} \beta_i = 1, \\
\sum_{i=1}^{m} \alpha_i \beta_i &= \sum_{i=1}^{m} \beta_i u_0, \\
\sum_{i=1}^{m} \alpha_i \gamma_i &= \sum_{i=1}^{m} \beta_i \gamma_i = J^*(u_0), \\
\sum_{i=1}^{m} \alpha_i \theta_i &= \sum_{i=1}^{m} \beta_i \theta_i.
\end{aligned}
\]

(74)

Since \( \alpha_i \geq 0 \) for every \( i \) and \( \beta_i = 0 \) for every \( i \in \{1, \ldots, m\} \setminus \{k, k+1\} \), we have \( \alpha_k + \alpha_{k+1} \leq 1 = \beta_k + \beta_{k+1} \). As \( \alpha \neq \beta \), then one or both of the inequalities \( \alpha_k < \beta_k \) and \( \alpha_{k+1} < \beta_{k+1} \) hold. This leaves three possible cases, and we now show that each case leads to a contradiction.

Case 1: Let \( \alpha_k < \beta_k \) and \( \alpha_{k+1} \geq \beta_{k+1} \). Define the coefficient \( c_i \) by

\[
c_i := \begin{cases} 
\frac{\alpha_i - \beta_i}{\beta_k - \alpha_k}, & i \neq k, \\
0, & i = k.
\end{cases}
\]

The following equations then hold

\[
\begin{aligned}
(c_1, \ldots, c_m) &\in \Delta_m \text{ with } c_k = 0, \\
\sum_{i \neq k} c_i \alpha_i &= p_k, \\
\sum_{i \neq k} c_i \beta_i &= \gamma_k, \\
\sum_{i \neq k} c_i \theta_i &= \theta_k.
\end{aligned}
\]

These equations, however, violate assumption (A3), and so we get a contradiction.

Case 2: Let \( \alpha_k \geq \beta_k \) and \( \alpha_{k+1} < \beta_{k+1} \). A similar argument as in case 1 can be applied here by exchanging the indices \( k \) and \( k+1 \) to derive a contradiction.

Case 3: Let \( \alpha_k < \beta_k \) and \( \alpha_{k+1} < \beta_{k+1} \). From Eq. (74), we obtain

\[
\begin{aligned}
\beta_k - \alpha_k + \beta_{k+1} - \alpha_{k+1} &= \sum_{i \neq k, k+1} \alpha_i, \\
(\beta_k - \alpha_k)p_k + (\beta_{k+1} - \alpha_{k+1})\gamma_{k+1} &= \sum_{i \neq k, k+1} \alpha_i \beta_i, \\
(\beta_k - \alpha_k)\gamma_k + (\beta_{k+1} - \alpha_{k+1})\theta_{k+1} &= \sum_{i \neq k, k+1} \alpha_i \gamma_i, \\
(\beta_k - \alpha_k)\theta_k + (\beta_{k+1} - \alpha_{k+1})\theta_{k+1} &= \sum_{i \neq k, k+1} \alpha_i \theta_i.
\end{aligned}
\]

(75)

Define two numbers \( q_k \) and \( q_{k+1} \) by

\[
q_k := \frac{\sum_{i \leq k} \alpha_i \beta_i}{\sum_{i \leq k} \alpha_i} \quad \text{and} \quad q_{k+1} := \frac{\sum_{i > k+1} \alpha_i \beta_i}{\sum_{i > k+1} \alpha_i}.
\]

(76)

Note that from the first two equations in (74) and the assumption that \( \alpha_k < \beta_k \) and \( \alpha_{k+1} < \beta_{k+1} \), there exist \( i_1 < k \) and \( i_2 > k+1 \) such that \( \alpha_{i_1} \neq 0 \) and \( \alpha_{i_2} \neq 0 \), and hence the numbers \( q_k \) and \( q_{k+1} \) are well-defined. By definition, we have \( q_k < p_k < q_{k+1} < q_{k+1} \). Therefore, there exist \( b_k, b_{k+1} \in (0, 1) \) such that

\[
p_k = b_k q_k + (1 - b_k) q_{k+1} \quad \text{and} \quad p_{k+1} = b_{k+1} q_k + (1 - b_{k+1}) q_{k+1}.
\]

(77)

A straightforward computation yields

\[
b_k = \frac{q_{k+1} - q_k}{q_k - q_k} \quad \text{and} \quad b_{k+1} = \frac{q_{k+1} - q_{k+1}}{q_k - q_{k+1}}.
\]

(78)

Define the coefficients \( c_i^k \) and \( c_i^{k+1} \) as follows

\[
c_i^k := \begin{cases} 
\frac{b_k \alpha_i}{\sum_{i \leq k} \alpha_i}, & i \leq k, \\
\frac{(1 - b_k) \alpha_i}{\sum_{i > k+1} \alpha_i}, & i > k+1, \\
0, & \text{otherwise},
\end{cases}
\]

and

\[
c_i^{k+1} := \begin{cases} 
\frac{b_{k+1} \alpha_i}{\sum_{i \leq k+1} \alpha_i}, & i \leq k, \\
\frac{(1 - b_{k+1}) \alpha_i}{\sum_{i > k+1} \alpha_i}, & i > k+1, \\
0, & \text{otherwise}.
\end{cases}
\]

(79)

These coefficients satisfy \( c_i^k, c_i^{k+1} \in [0, 1] \) for any \( i \) and \( \sum_{i=1}^{m} c_i^k = \sum_{i=1}^{m} c_i^{k+1} = 1 \). In other words, we have

\[
(c_1^k, \ldots, c_m^k) \in \Delta_m \text{ with } c_k^k = 0 \quad \text{and} \quad (c_1^{k+1}, \ldots, c_m^{k+1}) \in \Delta_m \text{ with } c_{k+1}^{k+1} = 0.
\]

(80)
Hence, the first equality in Eq. (9) holds for the coefficients \((c_1^k, \ldots, c_m^k)\) with the index \(k\) and also for the coefficients \((c_1^{k+1}, \ldots, c_m^{k+1})\) with the index \(k + 1\). We show next that these coefficients satisfy the second and third equalities in (9) and draw a contradiction with assumption (A3).

Using Eqs. (76), (77), and (79) to write the formulas for \(p_k\) and \(p_{k+1}\) via the coefficients \(c_i^k\) and \(c_i^{k+1}\), we find

\[
p_k = b_k \sum_{i<k} \frac{\alpha_i p_i}{\alpha_i} + (1 - b_k) \sum_{i>k} \frac{\alpha_i p_i}{\alpha_i} = \sum_{i \neq k} c_i^k p_i = \sum_{i \neq k} c_i^k p_i; \\
p_{k+1} = b_{k+1} \sum_{i<k} \frac{\alpha_i p_i}{\alpha_i} + (1 - b_{k+1}) \sum_{i>k+1} \frac{\alpha_i p_i}{\alpha_i} = \sum_{i \neq k+1} c_i^{k+1} p_i = \sum_{i \neq k+1} c_i^{k+1} p_i,
\]

where the last equalities in the two formulas above hold because \(c_i^k = 0\) and \(c_i^{k+1} = 0\) by definition. Hence the second equality in Eq. (9) also holds for both the index \(k\) and \(k + 1\).

From the third equality in Eq. (77), assumption (A2), Eq. (81), and Jensen’s inequality, we have

\[
\sum_{i \neq k, k+1} \alpha_i \gamma_i = (\beta_k - \alpha_k) \gamma_k + (\beta_{k+1} - \alpha_{k+1}) \gamma_{k+1} \\
= (\beta_k - \alpha_k) g(p_k) + (\beta_{k+1} - \alpha_{k+1}) g(p_{k+1}) \\
= (\beta_k - \alpha_k) g \left( \sum_{i \neq k, k+1} c_i^k p_i \right) + (\beta_{k+1} - \alpha_{k+1}) g \left( \sum_{i \neq k, k+1} c_i^{k+1} p_i \right) \\
\leq (\beta_k - \alpha_k) \left( \sum_{i \neq k, k+1} c_i^k g(p_i) \right) + (\beta_{k+1} - \alpha_{k+1}) \left( \sum_{i \neq k, k+1} c_i^{k+1} g(p_i) \right) \\
= \sum_{i \neq k, k+1} ((\beta_k - \alpha_k) c_i^k + (\beta_{k+1} - \alpha_{k+1}) c_i^{k+1}) g(p_i) \\
= \sum_{i \neq k, k+1} ((\beta_k - \alpha_k) c_i^k + (\beta_{k+1} - \alpha_{k+1}) c_i^{k+1}) \gamma_i.
\]

We now compute and simplify the coefficients \((\beta_k - \alpha_k)c_i^k + (\beta_{k+1} - \alpha_{k+1})c_i^{k+1}\) in the formula above. First, consider the case when \(i < k\). Eqs. (78) and (79) imply

\[
(\beta_k - \alpha_k)c_i^k + (\beta_{k+1} - \alpha_{k+1})c_i^{k+1} \\
= (\beta_k - \alpha_k) \frac{b_k \alpha_i}{\sum_{\omega<k} \alpha_{\omega}} + (\beta_{k+1} - \alpha_{k+1}) \frac{b_{k+1} \alpha_i}{\sum_{\omega<k} \alpha_{\omega}} \\
= \frac{\alpha_i}{\sum_{\omega<k} \alpha_{\omega}} ((\beta_k - \alpha_k) b_k + (\beta_{k+1} - \alpha_{k+1}) b_{k+1}) \\
= \frac{\alpha_i}{\sum_{\omega<k} \alpha_{\omega}} \left( \frac{q_{k+1} - p_k}{q_{k+1} - q_k} + (\beta_{k+1} - \alpha_{k+1}) \frac{q_{k+1} - p_{k+1}}{q_{k+1} - q_k} \right) \\
= \frac{\alpha_i}{\sum_{\omega<k} \alpha_{\omega}} \cdot \frac{1}{q_{k+1} - q_k} ((\beta_k - \alpha_k + \beta_{k+1} - \alpha_{k+1}) q_{k+1} - (\beta_k - \alpha_k) p_k - (\beta_{k+1} - \alpha_{k+1}) p_{k+1}).
\]
Applying the first two equalities in Eq. (75) and Eq. (76) to the last formula above, we obtain

\[(\beta_k - \alpha_k)c^k_i + (\beta_{k+1} - \alpha_{k+1})c^{k+1}_i\]

for all indices \(i \neq k, k+1\). Let

\[
\sum_{i\neq k,k+1} \frac{1}{q_{k+1} - q_k} \left( \sum_{i\neq k,k+1} \alpha_i p_i \right) - \sum_{i\neq k,k+1} \alpha_i p_i = 0,
\]

which contradicts the last inequality in Eq. (75).

The third equality in (9) also holds for both indices \(i, j, l \leq 1\). Let \(x\) be the minimizer in Eq. (14) evaluated at \(t_0\) and Eq. (72) follows from the definition of \(H\) in (14). \(\square\)

### B.3. Statement and proof of Lemma B.3

**Lemma B.3.** Consider the one-dimensional case, i.e., \(n = 1\). Let \(p_1, \ldots, p_m \in \mathbb{R}\) satisfy \(p_1 < \cdots < p_m\). Suppose assumptions (A1)-(A2) hold. Let \(x \in \mathbb{R}\) and \(t > 0\). Assume \(j, k, l \) are three indices such that \(1 \leq j \leq k < l \leq m\) and

\[
j, l \in \arg \max_{i\in\{1,\ldots,m\}} \{xp_i - t\theta_i - \gamma_i\}.
\]
Then there holds

\[ \frac{\theta_l - \theta_k}{p_l - p_k} \leq \frac{\theta_l - \theta_j}{p_l - p_j}. \]

**Proof.** Note that Eq. (86) holds trivially when \( j = k \), so we only need to consider the case when \( j < k < l \). On the one hand, Eq. (85) implies

\[ xp_j - t\theta_j - \gamma_j = xp_l - t\theta_l - \gamma_l \geq xp_k - t\theta_k - \gamma_k, \]

which yields

\[ \gamma_l - \gamma_k \leq x(p_l - p_k) - t(\theta_l - \theta_k), \]

\[ \gamma_l - \gamma_j = x(p_l - p_j) - t(\theta_l - \theta_j). \]

On the other hand, for each \( i \in \{ j, j+1, \ldots, l-1 \} \) let \( q_i \in (p_i, p_{i+1}) \) and \( x_i \in \partial J^*(q_i) \). Such \( x_i \) exists because \( q_i \in \text{int dom } J^* \), so that the subdifferential \( \partial J^*(q_i) \) is non-empty. Then \( q_i \in \partial J(x_i) \) and Lem. [B.1] imply

\[ x_ip_l - \gamma_i = x_ip_{i+1} - \gamma_{i+1} = \max_{\omega \in \{1, \ldots, m\}} \{ x_ip_{m} - \gamma_{m} \}. \]

A straightforward computation yields

\[ \gamma_l - \gamma_k = \sum_{i=k}^{l-1} (\gamma_{i+1} - \gamma_i) = \sum_{i=k}^{l-1} x_i(p_{i+1} - p_i), \]

\[ \gamma_l - \gamma_j = \sum_{i=j}^{l-1} (\gamma_{i+1} - \gamma_i) = \sum_{i=j}^{l-1} x_i(p_{i+1} - p_i). \]

Combining the two equalities above with Eq. (87), we conclude that

\[ x(p_l - p_k) - t(\theta_l - \theta_k) \geq \sum_{i=k}^{l-1} x_i(p_{i+1} - p_i), \]

\[ x(p_l - p_j) - t(\theta_l - \theta_j) = \sum_{i=j}^{l-1} x_i(p_{i+1} - p_i). \]

Now, divide the inequality above by \( t(p_l - p_k) > 0 \) (because by assumption \( t > 0 \) and \( l > k \), which implies that \( p_l > p_k \)), divide the equality above by \( t(p_l - p_j) > 0 \) (because \( l > j \), which implies that \( t(p_l - p_j) \neq 0 \)), and rearrange the terms to obtain

\[ \frac{\theta_l - \theta_k}{p_l - p_k} \leq \frac{x}{t} \frac{1}{t} \sum_{i=k}^{l-1} x_i(p_{i+1} - p_i), \]

\[ \frac{\theta_l - \theta_j}{p_l - p_j} \leq \frac{x}{t} \frac{1}{t} \sum_{i=j}^{l-1} x_i(p_{i+1} - p_i). \]

Recall that \( q_1 < q_{j+1} < \cdots < q_l \) and \( x_i \in \partial J^*(q_i) \) for any \( j \leq i < l \). Since the function \( J^* \) is convex, the subdifferential operator \( \partial J^* \) is a monotone non-decreasing operator [60] Def. IV.4.1.3, and Prop. VI.6.1.1], which yields \( x_j \leq x_{j+1} \leq \cdots \leq x_{l-1} \). Using that \( p_1 < p_2 < \cdots < p_m \) and \( j < k < l \), we obtain

\[ \sum_{i=k}^{l-1} x_i(p_{i+1} - p_i) \geq \sum_{i=k}^{l-1} x_k(p_{i+1} - p_i) \geq \sum_{i=j}^{k-1} x_k(p_{i+1} - p_i) \]

To proceed, we now use that fact that if four real numbers \( a, c \in \mathbb{R} \) and \( b, d > 0 \) satisfy \( \frac{a}{c} \geq \frac{d}{e} \), then \( \frac{a}{c} \geq \frac{d}{e} \). Combining this fact with inequality (89), we find

\[ \frac{\sum_{i=k}^{l-1} x_i(p_{i+1} - p_i)}{p_l - p_k} \geq \frac{\sum_{i=k}^{l-1} x_k(p_{i+1} - p_i)}{p_l - p_k} \]

We combine the inequality above with (88) to obtain

\[ \frac{\theta_l - \theta_k}{p_l - p_k} \leq \frac{\theta_l - \theta_j}{p_l - p_j}. \]
which concludes the proof. \(\square\)

B.4. Proof of Proposition 3.1

Proof of (i): First, note that \(u\) is piecewise constant. Second, recall that \(J\) is defined as the pointwise maximum of a finite number of affine functions. Therefore, the initial data \(u(\cdot, 0) = \nabla J(\cdot)\) (recall that here, the gradient \(\nabla\) is taken in the sense of distribution) is bounded and of locally bounded variation (see [13, Chap. 5, page 167] for the definition of locally bounded variation). Finally, the flux function \(H\), defined in Eq. (14), is Lipschitz continuous in \(\text{dom } J^*\) by Lem. B.2. It can therefore be extended to \(\mathbb{R}\) while preserving its Lipschitz property [31, Thm. 4.16]. Therefore, we can invoke [22, Prop. 2.1] to conclude that \(u\) is the entropy solution to the conservation law [32] provided it satisfies the two following conditions. Let \(\bar{x}(t)\) be any smooth line of discontinuity of \(u\). Fix \(t > 0\) and define \(u^-\) and \(u^+\) as

\[
 u^- := \lim_{x \to \bar{x}(t)^-} u(x, t), \quad \text{and} \quad u^+ := \lim_{x \to \bar{x}(t)^+} u(x, t).
\]

Then the two conditions are:

1. The curve \(\bar{x}(t)\) is a straight line with the slope

\[
 \frac{dx}{dt} = \frac{H(u^+) - H(u^-)}{u^+ - u^-}.
\]

2. For any \(u_0\) between \(u^+\) and \(u^-\), we have

\[
 \frac{H(u^+) - H(u_0)}{u^+ - u_0} \leq \frac{H(u^+) - H(u^-)}{u^+ - u^-}.
\]

First, we prove the first condition and Eq. (91). According to the definition of \(u\) in Eq. (91), the range of \(u\) is the compact set \([p_1, \ldots, p_m]\). As a result, \(u^-\) and \(u^+\) are in the range of \(u\), i.e., there exist indices \(j\) and \(l\) such that

\[
 u^- = p_j, \quad \text{and} \quad u^+ = p_l.
\]

Let \((\bar{x}(s), s)\) be a point on the curve \(\bar{x}\) which is not one of the endpoints. Since \(u\) is piecewise constant, there exists a neighborhood \(\mathcal{N}\) of \((\bar{x}(s), s)\) such that for any \((x^-, t), (x^+, t)\) \(\in \mathcal{N}\) satisfying \(x^- < \bar{x}(t) < x^+\), we have \(u(x^-, t) = u^- = p_j\) and \(u(x^+, t) = u^+ = p_l\). In other words, if \(x^-, x^+, t\) are chosen as above, according to the definition of \(u\) in Eq. (31), we have

\[
 j \in \arg\max_{i \in \{1, \ldots, m\}} \{x^- p_i - t \theta_i - \gamma_i\} \quad \text{and} \quad l \in \arg\max_{i \in \{1, \ldots, m\}} \{x^+ p_i - t \theta_i - \gamma_i\}.
\]

Define a sequence \(\{x_k^\pm\}_{k=1}^{\infty} \subset (\infty, \bar{x}(s))\) such that \((x_k^-, s) \in \mathcal{N}\) for any \(k \in \mathbb{N}\) and \(\lim_{k \to +\infty} x_k^- = \bar{x}(s)\). By Eq. (94), we have

\[
 x_k^+ p_j - s \theta_j - \gamma_j \geq x_k^- p_i - s \theta_i - \gamma_i \quad \text{for any } i \in \{1, \ldots, m\}.
\]

When \(k\) approaches infinity, the above inequality implies

\[
 \bar{x}(s) p_j - s \theta_j - \gamma_j \geq \bar{x}(s) p_i - s \theta_i - \gamma_i \quad \text{for any } i \in \{1, \ldots, m\}.
\]

In other words, we have

\[
 j \in \arg\max_{i \in \{1, \ldots, m\}} \{\bar{x}(s) p_i - s \theta_i - \gamma_i\}.
\]

Similarly, define a sequence \(\{x_k^+\}_{k=1}^{\infty} \subset (\bar{x}(s), +\infty)\) such that \((x_k^+, s) \in \mathcal{N}\) for any \(k \in \mathbb{N}\) and \(\lim_{k \to +\infty} x_k^+ = \bar{x}(s)\). Using a similar argument as above, we can conclude that

\[
 l \in \arg\max_{i \in \{1, \ldots, m\}} \{\bar{x}(s) p_i - s \theta_i - \gamma_i\}.
\]

By a continuity argument, Eqs. (95) and (96) also hold for the end points of \(\bar{x}\). In conclusion, for any \((\bar{x}(t), t)\) on the curve \(\bar{x}\), we have

\[
 j, l \in \arg\max_{i \in \{1, \ldots, m\}} \{\bar{x}(t) p_i - t \theta_i - \gamma_i\},
\]

which implies that

\[
 \bar{x}(t) p_l - t \theta_l - \gamma_l = \bar{x}(t) p_j - t \theta_j - \gamma_j.
\]

Therefore, the curve \(\bar{x}(t)\) lies on the straight line

\[
 x(p_l - p_j) - t(\theta_l - \theta_j) - (\gamma_l - \gamma_j) = 0
\]
and Eq. (93) and Lem. 3.2 (iii) imply that its slope equals

\[ \frac{dx}{dt} = \frac{\theta_l - \theta_j}{p_l - p_j} = \frac{H(u^+) - H(u^-)}{u^+ - u^-}. \]

This proves Eq. (91) and the first condition holds.

It remains to show the second condition. Since \( u \) equals \( \nabla_x f \) and \( f \) is convex by Thm. 3.1 its corresponding subdifferential operator \( u \) is monotone non-decreasing with respect to \( x \) Def. IV.4.1.3 and Prop. VI.6.1.1. As a result, \( u^- < u^+ \) and \( u_0 \in (u^-, u^+) \), where we still adopt the notation \( u^- = p_j \) and \( u^+ = p_l \). Recall that Lem. 3.2 (iii) implies \( H(p_i) = \theta_i \) for any \( i \). Then, Eq. (92) in the second condition becomes

\[ \frac{\theta_l - H(u_0)}{p_l - u_0} \leq \frac{\theta_l - \theta_j}{p_l - p_j}. \]

Without loss of generality, we may assume that \( p_1 < p_2 < \cdots < p_m \). Then the fact \( p_j = u^- < u^+ = p_l \) implies \( j < l \). We consider the following two cases.

First, if there exists some \( k \) such that \( u_0 = p_k \), then \( H(u_0) = \theta_k \) by Lem. 3.2 (iii). Since \( u^- < u_0 < u^+ \), we have \( j < k < l \). Recall that Eq. (97) holds. Therefore the assumptions of Lem. B.3 are satisfied, which implies Eq. (98) holds.

Second, suppose \( u_0 \neq p_i \) for every \( i \in \{1, \ldots, m\} \). Then there exists some \( k \in \{j, j+1, \ldots, l-1\} \) such that \( p_k < u_0 < p_{k+1} \). Lem. B.3 then implies that Eqs. (72) and (73) hold, that is,

\[ H(u_0) = \beta_k \theta_k + \beta_{k+1} \theta_{k+1}, \quad u_0 = \beta_k p_k + \beta_{k+1} p_{k+1}, \quad \text{and} \quad \beta_k + \beta_{k+1} = 1. \]

Using these three equations, we can write the left hand side of Eq. (98) as

\[ \frac{\theta_l - H(u_0)}{p_l - u_0} = \frac{\theta_l - \beta_k \theta_k - \beta_{k+1} \theta_{k+1}}{p_l - \beta_k p_k - \beta_{k+1} p_{k+1}} = \frac{\beta_k (\theta_l - \theta_k) + \beta_{k+1} (\theta_l - \theta_{k+1})}{\beta_k (p_l - p_k) + \beta_{k+1} (p_l - p_{k+1})}. \]

If \( k = l \), then this equation becomes

\[ \frac{\theta_l - H(u_0)}{p_l - u_0} = \frac{\theta_l - \theta_k}{p_l - p_k}. \]

Since \( j \leq k < l \) and Eq. (97) holds, then the assumptions of Lem. B.3 are satisfied. This allows us to conclude that Eq. (98) holds.

If \( k + 1 \neq l \), then using Eq. (97), the inequalities \( j \leq k < k + 1 < l \), and Lem. B.3 we obtain

\[ \frac{\beta_k (\theta_l - \theta_k)}{\beta_k (p_l - p_k) + \beta_{k+1} (p_l - p_{k+1})} \leq \frac{\theta_l - \theta_j}{p_l - p_j} \]

Note that if \( a_i \in \mathbb{R} \) and \( b_i \in (0, +\infty) \) for \( i \in \{1, 2, 3\} \) satisfy \( \frac{a_1}{b_1} \leq \frac{a_2}{b_2} \) and \( \frac{a_2}{b_2} \leq \frac{a_3}{b_3} \), then \( \frac{a_1 + a_3}{b_1 + b_3} \leq \frac{a_2}{b_2} \). Then, since \( \beta_k(p_l - p_k), \beta_{k+1}(p_l - p_{k+1}) \) and \( p_l - p_j \) are positive, we have

\[ \frac{\beta_k (\theta_l - \theta_k) + \beta_{k+1} (\theta_l - \theta_{k+1})}{\beta_k(p_l - p_k) + \beta_{k+1}(p_l - p_{k+1})} \leq \frac{\theta_l - \theta_j}{p_l - p_j}. \]

Hence Eq. (98) follows directly from the inequality above and Eq. (99).

Therefore, the two conditions, including Eqs. (91) and (92), are satisfied and we apply [32] Prop 2.1 to conclude that the function \( u \) is the entropy solution to the conservation law [32].

Proof of (ii) (sufficiency): Without loss of generality, assume \( p_1 < p_2 < \cdots < p_m \). Let \( C \in \mathbb{R} \). Suppose \( \tilde{H} \) satisfies \( \tilde{H}(p_i) = H(p_i) + C \) for each \( i \in \{1, \ldots, m\} \) and \( \tilde{H}(p) \geq H(p) + C \) for any \( p \in [p_1, p_m] \). We want to prove that \( u \) is the entropy solution to the conservation law [33].

As in the proof of (i), we apply [32] Prop 2.1 and [32] and (92). Let \( \tilde{x}(t) \) be any smooth line of discontinuity of \( u \), define \( u^- \) and \( u^+ \) by Eq. (100) (and recall that \( u^- = p_j \) and \( u^+ = p_l \)), and let \( u_0 \in (u^-, u^+) \). We proved in the proof of (i) that \( \tilde{x}(t) \) is a straight line, and so it suffices to prove that

\[ \frac{dx}{dt} = \frac{H(u^+) - H(u^-)}{u^+ - u^-}, \quad \text{and} \quad \frac{H(u^+) - H(u_0)}{u^+ - u_0} \leq \frac{H(u^+) - H(u^-)}{u^+ - u^-}. \]

We start with proving the equality in Eq. (100). By assumption, there holds

\[ \tilde{H}(u^-) = \tilde{H}(p_j) = H(p_j) + C = H(u^-) + C \quad \text{and} \quad \tilde{H}(u^+) = \tilde{H}(p_l) = H(p_l) + C = H(u^+) + C. \]
We combine Eq. (101) with Eq. (91), (which we proved in the proof of (i)), we obtain
\[
\frac{d\tilde{x}}{dt} = H(u^+) - H(u^-) = \frac{H(u^+) + C - (H(u^-) + C)}{u^+ - u^-} = \frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-}.
\]

Therefore, the equality in (100) holds.

Next, we prove the inequality in Eq. (100). Since \(u_0 \in (u^-, u^+) \subseteq [p_1, p_m]\), by assumption there holds \(\tilde{H}(u_0) \geq H(u_0) + C\). Taken together with Eqs. (92) and (101), we get
\[
\frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-} \leq \frac{H(u^+) - H(u^-)}{u^+ - u^-} \leq \frac{H(u^+) - H(u^-)}{u^+ - u^-} = \frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-},
\]
which shows that the inequality in Eq. (100) holds.

Hence, we can invoke [32, Prop 2.1] to conclude that \(u\) is the entropy solution to the conservation law (33).

Proof of (ii) (necessity): Suppose that \(u\) is the entropy solution to the conservation law (33). We prove that there exists \(C \in \mathbb{R}\) such that \(\tilde{H}(p_i) = H(p_i) + C\) for any \(i\) and \(\tilde{H}(p) \geq H(p) + C\) for any \(p \in [p_1, p_m]\).

By Lem. 3.4 for each \(i \in \{1, \ldots, m\}\) there exist \(x \in \mathbb{R}\) and \(t > 0\) such that
\[
(102) \quad f(\cdot, t) \text{ is differentiable at } x, \text{ and } \nabla_x f(x, t) = p_i.
\]
Moreover, the proof of Lem. 3.4 implies there exists \(T > 0\) such that for any \(0 < t < T\), there exists \(x \in \mathbb{R}\), such that Eq. (102) holds. As a result, there exists \(t > 0\) such that for each \(i \in \{1, \ldots, m\}\), there exists \(x_i \in \mathbb{R}\) satisfying Eq. (102) at the point \((x_i, t)\), which implies \(u(x_i, t) = p_i\). Note that \(p_i \neq p_j\) implies that \(x_i \neq x_j\). (Indeed, if \(x_i = x_j\), then \(p_i = \nabla_x f(x_i, t) = \nabla_x f(x_j, t) = p_j\) which gives a contradiction since \(p_i \neq p_j\) by assumption (A1).) As mentioned before, the function \(u(\cdot, t) \equiv \nabla_x f\) is a monotone non-decreasing operator and \(p_i\) is increasing with respect to \(i\), and therefore \(x_1 < x_2 < \cdots < x_m\). Since \(u\) is piecewise constant, for each \(k \in \{1, \ldots, m - 1\}\) there exists a curve of discontinuity of \(u\) with \(u = p_k\) on the left hand side of the curve and \(u = p_{k+1}\) on the right hand side of the curve. Let \(\tilde{x}(s)\) be such a curve and let \(u^-\) and \(u^+\) be the corresponding numbers defined in Eq. (90). The argument above proves that we have \(u^- = p_k\) and \(u^+ = p_{k+1}\).

Since \(u\) is the piecewise constant entropy solution, we invoke [32, Prop 2.1] to conclude that the two aforementioned conditions hold for the curve \(\tilde{x}(s)\), i.e., (100) holds with \(u^- = p_k\) and \(u^+ = p_{k+1}\). From the equality in (100) and Eq. (91) proved in (i), we deduce
\[
\frac{\tilde{H}(p_{k+1}) - \tilde{H}(p_k)}{p_{k+1} - p_k} = \frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-} = \frac{d\tilde{x}}{dt} = \frac{H(u^+) - H(u^-)}{u^+ - u^-} = \frac{H(p_{k+1}) - H(p_k)}{p_{k+1} - p_k}.
\]

Since \(k\) is an arbitrary index, the equality above implies that \(\tilde{H}(p_{k+1}) - \tilde{H}(p_k) = H(p_{k+1}) - H(p_k)\) holds for any \(k \in \{1, \ldots, m - 1\}\). Therefore, there exists \(C \in \mathbb{R}\) such that
\[
(103) \quad \tilde{H}(p_k) = H(p_k) + C \text{ for any } k \in \{1, \ldots, m\}.
\]

It remains to prove \(\tilde{H}(u_0) \geq H(u_0) + C\) for all \(u_0 \in [p_k, p_{k+1}]\). If this inequality holds, then the statement follows because \(k\) is an arbitrary index. We already proved that \(\tilde{H}(u_0) \geq H(u_0) + C\) for \(u_0 = p_k\) with \(k \in \{1, \ldots, m\}\). Therefore, we need to prove that \(\tilde{H}(u_0) \geq H(u_0) + C\) for all \(u_0 \in (p_k, p_{k+1})\). Let \(u_0 \in (p_k, p_{k+1})\). By Eq. (103) and the inequality in (100), we have
\[
(104) \quad \frac{H(p_{k+1}) + C - \tilde{H}(u_0)}{p_{k+1} - u_0} = \frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-} \leq \frac{\tilde{H}(u^+) - \tilde{H}(u^-)}{u^+ - u^-} = \frac{H(p_{k+1}) - H(p_k)}{p_{k+1} - p_k}.
\]

By Lem. 3.2 and a straightforward computation, we also have
\[
(105) \quad \frac{H(p_{k+1}) - H(u_0)}{p_{k+1} - u_0} = \frac{H(p_{k+1}) - H(p_k)}{p_{k+1} - p_k}.
\]

Comparing Eqs. (104) and (105), we obtain \(\tilde{H}(u_0) \geq H(u_0) + C\). Since \(k\) is arbitrary, we conclude that \(\tilde{H}(u_0) \geq H(u_0) + C\) holds for all \(u_0 \in [p_1, p_m]\) and the proof is complete.
APPENDIX C. PROOF OF PROPOSITION 3.2

To prove this proposition, we use the following lemma.

Lemma C.1. Let \( \{(p_i, \gamma_i)\}_{i=1}^{m} \subset \mathbb{R}^n \times \mathbb{R} \) and \( \epsilon > 0 \). Then the function \( w_{\epsilon}(x, t) : \mathbb{R}^n \to \mathbb{R} \) defined by

\[
(106) \quad w_{\epsilon}(x, t) := \sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x) + \frac{\epsilon}{2} \|p_i\|^2 - \gamma_i\right)/\epsilon}
\]

is the unique smooth solution to the Cauchy problem

\[
(107) \quad \begin{cases}
\frac{\partial w_{\epsilon}}{\partial t}(x, t) = \frac{\epsilon}{2} \Delta_x w_{\epsilon}(x, t) & \text{in } \mathbb{R}^n \times (0, +\infty), \\
w_{\epsilon}(x, 0) = \sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x) - \gamma_i\right)/\epsilon} & \forall x \in \mathbb{R}^n.
\end{cases}
\]

Moreover, \( w_{\epsilon} \) is jointly log-convex in \((x, t)\).

Proof. A short calculation shows that the function \( w_{\epsilon} \) defined in Eq. \( (106) \) solves the Cauchy problem \( (107) \), and uniqueness holds by strict positiveness of the initial data (see [129], Chap. VIII, Thm. 2.2), and note that the uniqueness result can easily be generalized to \( n > 1 \).

Now, let \( \lambda \in [0, 1] \) and \((x_1, t_1)\) and \((x_2, t_2)\) be such that \( x = \lambda x_1 + (1-\lambda)x_2 \) and \( t = \lambda t_1 + (1-\lambda)t_2 \). Then the Hölder’s inequality (see, e.g., [31], Thm. 6.2) implies

\[
\sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x) + \frac{\epsilon}{2} \|p_i\|^2 - \gamma_i\right)/\epsilon} \
\leq \left( \sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x_1) + \frac{\epsilon}{2} \|p_i\|^2 - \gamma_i\right)/\epsilon} \right)^{1-\lambda} \left( \sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x_2) + \frac{\epsilon}{2} \|p_i\|^2 - \gamma_i\right)/\epsilon} \right)^{\lambda},
\]

and we find \( w_{\epsilon}(x, t) \leq (w_{\epsilon}(x_1, t_1))^{\lambda} (w_{\epsilon}(x_2, t_2))^{1-\lambda} \), which implies that \( w_{\epsilon} \) is jointly log-convex in \((x, t)\). \( \Box \)

Proof of Proposition 3.2. By Lemma C.1, the function \( w_{\epsilon} \) solves the Cauchy problem \( (107) \). A short calculation then shows that the neural network \( f_{\epsilon} \) solves the second-order HJ equation \( (36) \), and is its unique solution because \( w_{\epsilon} \) is the unique solution to \( (107) \). Joint convexity in \((x, t)\) follows from log-convexity of \((x, t) \mapsto w_{\epsilon}(x, t)\) for every \( \epsilon > 0 \). That \( f_{\epsilon} \) satisfies the bounds \( (35) \) follow from a short calculation:

\[
\max_{i \in \{1, \ldots, m\}} \left\{ \langle p_i, x \rangle + \frac{t}{2} \|p_i\|^2 - \gamma_i \right\} = \epsilon \log \left( e^{\left(\max_{i \in \{1, \ldots, m\}} \{ \langle p_i, x \rangle + \epsilon \|p_i\|^2 - \gamma_i \}\right)/\epsilon} \right)
\leq \epsilon \log \left( \sum_{i=1}^{m} e^{\left(\epsilon(p_i \cdot x) + \frac{\epsilon}{2} \|p_i\|^2 - \gamma_i\right)/\epsilon} \right)
\leq \epsilon \log \left( m e^{\left(\max_{i \in \{1, \ldots, m\}} \{ \langle p_i, x \rangle + \epsilon \|p_i\|^2 - \gamma_i \}\right)/\epsilon} \right)
\leq \max_{i \in \{1, \ldots, m\}} \left\{ \langle p_i, x \rangle + \frac{t}{2} \|p_i\|^2 - \gamma_i \right\} + \epsilon \log(m).
\]

These bounds imply that the limit \( (37) \) holds for every \( x \in \mathbb{R}^n \) and \( t \geq 0 \). Finally, under assumptions (A1)–(A3) the right hand side of this limit satisfies the HJ equation \( (16) \) with \( \bar{H} = -\frac{1}{2} \|\cdot\|^2 \) as all conditions in Thm. 3.1 are satisfied. This concludes the proof.

References

[1] Aaibid, M., Sayah, A.: A direct proof of the equivalence between the entropy solutions of conservation laws and viscosity solutions of Hamilton-Jacobi equations in one-space variable. JIPAM. Journal of Inequalities in Pure and Applied Mathematics 7(2), Article 64, 11 (2006) (Cited on page 10)
[2] Akian, M., Bapat, R., Gaubert, S.: Max-plus algebra. Handbook of linear algebra 39 (2006) (Cited on page 3)
[3] Akian, M., Gaubert, S., Lakhouna, A.: The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. SIAM Journal on Control and Optimization 47(2), 817–848 (2008) (Cited on page 4)
[4] Alla, A., Falcone, M., Saluzzi, L.: An efficient DP algorithm on a tree-structure for finite horizon optimal control problems. SIAM Journal on Scientific Computing 41(4), A2384–A2406 (2019) (Cited on page [4])

[5] Alla, A., Falcone, M., Volkwein, S.: Error analysis for POD approximations of infinite horizon problems via the dynamic programming approach. SIAM Journal on Control and Optimization 55(5), 3091–3115 (2017) (Cited on page [4])

[6]Arnol’d, V.I.: Mathematical methods of classical mechanics, Graduate Texts in Mathematics, vol. 60. Springer-Verlag, New York (1989). Translated from the 1974 Russian original by K. Vogtmann and A. Weinstein, Corrected reprint of the second (1989) edition (Cited on page [6])

[7] Bachouch, A., Huré, C., Langrené, N., Pham, H.: Deep neural networks algorithms for stochastic control problems on finite horizon: numerical applications. arXiv preprint arXiv:1812.05916 (2018) (Cited on page [2])

[8] Bardi, M., Capuzzo-Dolcetta, I.: Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Systems & Control: Foundations & Applications. Birkhäuser Boston, Inc., Boston, MA (1997). DOI 10.1007/978-0-8176-4755-1. With appendices by Maurizio Falcone and Pierpaolo Soravia (Cited on page [1])

[9] Bardi, M., Evans, L.: On Hopf’s formulas for solutions of Hamilton-Jacobi equations. Nonlinear Analysis: Theory, Methods & Applications 8(11), 1373 – 1381 (1984). DOI 10.1016/0362-546X(84)90020-8 (Cited on pages [1] and [8])

[10] Barles, G.: Solutions de viscosité des équations de Hamilton-Jacobi. Mathématiques et Applications. Springer-Verlag Berlin Heidelberg (1994) (Cited on page [1])

[11] Barles, G., Tourin, A.: Commutation properties of semigroups for first-order Hamilton-Jacobi equations and application to multi-time equations. Indiana University Mathematics Journal 50(4), 1523–1544 (2001). URL http://www.jstor.org/stable/24902783 (Cited on page [19])

[12] Barron, E., Evans, L., Jensen, R.: Viscosity solutions of Isaacs’ equations and differential games with Lipschitz controls. Journal of Differential Equations 53(2), 213 – 233 (1984). DOI 10.1016/0022-0396(84)90040-8 (Cited on page [1])

[13] Beck, C., Becker, S., Cheridito, P., Jentzen, A., Neufeld, A.: Deep splitting method for parabolic pdes. arXiv preprint arXiv:1907.03452 (2019) (Cited on page [2])

[14] Beck, C., Becker, S., Grohs, P., Jaafari, N., Jentzen, A.: Solving stochastic differential equations and kolmogorov equations by means of deep learning. arXiv preprint arXiv:1806.00421 (2018) (Cited on page [2])

[15] Beck, C., Weinan, E., Jentzen, A.: Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. Journal of Nonlinear Science 29(4), 1563–1619 (2019) (Cited on page [2])

[16] Bellman, R.E.: Adaptive control processes: a guided tour. Princeton university press (1961) (Cited on page [1])

[17] Berg, J., Nystrm, K.: A unified deep artificial neural network approach to partial differential equations in complex geometries. Neurocomputing 317, 28 – 41 (2018). DOI 10.1016/j.neucom.2018.06.056 (Cited on page [2])

[18] Bertsekas, D.P.: Reinforcement learning and optimal control. Athena Scientific, Belmont, Massachusetts (2019) (Cited on page [1])

[19] Bokanowski, O., Garcke, J., Griebel, M., Klompmaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. Journal of Scientific Computing 55(3), 575–605 (2013) (Cited on page [1])

[20] Bonnans, J.F., Shapiro, A.: Perturbation analysis of optimization problems. Springer Series in Operations Research. Springer-Verlag, New York (2000). DOI 10.1007/978-1-4612-1394-9 (Cited on page [8])

[21] Buckdahn, R., Cardaliaguet, P., Quincampoix, M.: Some recent aspects of differential game theory. Dynamic Games and Applications 1(4), 74–114 (2011). DOI 10.1007/s13235-010-0005-0 (Cited on page [7])

[22] Carathéodory, C.: Calculus of variations and partial differential equations of the first order. Part I: Partial differential equations of the first order. Translated by Robert B. Dean and Julius J. Brandstatter. Holden-Day, Inc., San Francisco-London-Amsterdam (1965) (Cited on page [1])
[23] Carathéodory, C.: Calculus of variations and partial differential equations of the first order. Part II: Calculus of variations. Translated from the German by Robert B. Dean, Julius J. Brandstatter, translating editor. Holden-Day, Inc., San Francisco-London-Amsterdam (1967) (Cited on page 1)

[24] Cardin, F., Viterbo, C.: Commuting Hamiltonians and Hamilton-Jacobi multi-time equations. Duke Math. J. 144(2), 235–284 (2008). DOI 10.1215/00127094-2008-036 (Cited on page 19)

[25] Caselles, V.: Scalar conservation laws and Hamilton-Jacobi equations in one-space variable. Nonlinear Analysis: Theory, Methods & Applications 18(5), 461 – 469 (1992). DOI 10.1016/0362-546X(92)90113-5 (Cited on page 10)

[26] Chan-Wai-Nam, Q., Mikael, J., Warin, X.: Machine learning for semi linear pdes. Journal of Scientific Computing 79(3), 1667–1712 (2019) (Cited on page 2)

[27] Corrias, L., Falcone, M., Natalini, R.: Numerical schemes for conservation laws via Hamilton-Jacobi equations. Bulletin of the American mathematical society 27(1), 1–67 (1992). DOI 10.1090/S0273-0979-1992-00266-5 (Cited on page 1)

[28] Courant, R., Hilbert, D.: Methods of mathematical physics. Vol. II. Wiley Classics Library. John Wiley & Sons, Inc., New York (1989). Partial differential equations, Reprint of the 1962 original, A Wiley-Interscience Publication (Cited on page 1)

[29] Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. Bulletin of the American mathematical society 27(1), 1–67 (1992). DOI 10.1090/S0273-0979-1992-00266-5 (Cited on page 1)

[30] Dafermos, C.M.: Hyperbolic conservation laws in continuum physics, Grundlehren der Mathematischen Wissenschaften, vol. 325, fourth edn. Springer-Verlag, Berlin (2016). DOI 10.1007/978-3-662-49451-6 (Cited on page 10)

[31] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2(4), 303–314 (1989). DOI 10.1007/BF02551274 (Cited on page 2)

[32] Dafermos, C.M.: Polyhedral approximations of solutions of the initial value problem for a conservation law. Journal of Mathematical Analysis and Applications 38(1), 33 – 41 (1972). DOI 10.1016/0022-247X(72)90114-X (Cited on pages 32, 33 and 34)

[33] Dafermos, C.M.: Hyperbolic conservation laws in continuum physics, Grundlehren der Mathematischen Wissenschaften, vol. 325, fourth edn. Springer-Verlag, Berlin (2016). DOI 10.1007/978-3-662-49451-6 (Cited on page 10)

[34] Darbon, J.: On convex finite-dimensional variational methods in imaging sciences and Hamilton–Jacobi equations. SIAM Journal on Imaging Sciences 8(4), 2268–2293 (2015). DOI 10.1137/130944163 (Cited on pages 1, 2, 19 and 21)

[35] Darbon, J., Meng, T.: On decomposition models in imaging sciences and multi-time Hamilton-Jacobi partial differential equations. arXiv preprint arXiv:1906.09502 (2019) (Cited on pages 1, 2, 19, and 22)

[36] Darbon, J., Osher, S.: Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere. Research in the Mathematical Sciences 3(1), 19 (2016). DOI 10.1186/s40687-016-0068-7 (Cited on pages 1 and 2)

[37] Dissanayake, M.W.M.G., Phan-Thien, N.: Neural-network-based approximations for solving partial differential equations. Communications in Numerical Methods in Engineering 10(3), 195–201 (1994). DOI 10.1002/cnm.1640100303 (Cited on pages 1 and 2)

[38] Djeridane, B., Lygeros, J.: Neural approximation of PDE solutions: An application to reachability computations. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 3034–3039 (2006). DOI 10.1109/CDC.2006.377184 (Cited on page 2)

[39] Dockhorn, T.: A discussion on solving partial differential equations using neural networks. arXiv preprint arXiv:1904.07200 (2019) (Cited on page 2)

[40] Dolgov, S., Kalise, D., Kunisch, K.: A tensor decomposition approach for high-dimensional Hamilton-Jacobi-Bellman equations. arXiv preprint arXiv:1908.01533 (2019) (Cited on page 1)

[41] Dower, P.M., McEneaney, W.M., Zhang, H.: Max-plus fundamental solution semigroups for optimal control problems. In: 2015 Proceedings of the Conference on Control and its Applications, pp. 368–375. SIAM (2015) (Cited on page 1)
[42] E, W., Han, J., Jentzen, A.: Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics 5(4), 349–380 (2017). DOI 10.1007/s40304-017-0117-6 (Cited on page 2).

[43] Elliott, R.J.: Viscosity solutions and optimal control, Pitman Research Notes in Mathematics Series, vol. 165. Longman Scientific & Technical, Harlow; John Wiley & Sons, Inc., New York (1987) (Cited on page 1).

[44] Evans, L.C.: Partial differential equations, Graduate Studies in Mathematics, vol. 19, second edn. American Mathematical Society, Providence, RI (2010). DOI 10.1090/gsm/019 (Cited on page 2).

[45] Evans, L.C., Gariepy, R.F.: Measure theory and fine properties of functions, revised edn. Textbooks in Mathematics. CRC Press, Boca Raton, FL (2015) (Cited on page 32)

[46] Evans, L.C., Souganidis, P.E.: Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. Indiana University Mathematics Journal 33(5), 773–797 (1984) (Cited on page 1).

[47] Farimani, A.B., Gomes, J., Pande, V.S.: Deep Learning the Physics of Transport Phenomena. arXiv e-prints (2017) (Cited on page 2).

[48] Fleming, W., McEneaney, W.: A max-plus-based algorithm for a Hamilton–Jacobi–Bellman equation of nonlinear filtering. SIAM Journal on Control and Optimization 38(3), 683–710 (2000). DOI 10.1137/S036301299832433 (Cited on page 1).

[49] Fleming, W.H., Rishel, R.W.: Deterministic and stochastic optimal control. Bulletin of the American Mathematical Society 82, 869–870 (1976) (Cited on page 1).

[50] Fleming, W.H., Soner, H.M.: Controlled Markov processes and viscosity solutions, vol. 25. Springer Science & Business Media (2006) (Cited on page 1).

[51] Folland, G.B.: Real analysis: modern techniques and their applications. John Wiley & Sons (2013) (Cited on pages 8, 32, and 35).

[52] Fujii, M., Takahashi, A., Takahashi, M.: Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. Asia-Pacific Financial Markets 26(3), 391–408 (2019). DOI 10.1007/s10690-019-09271-7 (Cited on page 2).

[53] Garcke, J., Kröner, A.: Suboptimal feedback control of PDEs by solving HJB equations on adaptive sparse grids. Journal of Scientific Computing 70(1), 1–28 (2017) (Cited on page 1).

[54] Gaubert, S., McEneaney, W., Qu, Z.: Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference, pp. 1054–1061. IEEE (2011) (Cited on page 1).

[55] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org (Cited on page 3).

[56] Grohs, P., Jentzen, A., Salimova, D.: Deep neural network approximations for monte carlo algorithms. arXiv preprint arXiv:1908.10828 (2019) (Cited on page 2).

[57] Han, J., Jentzen, A., E, W.: Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences 115(34), 8505–8510 (2018). DOI 10.1073/pnas.1718942115 (Cited on page 2).

[58] Han, J., Zhang, L., Weinan, E.: Solving many-electron schrödinger equation using deep neural networks. Journal of Computational Physics p. 108929 (2019) (Cited on page 2).

[59] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). DOI 10.1109/CVPR.2016.90 (Cited on page 21).

[60] Hiriart-Urruty, J.B., Lemaréchal, C.: Convex analysis and minimization algorithms I: Fundamentals, vol. 305. Springer science & business media (1993) (Cited on pages 3, 8, 23, 27, 31, and 33).

[61] Hiriart-Urruty, J.B., Lemaréchal, C.: Convex analysis and minimization algorithms II: Advanced Theory and Bundle Methods, vol. 306. Springer science & business media (1993) (Cited on pages 9, 14, 19, 22, 25, and 26).

[62] Hopf, E.: Generalized solutions of non-linear equations of first order. J. Math. Mech. 14, 951–973 (1965) (Cited on pages 15 and 21).

[63] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Networks 4(2), 251–257 (1991). DOI 10.1016/0893-6080(91)90009-T (Cited on page 2).
39

[64] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359 – 366 (1989). DOI 10.1016/0893-6080(89)90020-8 (Cited on page 2)

[65] Horowitz, M.B., Damle, A., Burdick, J.W.: Linear Hamilton Jacobi Bellman equations in high dimensions. In: 53rd IEEE Conference on Decision and Control, pp. 5880–5887. IEEE (2014) (Cited on page 1)

[66] Hsieh, J.T., Zhao, S., Eismann, S., Mirabella, L., Ermon, S.: Learning neural PDE solvers with convergence guarantees. In: International Conference on Learning Representations (2019) (Cited on page 2)

[67] Hu, C., Shu, C.: A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. SIAM Journal on Scientific Computing 21(2), 666–690 (1999). DOI 10.1137/S1064827598337282 (Cited on page 1)

[68] Huré, C., Pham, H., Bachouch, A., Langrené, N.: Deep neural networks algorithms for stochastic control problems on finite horizon, part i: convergence analysis. arXiv preprint arXiv:1812.04300 (2018) (Cited on page 2)

[69] Huré, C., Pham, H., Warin, X.: Some machine learning schemes for high-dimensional nonlinear pdes. arXiv preprint arXiv:1902.01599 (2019) (Cited on page 2)

[70] Ishii, H.: Representation of solutions of Hamilton-Jacobi equations. Nonlinear Analysis: Theory, Methods & Applications 12(2), 121 – 146 (1988). DOI 10.1016/0362-546X(88)90030-2 (Cited on page 1)

[71] Jiang, F., Chou, G., Chen, M., Tomlin, C.J.: Using neural networks to compute approximate and guaranteed feasible hamilton-jacobi-bellman pde solutions. arXiv preprint arXiv:1611.03158 (2016) (Cited on page 2)

[72] Jiang, G., Peng, D.: Weighted ENO schemes for Hamilton–Jacobi equations. SIAM Journal on Scientific Computing 21(6), 2126–2143 (2000). DOI 10.1137/S106482759732455X (Cited on page 1)

[73] Jianyu, L., Siwei, L., Yingjian, Q., Yaping, H.: Numerical solution of elliptic partial differential equation using radial basis function neural networks. Neural Networks 16(5-6), 729–734 (2003) (Cited on page 2)

[74] Jin, S., Xin, Z.: Numerical passage from systems of conservation laws to Hamilton-Jacobi equations, and relaxation schemes. SIAM Journal on Numerical Analysis 35(6), 2385–2404 (1998). DOI 10.1137/S0036142996314366 (Cited on page 1)

[75] Kalise, D., Kundu, S., Kunisch, K.: Robust feedback control of nonlinear PDEs by numerical approximation of high-dimensional Hamilton-Jacobi-Isaacs equations. arXiv preprint arXiv:1905.06276 (2019) (Cited on page 2)

[76] Kalise, D., Kunisch, K.: Polynomial approximation of high-dimensional Hamilton–Jacobi–Bellman equations and applications to feedback control of semilinear parabolic PDEs. SIAM Journal on Scientific Computing 40(2), A629–A652 (2018) (Cited on page 2)

[77] Kang, W., Wilcox, L.C.: Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. Computational Optimization and Applications 68(2), 289–315 (2017) (Cited on page 1)

[78] Karlsen, K., Risebro, H.: A note on front tracking and the equivalence between viscosity solutions of hamilton-jacobi equations and entropy solutions of scalar conservation laws. Nonlinear Analysis 50 (2002). DOI 10.1016/S0362-546X(01)00753-2 (Cited on page 10)

[79] Khoo, Y., Lu, J., Ying, L.: Solving parametric pde problems with artificial neural networks. arXiv preprint arXiv:1707.03531 (2017) (Cited on page 2)

[80] Khoo, Y., Lu, J., Ying, L.: Solving for high-dimensional committor functions using artificial neural networks. Research in the Mathematical Sciences 6(1), 1 (2019) (Cited on page 2)

[81] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015) (2015) (Cited on page 5)

[82] Kružkov, S.N.: Generalized solutions of nonlinear first order equations with several independent variables. II. Mathematics of the USSR-Sbornik 1(1), 93–116 (1967). DOI 10.1070/sm1967v001n01abeh001969 (Cited on page 10)

[83] Kunisch, K., Volkwein, S., Xie, L.: HJB-POD-based feedback design for the optimal control of evolution problems. SIAM Journal on Applied Dynamical Systems 3(4), 701–722 (2004) (Cited on page 1)
Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks 9(5), 987–1000 (1998). DOI 10.1109/72.712178 (Cited on page 2)

Lagaris, I.E., Likas, A.C., Papageorgiou, D.G.: Neural-network methods for boundary value problems with irregular boundaries. IEEE Transactions on Neural Networks 11(5), 1041–1049 (2000). DOI 10.1109/72.780037 (Cited on page 2)

Lambrianides, P., Gong, Q., Venturi, D.: A new scalable algorithm for computational optimal control under uncertainty. arXiv preprint arXiv:1909.07960 (2019) (Cited on page 2)

Landau, L., Lifschic, E.: Course of theoretical physics. vol. 1: Mechanics. Oxford (1978) (Cited on page 3)

LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015) (Cited on page 3)

Lee, H., Kang, I.S.: Neural algorithm for solving differential equations. Journal of Computational Physics 91(1), 110–131 (1990) (Cited on page 2).

Lions, P.L., Rochet, J.C.: Hopf formula and multitime Hamilton-Jacobi equations. Proceedings of the American Mathematical Society 96(1), 79–84 (1986). URL http://www.jstor.org/stable/2045587 (Cited on page 19)

Lions, P.L., Souganidis, P.E.: Convergence of MUSCL and filtered schemes for scalar conservation laws and Hamilton-Jacobi equations. Numerische Mathematik 69(4), 441–470 (1995). DOI 10.1007/s002110050102 (Cited on page 10)

Long, Z., Lu, Y., Dong, B.: Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. Journal of Computational Physics 399, 108925 (2019). DOI 10.1016/j.jcp.2019.108925 (Cited on page 2)

Long, Z., Lu, Y., Ma, X., Dong, B.: Pde-net: Learning pdes from data. arXiv preprint arXiv:1710.09668 (2017) (Cited on page 2)

Lye, K.O., Mishra, S., Ray, D.: Deep learning observables in computational fluid dynamics. arXiv preprint arXiv:1903.03040 (2019) (Cited on page 2)

McEneaney, W.: Max-plus methods for nonlinear control and estimation. Springer Science & Business Media (2006) (Cited on page 3)

McEneaney, W.: A curse-of-dimensionality-free numerical method for solution of certain HJB PDEs. SIAM Journal on Control and Optimization 46(4), 1239–1276 (2007). DOI 10.1137/040610830 (Cited on page 1)

McEneaney, W.M., Deshpande, A., Gaubert, S.: Curve-of-complexity attenuation in the curve-of-dimensionality-free method for HJB PDEs. In: 2008 American Control Conference, pp. 4684–4690. IEEE (2008) (Cited on page 4)

McEneaney, W.M., Kluberg, L.J.: Convergence rate for a curve-of-dimensionality-free method for a class of hjb pdes. SIAM Journal on Control and Optimization 48(5), 3052–3079 (2009) (Cited on page 4)

McFall, K.S., Mahan, J.R.: Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions. IEEE Transactions on Neural Networks 20(8), 1221–1233 (2009). DOI 10.1109/TNN.2009.2020735 (Cited on page 2)

Meade, A., Fernandez, A.: The numerical solution of linear ordinary differential equations by feed-forward neural networks. Mathematical and Computer Modelling 19(12), 1–25 (1994). DOI 10.1016/0895-7177(94)00095-7 (Cited on page 2)

Meng, X., Karniadakis, G.E.: A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. arXiv preprint arXiv:1903.00104 (2019) (Cited on page 2)

Meng, X., Li, Z., Zhang, D., Karniadakis, G.E.: Ppinn: Parareal physics-informed neural network for time-dependent pdes. arXiv preprint arXiv:1909.10145 (2019) (Cited on page 2)

van Milligen, B.P., Tribaldos, V., Jiménez, J.A.: Neural network differential equation and plasma equilibrium solver. Phys. Rev. Lett. 75, 3594–3597 (1995). DOI 10.1103/PhysRevLett.75.3594 (Cited on page 2)

Motta, M., Rampazzo, F.: Nonsmooth multi-time Hamilton-Jacobi systems. Indiana University Mathematics Journal 55(5), 1573–1614 (2006). URL http://www.jstor.org/stable/24902657 (Cited on page 2)
[105] Niarchos, K.N., Lygeros, J.: A neural approximation to continuous time reachability computations. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 6313–6318 (2006). DOI 10.1109/CDC.2006.377358 (Cited on page 2)

[106] Osher, S., Shu, C.: High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. SIAM Journal on Numerical Analysis 28(4), 907–922 (1991). DOI 10.1137/0728049 (Cited on page 1)

[107] Pang, G., Lu, L., Karniadakis, G.E.: fpinns: Fractional physics-informed neural networks. SIAM Journal on Scientific Computing 41(4), A2603–A2626 (2019) (Cited on page 2)

[108] Pham, H., Pham, H., Warin, X.: Neural networks-based backward scheme for fully nonlinear pdes. arXiv preprint arXiv:1908.00412 (2019) (Cited on page 2)

[109] Pinkus, A.: Approximation theory of the MLP model in neural networks. In: Acta numerica, 1999, Acta Numer., vol. 8, pp. 143–195. Cambridge Univ. Press, Cambridge (1999) (Cited on page 2)

[110] Plaskacz, S., Quincampoix, M.: Oleinik-Lax formulas and multitime Hamilton-Jacobi systems. Nonlinear Analysis: Theory, Methods & Applications 51(6), 957–967 (2002). DOI 10.1016/S0362-546X(01)00871-9 (Cited on page 2)

[111] Raissi, M.: Deep hidden physics models: Deep learning of nonlinear partial differential equations. The Journal of Machine Learning Research 19(1), 932–955 (2018) (Cited on page 2)

[112] Raissi, M.: Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations. arXiv preprint arXiv:1804.07010 (2018) (Cited on page 2)

[113] Raissi, M., Perdikaris, P., Karniadakis, G.: 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). DOI 10.1016/j.jcp.2018.10.045 (Cited on page 2)

[114] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561 (2017) (Cited on page 2)

[115] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566 (2017) (Cited on page 2)

[116] Reisinger, C., Zhang, Y.: Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems. arXiv preprint arXiv:1903.06652 (2019) (Cited on page 2)

[117] Rochet, J.: The taxation principle and multi-time Hamilton-Jacobi equations. Journal of Mathematical Economics 14(2), 113 – 128 (1985). DOI 10.1016/0304-4068(85)90015-1 (Cited on page 19)

[118] Rockafellar, R.T.: Convex analysis. Princeton university press (1970) (Cited on pages 3, 4, 8, 22)

[119] Royo, V.R., Tomlin, C.: Recursive regression with neural networks: Approximating the hjpde solution. arXiv preprint arXiv:1611.02739 (2016) (Cited on page 2)

[120] Rudd, K., Muro, G.D., Ferrari, S.: A constrained backpropagation approach for the adaptive solution of partial differential equations. IEEE Transactions on Neural Networks and Learning Systems 25(3), 571–584 (2014). DOI 10.1109/TNNLS.2013.2277601 (Cited on page 2)

[121] Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks 61, 85 – 117 (2015). DOI 10.1016/j.neunet.2014.09.003 (Cited on page 3)

[122] Sirignano, J., Spiliopoulos, K.: DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics 375, 1339 – 1364 (2018). DOI 10.1016/j.jcp.2018.08.029 (Cited on page 2)

[123] Tang, W., Shan, T., Dang, X., Li, M., Yang, F., Xu, S., Wu, J.: Study on a Poisson’s equation solver based on deep learning technique. In: 2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), pp. 1–3 (2017). DOI 10.1109/EDAPS.2017.8277017 (Cited on page 2)

[124] Tassa, Y., Erez, T.: Least squares solutions of the HJB equation with neural network value-function approximators. IEEE Transactions on Neural Networks 10(4), 1031–1041 (2007). DOI 10.1109/TNN.2007.9099249 (Cited on page 2)

[125] Tho, N.: Hopf-Lax-Oleinik type formula for multi-time Hamilton-Jacobi equations. Acta Math. Vietnamica 30, 275–287 (2005) (Cited on page 19)
[126] Todorov, E.: Efficient computation of optimal actions. Proceedings of the national academy of sciences 106(28), 11478–11483 (2009) (Cited on page 1)
[127] Uchiyama, T., Sonehara, N.: Solving inverse problems in nonlinear pdes by recurrent neural networks. In: IEEE International Conference on Neural Networks, pp. 99–102. IEEE (1993) (Cited on page 2)
[128] Weinan, E., Yu, B.: The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics 6(1), 1–12 (2018) (Cited on page 2)
[129] Widder, D.V.: The heat equation, vol. 67. Academic Press (1976) (Cited on page 35)
[130] Yadav, N., Yadav, A., Kumar, M.: An introduction to neural network methods for differential equations. SpringerBriefs in Applied Sciences and Technology. Springer, Dordrecht (2015). DOI 10.1007/978-94-017-9816-7 (Cited on page 2)
[131] Yang, L., Zhang, D., Karniadakis, G.E.: Physics-informed generative adversarial networks for stochastic differential equations. arXiv preprint arXiv:1811.02033 (2018) (Cited on page 2)
[132] Yang, Y., Perdikaris, P.: Adversarial uncertainty quantification in physics-informed neural networks. Journal of Computational Physics 394, 136–152 (2019) (Cited on page 2)
[133] Yegorov, I., Dower, P.M.: Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton–Jacobi equations. Applied Mathematics & Optimization pp. 1–49 (2017) (Cited on page 1)
[134] Zhang, D., Guo, L., Karniadakis, G.E.: Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks. arXiv preprint arXiv:1905.01205 (2019) (Cited on page 2)
[135] Zhang, D., Lu, L., Guo, L., Karniadakis, G.E.: Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. Journal of Computational Physics 397, 108850 (2019) (Cited on page 2)

(J. Darbon) Division of Applied Mathematics, Brown University.
E-mail address, J. Darbon: jerome.darbon@brown.edu

(G. P. Langlois) Division of Applied Mathematics, Brown University.
E-mail address, G. P. Langlois: gabriel.provencher.langlois@brown.edu

(T. Meng) Division of Applied Mathematics, Brown University.
E-mail address, T. Meng: tingwei.meng@brown.edu