INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

Deep Learning Methods For Inverse Problems Using Connections Between Proximal Operators And Hamilton–Jacobi Equations

OLUWATOSIN AKANDE[1], GABRIEL P. LANGLOIS[2], AND AKWUM ONWUNTA[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, 18015 USA

[2]Department of Mathematics, University of Illinois Urbana-Champaign, Chicago, IL, USA

ISE Technical Report 25T-025

LEHIGH UNIVERSITY

DEEP LEARNING METHODS FOR INVERSE PROBLEMS USING CONNECTIONS BETWEEN PROXIMAL OPERATORS AND HAMILTON–JACOBI EQUATIONS

OLUWATOSIN AKANDE*, GABRIEL P. LANGLOIS†, AND AKWUM ONWUNTA‡

Abstract. Inverse problems are important mathematical problems that seek to recover model parameters from noisy data. Since inverse problems are often ill-posed, they require regularization or incorporation of prior information about the underlying model or unknown variables. Proximal operators, ubiquitous in nonsmooth optimization, are central to this because they provide a flexible and convenient way to encode priors and build efficient iterative algorithms. They have also recently become key to modern machine learning methods, e.g., for plug-and-play methods for learned denoisers and deep neural architectures for learning priors of proximal operators. The latter was developed partly due to recent work characterizing proximal operators of nonconvex priors as subdifferential of convex potentials. In this work, we propose to leverage connections between proximal operators and Hamilton–Jacobi partial differential equations (HJ PDEs) to develop novel deep learning architectures for learning the prior. In contrast to other existing methods, we learn the prior directly without recourse to inverting the prior after training. We present several numerical results that demonstrate the efficiency of the proposed method in high dimensions.

1. Introduction. Inverse problems are ubiquitous mathematical problems that primarily aim at recovering model parameters from noisy data. They arise in many scientific and engineering applications for, e.g., recovering an image from noisy measurements, deblurring, tomographic reconstruction, and compressive sensing [3, 10, 39, 4]. Since inverse problems are often ill-posed, it is essential to include regularization or prior information about the underlying model or unknown variables. Proximal operators are central to this: they provide a flexible and computationally convenient way to encode priors and to build efficient iterative algorithms (e.g., proximal (sub)gradients, the alternating direction method of multipliers, and other splitting methods). More recently, proximal operators have become key ingredients for state-of-the-art machine learning methods, e.g., plug-and-play methods that replace explicit regularizers by learned denoisers [38, 40], and deep neural architectures that parameterize proximal maps or their gradients, such as learned proximal networks (LPNs) [33]. These developments have made proximal methods practical and powerful computational tools.

Formally, the proximal operator of a proper function $J\colon \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is defined via an observed data $\boldsymbol{x} \in \mathbb{R}^n$, a parameter $t > 0$, and the minimization problem

$$S(\boldsymbol{x}, t) = \min_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ \frac{1}{2t} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + J(\boldsymbol{y}) \right\}. \tag{1.1}$$

The proximal operator $\operatorname{prox}_{tJ}\colon \mathbb{R}^n \to \mathbb{R}$ is the set-valued function

$$\operatorname{prox}_{tJ}(\boldsymbol{x}) = \operatorname*{arg\,min}_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ \frac{1}{2t} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + J(\boldsymbol{y}) \right\}. \tag{1.2}$$

Here, t controls the trade-off between the quadratic data-fidelity term and the prior J. In practice one often works directly with prox_{tJ} rather than the prior.

*Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015, USA, (oaa323@lehigh.edu)

†Department of Mathematics, University of Illinois Urbana-Champaign, Chicago, IL, USA (gp42@illinois.edu).

‡Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015, USA, (ako221@lehigh.edu).

The recent work of Gribonval and Nikolova [37] in nonsmooth optimization has extended the characterization of proximal operators with convex priors to those with nonconvex priors, showing in particular they are functions that are subdifferentials of certain convex potentials. These properties, in particular, were used in [33] to develop new deep learning methods, called learned proximal networks (LPNs), to learn from data the underlying prior of a proximal operator.

The paper [37] did not, however, discuss the well-established, existing connections between proximal operators and Hamilton–Jacobi Partial Differential Equations (HJ PDEs) [21, 29, 23, 14, 48]. To see these connections, consider the following HJ PDE with quadratic Hamiltonian function and whose initial data is the prior J:

$$\begin{cases} \dfrac{\partial S}{\partial t}(\boldsymbol{x},t) + \dfrac{1}{2}\left\|\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t)\right\|_2^2 = 0, & \boldsymbol{x}\in\mathbb{R}^n\times(0,+\infty), \\ S(\boldsymbol{x},0) = J(\boldsymbol{x}), & \boldsymbol{x}\in\mathbb{R}^n. \end{cases} \tag{1.3}$$

If J is uniformly Lipschitz continuous, then the unique *viscosity solution* of the HJ PDE is given by (1.1). Moreover, at a point of differentiability $\boldsymbol{x}$, there holds

$$\operatorname{prox}_{tJ}(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t). \tag{1.4}$$

Moreover, the viscosity solution satisfies the crucial property that $\boldsymbol{x}\mapsto \frac{1}{2}\left\|\boldsymbol{x}\right\|_2^2 - tS(\boldsymbol{x},t)$ is convex; that is, when paired with (1.4), the function $\operatorname{prox}_{tJ}(\boldsymbol{x})$ is obtained from differentiating a convex function. This formally connects proximal operators to HJ PDEs, which we emphasize was previously known and established, and the (stronger) characterization obtained in [37][1].

In this paper, we leverage the theory of viscosity solutions of HJ PDEs to develop novel deep learning methods to learn from data the prior function J in (1.2). To describe our approach, consider the case when the solution $(\boldsymbol{x},t)\mapsto S(\boldsymbol{x},t)$ to the HJ PDE (1.3) is known. (We will consider the case when only samples of it are known in the next paragraph.) This problem was investigated in [7, 16, 17, 30, 46]. In particular, [30] showed that when $\boldsymbol{x}\mapsto S(\boldsymbol{x},t)$ is uniformly Lipschitz continuous and $\boldsymbol{x}\mapsto \frac{1}{2}\left\|\boldsymbol{x}\right\|_2^2 - tS(\boldsymbol{x},t)$ is convex, there exists a prior J that can recover $S(\boldsymbol{x},t)$ exactly. Moreover, there is a natural candidate for the prior, obtained by reversing the time in the HJ PDE (1.3) and using $(\boldsymbol{x},t)\mapsto S(\boldsymbol{x},t)$ as the terminal condition. The resulting *backward viscosity solution* yields the prior $J_{\mathrm{BVS}}\colon \mathbb{R}^n\to\mathbb{R}$ which admits the representation formula

$$J_{\mathrm{BVS}}(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n}\left\{S(\boldsymbol{x},t) - \frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2\right\}. \tag{1.5}$$

Here, $J(\boldsymbol{y}) \geqslant J_{\mathrm{BVS}}(\boldsymbol{y})$ for every $\boldsymbol{y}\in\mathbb{R}^n$, with $J_{\mathrm{BVS}}(\boldsymbol{y}) = J(\boldsymbol{y})$ whenever $\boldsymbol{y} = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t)$, where $\boldsymbol{x}$ is a point of differentiability of $\boldsymbol{x}\mapsto S(\boldsymbol{x},t)$. Moreover,

$$\inf_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2 + J_{\mathrm{BVS}}(\boldsymbol{y})\right\} = S(\boldsymbol{x},t) \text{ for every } \boldsymbol{x}\in\mathbb{R}^n.$$

Thus the prior J_{BVS} recovers the function $x\mapsto S(\boldsymbol{x},t)$, although in general prox_{tJ} and $\operatorname{prox}_{tJ_{\mathrm{BVS}}}$ may not agree everywhere. Nonetheless, this provides a principled way to estimate the prior, at least when $S(\boldsymbol{x},t)$ is known.

[1] To the best our knowledge, this characterization result was unknown in the theory of HJ PDEs.

We focus in this paper on the case when $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is unknown but have access to some samples $\{\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t)\}_{k=1}^K$ with t fixed. We propose to learn the prior $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y})$ by leveraging the crucial fact that $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ is convex, thus enabling approaches based on deep learning and convex neural networks.

Related works: Hamilton–Jacobi PDEs are important to many scientific and engineering applications arising in e.g., optimal control [5, 35, 42, 50] and physics [12, 13], inverse problems for imaging sciences [21, 26, 23, 25, 28], optimal transport [44, 47], game theory [8, 32, 52], and machine learning [15, 53]. Recent works focus on developing specialized solution methods for solving high-dimensional HJ PDEs, using, e.g., representation formulas or deep learning methods. These specialized methods leverage certain properties of HJ PDEs, including stochastic aspects and representation formulas [6, 42, 29, 28], to approximate solutions to HJ PDEs more accurately and efficiently than general-purpose methods. See, e.g., [45, 22, 27, 24, 49] for recent works along these lines and [43] for a review of the state-of-the-art numerical methods for HJ PDEs.

Deep learning methods have become popular for computing solutions to high-dimensional PDEs as well as their inverse problems. They are popular because neural networks can be trained on data to approximate high-dimensional, nonlinear functions using efficient optimization algorithms. They have been used to approximate solutions to PDEs without any discretization with numerical grids, and for this reason they can overcome, or at least mitigate, the curse of dimensionality. There is a fairly comprehensive literature on deep learning methods for solving PDEs in general, e.g., see [9, 20, 41].

Organization of this paper: We present background information on proximal operators, Hamilton–Jacobi equations, and convex neural networks in section 2. Next, we discuss recent results concerning the inverse problem for Hamilton–Jacobi equations when the solution is available, and how they relate to proximal operators and learning priors in inverse problems, in section 3. Our main theoretical results are presented in section 4, where we study the inverse problem for Hamilton–Jacobi equations when only incomplete information is available about its solution. We suggest via arguments from max-plus algebra theory for Hamilton–Jacobi PDEs how to learn from data the solution to a certain Hamilton-Jacobi–Jacobi terminal value problem, which can then be used as an estimate for learning the prior function in a proximal operator. We present in section 5 some numerical experiments for learning the initial data of certain Hamilton–Jacobi PDEs using convex neural networks and the theory of inverse Hamilton–Jacobi PDEs. Finally, we summarize our results in section 6.

2. Background.

We present here some background on proximal operators, HJ PDEs, connections between them, and convex neural networks. For comprehensive references, we refer the reader to [11, 31, 51].

2.1. Proximal operators.

Let $J\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ denote a proper function (i.e., $J(\boldsymbol{x}) < +\infty$ for some $\boldsymbol{x} \in \mathbb{R}^n$ and $J(\boldsymbol{x}) > -\infty$ for every $\boldsymbol{x} \in \mathbb{R}^n$). Consider the minimization problem $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ defined in (1.1) and its proximal operator $(\boldsymbol{x}, t) \mapsto \operatorname{prox}_{tJ}(\boldsymbol{x})$ defined in (1.2). We say a proper function $f_t\colon \mathbb{R}^n \to \mathbb{R}$ is a proximal operator of tJ if $f_t(\boldsymbol{x}) \in \operatorname{prox}_{tJ}(\boldsymbol{x})$ for every $\boldsymbol{x} \in \mathbb{R}^n$. Gribonval and Nikolova [37] proved that proximal operators are characterized in terms of the function

$\psi\colon \mathbb{R}^n \times [0,+\infty) \to \mathbb{R} \cup \{+\infty\}$ defined by

$$\psi(\boldsymbol{x},t) = \frac{1}{2}\|\boldsymbol{x}\|_2^2 - tS(\boldsymbol{x},t). \tag{2.1}$$

THEOREM 2.1. *A proper function $f_t\colon \mathbb{R}^n \to \mathbb{R}^n$ is a proximal operator of tJ if and only if $\boldsymbol{x} \mapsto \psi(\boldsymbol{x},t)$ is proper, lower semicontinuous, and convex and $f_t(\boldsymbol{x}) \in \partial_{\boldsymbol{x}}\psi(\boldsymbol{x},t)$. Moreover, f_t is uniformly Lipschitz continuous with constant $L > 0$ if and only if $\boldsymbol{x} \mapsto (1-1/L)\|\boldsymbol{x}\|_2^2/2 + tJ(\boldsymbol{x})$ is proper, lower semicontinuous and convex.*

Proof. See [37, Theorem 3 and Proposition 2] ∎

The characterization of proximal operators in Theorem 2.1 is closely related to the concepts of *semiconcave* and *semiconvex* functions.

DEFINITION 2.2. *Let $\mathcal{C} \subset \mathbb{R}^n$. We say $g\colon \mathcal{C} \to \mathbb{R}$ is C-semiconcave with $C \geqslant 0$ if it is continuous and*

$$\lambda g(\boldsymbol{x}_1) + (1-\lambda)g(\boldsymbol{x}_2) - g(\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) \leqslant \lambda(1-\lambda)C\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2$$

for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ such that $\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \subset \mathcal{C}$ and $\lambda \in [0,1]$. We say g is semiconvex if $-g$ is semiconcave.

REMARK 2.1. *It can be shown [11, Chapter 1] that a function g is C-semiconcave with $C \geqslant 0$ if and only if $\boldsymbol{x} \mapsto g(\boldsymbol{x}) - \frac{C}{2}\|\boldsymbol{x}\|_2^2$ is concave, if and only if $g = g_1 + g_2$, where g_1 is concave and $g_2 \in C^2(\mathbb{R}^n)$ with $\left\|\nabla_{\boldsymbol{x}}^2 g_2\right\|_\infty \leqslant C$.*

Combining formula (2.1), Definition 2.2 and Remark 2.1, we find $\boldsymbol{x} \mapsto \psi(\boldsymbol{x},t)$ is convex if and only if $\boldsymbol{x} \mapsto tS(\boldsymbol{x},t)$ is semiconcave. We will see later that semiconcavity is an important concept in the theory of HJ PDEs for characterizing their generalized solutions. But before moving on to present some background on HJ PDEs, we give below an instructive example.

EXAMPLE 2.1 (The negative absolute value prior). *Let $J(x) = -|x|$ and consider the one-dimensional problem*

$$S(x,t) = \min_{y\in\mathbb{R}} \left\{ \frac{1}{2t}(x-y)^2 - |y| \right\}.$$

A global minimum y^* of this problem satisfies the first-order optimality condition

$$0 \in (y^*-x)/t - \partial|y^*| \iff y^* \in \begin{cases} x+t, & \text{if } y^* > 0, \\ [x-t, x+t] & \text{if } y^* = 0, \\ x-t, & \text{if } y^* < 0. \end{cases}$$

If $x > t$, the only minimum is $y^* = x+t$. Likewise, if $x < -t$, the only minimum is $y^* = x-t$. In either cases, $S(x,t) = -x - \frac{t}{2}$. If $0 < x \leqslant t$, there are two local minimums, 0 and $x+t$, but the global minimum is attained at $x+t$ and yields $S(x,t) = -\frac{t}{2} - x$. Likewise, if $-t \leqslant x < 0$, there are two local minimums, 0 and $x-t$, but the global minimum is attained at $x-t$ and yields $S(x,t) = -\frac{t}{2} + x$. Finally, if $x = 0$, there are three local minimums, $-t$, 0, and t. The global minimums are attained at $-t$ or t, yielding $S(0,t) = -t/2$. Hence we find

$$S(x,t) = -\frac{t}{2} - |x| \quad \text{and} \quad prox_{tJ}(x) = \begin{cases} x+t, & \text{if } x > 0, \\ \{-t, t\} & \text{if } x = 0, \\ x-t, & \text{if } x < 0. \end{cases} \tag{2.2}$$

Thus, a selection $f_t(x) \in prox_{tJ}(x)$ differs only at $x = 0$. In any case, the function $x \mapsto \psi(x,t)$ in Theorem 2.1 and its subdifferential $x \mapsto \partial_x \psi(x,t)$ are given by

$$\psi(x,t) = \frac{1}{2}x^2 - tS(x,t) = \frac{1}{2}x^2 + t|x| + \frac{t^2}{2} \quad \text{and} \quad \partial_x \psi(x) = \begin{cases} x+t, & \text{if } x > 0, \\ [-t,t], & \text{if } x = 0, \\ x-t, & \text{if } x < 0. \end{cases}$$

We see that any selection $f_t(x) \in prox_{tJ}(x)$ satisfies $f(x) \in \partial\psi(x,t)$.

2.2. Hamilton–Jacobi Equations. In this section, we briefly review some elements of the theory of HJ PDEs, including the method of characteristics, viscosity solutions of HJ PDEs, and the Lax–Oleinik formula, and discuss how these concepts tie together to proximal operators. The discussion is not comprehensive; see [31] and references therein for a more detailed treatment. To ease the presentation, we consider only the first-order HJ PDEs (1.3).

2.2.1. Characteristic equations. The characteristic equations of (1.3) are given by the dynamical system

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{p}(t), \\ \dot{\boldsymbol{p}}(t) &= 0, \\ \dot{\boldsymbol{z}}(t) &= \frac{1}{2}\left\|\boldsymbol{p}(t)\right\|_2^2, \end{cases} \tag{2.3}$$

where $\boldsymbol{z}(t) = S(\boldsymbol{x}(t), t)$ and $\boldsymbol{x}(0) = J(\boldsymbol{x}(0)$. Here, $t \mapsto \boldsymbol{p}(t)$ is constant with $\boldsymbol{p}(t) \equiv \boldsymbol{p}(0) \in \mathbb{R}^n$. The characteristic line that arises from $\boldsymbol{x}(0) \in \mathbb{R}^n$ is $\boldsymbol{x}(t) = \boldsymbol{x}(0) + t\boldsymbol{p}(0)$, and so $\boldsymbol{z}(t) = \boldsymbol{z}(0) - \frac{1}{2}\left\|\boldsymbol{p}(0)\right\|_2^2$. Taken together, we find

$$S(\boldsymbol{x}(t), t) = J(\boldsymbol{x}(0)) + \frac{1}{2}\left\|\boldsymbol{p}(0)\right\|_2^2.$$

Writing $\boldsymbol{x}(t) \equiv \boldsymbol{x}$ and $\boldsymbol{p}(0) = \nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t)$ (assuming formally that the spatial gradient exists at $\boldsymbol{x}$) then $\boldsymbol{x}(0) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t)$, and so we find the representation

$$S(\boldsymbol{x}, t) = \frac{1}{2t}\left\|\nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t)\right\|_2^2 + J(\boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t)). \tag{2.4}$$

This gives an implicit representation between S, its spatial gradient, and the initial data J. Next, we turn to the explicit representation of solutions to (1.3).

2.2.2. Viscosity solutions and the Lax–Oleinik formula. The initial value problem (1.3) (and HJ PDEs with general Hamiltonians) may not have a unique generalized solution, i.e., those satisfying the HJ PDE almost everywhere along with the initial condition $S(\boldsymbol{x}, 0) = J(\boldsymbol{x})$.

EXAMPLE 2.2. *Let $J \equiv 0$ in (1.3) and take $n = 1$. The corresponding HJ PDE has infinitely many solutions: For instance, the functions S_1 and S_2 given by*

$$S_1(x,t) = 0, \quad S_2(x,t) = \begin{cases} 0, & \text{if } |x| \geqslant t, \\ x - t, & \text{if } 0 \leqslant x \leqslant t, \\ -x - t, & \text{if } -t \leqslant x \leqslant 0, \end{cases}$$

satisfy $S_1(x,0) = S_2(x,0) = J(x) = 0$ and both solve the corresponding HJ PDE almost everywhere.

The notion of *viscosity solution* was introduced in [19] to solve this problem. Under appropriate conditions (see [6, 18, 19]), the viscosity solution is unique and admits a representation formula. Specifically, for the initial value problem (1.3) with uniformly Lipschitz continuous initial data J, the unique viscosity solution is given by the Lax–Oleinik formula (with quadratic Hamiltonian)

$$S(\boldsymbol{x},t) = \inf_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ \frac{1}{2t} \|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + J(\boldsymbol{y}) \right\}. \tag{2.5}$$

The (unique) viscosity solution has two important properties. First, the function $\boldsymbol{x} \mapsto S(\boldsymbol{x},t)$ is (1/t)-semiconcave. This is equivalent to requiring the function $\boldsymbol{x} \mapsto \psi(\boldsymbol{x},t)$ defined in (2.1) to be convex, exactly as stipulated in Theorem 2.1. Second, at any point of differentiability of $\boldsymbol{x} \mapsto S(\boldsymbol{x},t)$, there holds

$$\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t) = \frac{\boldsymbol{x} - f_t(\boldsymbol{x})}{t} \iff f_t(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t), \tag{2.6}$$

where $f_t(\boldsymbol{x})$ denote a global minimum in (2.5). Note that substituting this expression in formula (2.4) obtained from the characteristic equations yields (2.5), as expected.

EXAMPLE 2.3 (The negative absolute value prior, continued.). *Let $J(x) = -|x|$ in the (one-dimensional) first-order HJ PDE (1.3). The function J is uniformly Lipschitz continuous and, as such, the Lax–Oleinik formula $S(x,t) = -\frac{t}{2} - |x|$ is the unique viscosity solution of the corresponding HJ PDE. Note $\boldsymbol{x} \mapsto S(\boldsymbol{x},t)$ is differentiable everywhere except at $\boldsymbol{x} = 0$ and $prox_{tJ}(\boldsymbol{x}) = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x},t)$ everywhere except at $\boldsymbol{x} = 0$ (see (2.2)).*

In summary, a proper function f_t is a proximal operator of tJ whenever the function $(\boldsymbol{x},t) \mapsto S(\boldsymbol{x},t)$ is the viscosity solution of the HJ initial value problem (1.3). The minimization problem underlying $\operatorname{prox}_{tJ}(\boldsymbol{x})$ is exactly the Lax–Oleinik representation formula of the viscosity solution of (1.3). We will see in the next section how to leverage these connections for learning the prior when $\boldsymbol{x} \mapsto J(\boldsymbol{x})$ is not available but $(\boldsymbol{x},t) \mapsto S(\boldsymbol{x},t)$ is available. But before proceeding, we briefly review convex neural networks, which will be used later in this work.

2.3. Convex neural networks. Convex Neural Networks, specifically Input Convex Neural Networks (ICNN), were introduced by [2] to allow for the efficient optimization of neural networks within structured prediction and reinforcement learning tasks. The core premise of an ICNN is to constrain the network architecture such that the output is a convex function with respect to the input.

To achieve convexity, the network typically employs a recursive structure for $k = 0, \ldots, j-1$

$$\boldsymbol{z}_{k+1} = g(\boldsymbol{W}_k \boldsymbol{z}_k + \boldsymbol{H}_k y + \boldsymbol{b}_k), f(\boldsymbol{y};\theta) = \boldsymbol{z}_j, \tag{2.7}$$

where $\boldsymbol{y}$, $\boldsymbol{z}_k$ represent the input to the network and the hidden features at layer k, respectively, and g is the activation function. To guarantee the convexity of the output with respect to the input $\boldsymbol{y}$, specific constraints are imposed on the parameters and the activation function, which are (i) the weights $\boldsymbol{W}_k$, which connect the previous hidden layer to the current one, must be non-negative ($\boldsymbol{W}_k \geqslant 0$), and (ii) the activation function g must be convex and non-decreasing [33].

Following [33, Proposition 3.1], Fang et al. leverage the ICNN architecture and the characterization of proximal operators to develop Learned Proximal Networks

(LPN) for inverse problems. LPNs require stricter conditions than standard ICNNs. While standard ICNNs often use ReLU activation, LPNs require the activation function g to be twice continuously differentiable. This smoothness is essential to ensure that the proximal operator is the gradient of a twice continuous differentiable function [37, Theorem 2]. Consequently, LPNs typically utilize smooth activations like the softplus function, a $\beta-$smooth approximation of ReLU [33, Section 3].

3. Connections between learning priors and the inverse problem for Hamilton–Jacobi Equations. In this section, we discuss the inverse problem of learning the prior in the proximal operator (1.2): given $t > 0$ and some function $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$, assess whether there exists a a prior function J that can recover $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ and, if so, estimate it. Due to the connections between proximal operators and HJ Equations, as discussed in Subsections 2.1–2.2, our starting point will be to discuss the inverse problem from the point of view of HJ Equations.

We summarize in the next subsection some of the main results for this problem, based on the results of [30] and other related works [16, 17, 46].

3.1. Reachability and inverse problems for Hamilton–Jacobi equations. We consider here the inverse problem associated to the HJ initial value problem (1.3): given $t > 0$ and a function $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$, identify the set of initial data $J\colon \mathbb{R}^n \to \mathbb{R}$ such that the viscosity solution of (1.3) coincide with $S(\boldsymbol{x}, t)$. That is, we wish to characterize the set

$$
\begin{aligned}
I_t(S) \coloneqq \{&J\colon \mathbb{R}^n \to \mathbb{R} \text{ is uniformly Lipschitz continuous} \\
&: S(\boldsymbol{x}, t) \text{ is obtained from (1.3) at time } t\}.
\end{aligned} \tag{3.1}
$$

We say the function $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ is *reachable* if the set $I_t(S)$ is nonempty. The main reachability result for the initial value problem (1.3) is the following:

THEOREM 3.1. *Suppose $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is uniformly Lipschitz continuous. Then the set $I_t(S)$ defined in (3.1) is nonempty if and only if $\boldsymbol{x} \mapsto tS(\boldsymbol{x}, t)$ is semiconcave.*

Proof. This follows from [30, Theorem 2.2, Theorem 6.1, and Definition 6.2]. ∎

Now, assume $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ is reachable. What can be said about the nonempty set $I_t(S)$? Since $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ is obtained from evolving forward in time the prior function J from 0 to t according to (1.3), a natural approach is to do the opposite: evolve backward in time from t to 0 the function $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. That is, we consider the terminal value problem

$$
\begin{cases}
\dfrac{\partial \boldsymbol{w}}{\partial \tau}(\boldsymbol{y}, \tau) + \dfrac{1}{2} \left\| \nabla_{\boldsymbol{y}} \boldsymbol{w}(\boldsymbol{y}, \tau) \right\|_2^2 = 0 & (\boldsymbol{y}, \tau) \in \mathbb{R}^n \times [0, t), \\
\boldsymbol{w}(\boldsymbol{y}, t) = S(\boldsymbol{y}, t), & \boldsymbol{y} \in \mathbb{R}^n.
\end{cases} \tag{3.2}
$$

Under appropriate conditions, the terminal-value problem (3.2) has a unique viscosity solution:

THEOREM 3.2. *Suppose $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is uniformly Lipschitz continuous and semiconcave. Then the viscosity solution of the terminal-value problem (3.2) exists, is unique, and is given by the representation formula*

$$
\boldsymbol{w}(\boldsymbol{y}, \tau) = \sup_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ S(\boldsymbol{x}, t) - \frac{1}{2\tau} \left\| \boldsymbol{x} - \boldsymbol{y} \right\|_2^2 \right\}. \tag{3.3}
$$

Moreover, the function $\boldsymbol{y} \mapsto \tau \boldsymbol{w}(\boldsymbol{y}, \tau)$ is semiconvex with unit constant.

Proof. See [7, Section 4, Equation 4.4.2] and [11, Chapter 1]. ∎

The viscosity solution of (3.2) is sometimes called the backward viscosity solution (BVS) to distinguish it from the viscosity solution of the initial value problem (1.3). The BVS at $\tau = 0$ corresponds to fully evolving backward in time the function $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. In what follows, we write $J_{\mathrm{BVS}} \coloneqq \boldsymbol{w}(\cdot, 0)$. We can use (2.1) to write

$$tJ_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2 = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \{\langle \boldsymbol{x}, \boldsymbol{y}\rangle - \psi(\boldsymbol{x}, t)\}. \tag{3.4}$$

The right hand side is the convex conjugate of $\boldsymbol{x} \mapsto \psi(\boldsymbol{x}, t)$ evaluated at $\boldsymbol{x}$, which is well-defined because $\boldsymbol{x} \mapsto \psi(\boldsymbol{x}, t)$ is proper, lower semicontinuous and convex.

Theorem 3.2 suggests that J_{BVS} is an initial condition that can reach $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. The next result stipulates that this is correct and that it is "optimal", in the sense that it bounds from below for any other reachable initial condition $J \in I_t(S)$.

THEOREM 3.3. *Let J_{BVS} denote the solution of the backward HJ terminal value problem 3.2 at time $\tau = 0$. Then $J \in I_t(S)$ if and only if*

$$J(\boldsymbol{y}) \geqslant J_{BVS}(\boldsymbol{y}) \text{ for every } \boldsymbol{y} \in \mathbb{R}^n, \text{ with equality for every } \boldsymbol{y} \in X_t(S), \text{ where}$$

$$X_t(S) \coloneqq \{\boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t) : \boldsymbol{x} \mapsto S(\boldsymbol{x}, t) \text{ is differentiable at } \boldsymbol{x} \in \mathbb{R}^n\}.$$

Proof. See [30, Theorems 2.3 and 2.4]. ∎

Theorem 3.3 stipulates that J_{BVS} is equal everywhere to J on the set $\mathcal{X}_t(S)$ and bounds it from below elsewhere. This is a fundamental consequence of the semiconcavity of $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$, which regularizes the backward viscosity solution of (3.2). We illustrate this below with the negative absolute value prior.

EXAMPLE 3.1 (The negative absolute value prior, continued.). *Let $J(x) = -|x|$ in the (one-dimensional) first-order HJ PDE (1.3). Recall that the unique viscosity solution is given by the Lax–Oleinik formula $S(x, t) = -\frac{t}{2} - |x|$. We now would like to compute the corresponding unique backward viscosity solution to the terminal-value problem (3.2). The solution is well-defined because $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is uniformly Lipschitz continuous and concave. We have*

$$J_{BVS}(x) = \sup_{\boldsymbol{y}\in\mathbb{R}} \left\{-\frac{t}{2} - |y| - \frac{1}{2t}(x-y)^2\right\} = -\frac{t}{2} - \inf_{\boldsymbol{y}\in\mathbb{R}} \left\{\frac{1}{2t}(x-y)^2 + |y|\right\}.$$

The infimum on the right hand side corresponds to the proximal operator of the function $\boldsymbol{y} \mapsto |y|$, which is the soft-thresholding operator:

$$\arg\min_{y\in\mathbb{R}} \left\{\frac{1}{2t}(x-y)^2 + |y|\right\} = \begin{cases} x - t, & \text{if } x > t, \\ 0, & \text{if } x \in [-t, t], \\ x + t, & \text{if } x < -t. \end{cases}$$

This gives

$$J_{BVS}(x) = \begin{cases} -x, & \text{if } x > t, \\ -\frac{t}{2} - \frac{x^2}{2t}, & \text{if } x \in [-t, t], \\ x, & \text{if } x < -t. \end{cases}$$

Here, a simple calculation shows $\mathcal{X}_t(S) = (-\infty, -t] \cup [t, +\infty)$, and we find $J(x) > J_{BVS}(x)$ on $(-t, t)$, as expected from Theorem 3.3. Moreover,

$$tJ_{BVS}(x) + \frac{1}{2}x^2 = \begin{cases} \frac{1}{2}(x-t)^2 - \frac{t^2}{2}, & \text{if } x > t, \\ -\frac{t^2}{2}, & \text{if } x \in [-t, t], \\ \frac{1}{2}(x+t)^2 - \frac{t^2}{2}, & \text{if } x < -t, \end{cases}$$

and we observe $x \mapsto tJ_{BVS}(x) + \frac{1}{2}x^2$ is convex, as expected from Theorem 3.2.

The results here apply when the function $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is known. What happens when only a finite set of values of this function are available?

4. Learning priors and the inverse problem for Hamilton–Jacobi Equations with incomplete information. In this section, we consider the inverse problem of learning the prior in the proximal operator (1.2) with incomplete information: given $t > 0$ and a set of samples $\{\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t)\}_{k=1}^K$, estimate the prior J that best recovers $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. Recall from Theorem 3.1 that when $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is uniformly Lipschitz continuous, $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is reachable if and only if it is semi-concave. In this case, the prior $\boldsymbol{x} \mapsto J_{\mathrm{BVS}}(\boldsymbol{x})$ obtained from the HJ terminal value problem (3.2) provides a prior function that recovers $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ exactly. Hence we will focus on studying how to approximate the prior J_{BVS} from a set of samples.

Note that if the triplet $(\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t))$ is known, then (i) the function is $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$ is differentiable at $\boldsymbol{x}$ and (ii) the unique minimum in the Lax–Oleinik formula (2.5) can be represented via (2.6):

$$S(\boldsymbol{x}_k, t) = \frac{1}{2t} \|\boldsymbol{x}_k - \boldsymbol{y}_k\|_2^2 + J(\boldsymbol{y}_k), \text{ with } \boldsymbol{y}_k = \boldsymbol{x}_k - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t). \tag{4.1}$$

Moreover, Theorem 3.3 and formula (2.1) imply $J(\boldsymbol{y}_k) = J_{\mathrm{BVS}}(\boldsymbol{y}_k)$, $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ is convex. Thus one possible approach for estimating J_{BVS} is to approximate $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ piecewise from below at the points $\{\boldsymbol{y}_k\}_{k=1}^K$.

We consider the problem of approximating J_{BVS} piecewise from below and its implications in subsection 4.1. This approximation problem turns out to be related closely to max-plus algebra theory for approximating solutions to HJ PDEs [1, 34, 36]; we discuss this in subsection 4.2. We then consider in subsection 4.3 the more general problem of learning a convex function to approximate $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ directly, applying the discussions in subsection 4.1- subsection 4.2.

4.1. Piecewise approximations. We consider here piecewise approximations of the prior $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y})$ using the samples $\{\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t)\}_{k=1}^K$ and formula (4.1). We consider first using a piecewise affine minorant (PAM) approximation, and then, assuming some regularity on J_{BVS}, using a piecewise quadratic minorant (PQM) approximation.

4.1.1. Piecewise affine approximation. We first consider the PAM approximation of the convex function $\boldsymbol{y} \mapsto tJ_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$,"

$$tJ_{\mathrm{PAM}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2 := \max_{k \in \{1,\dots,K\}} \left\{ tJ_{\mathrm{BVS}}(\boldsymbol{y}_k) + \frac{1}{2}\|\boldsymbol{y}_k\|_2^2 + \langle \boldsymbol{x}_k, \boldsymbol{y} - \boldsymbol{y}_k \rangle \right\}. \tag{4.2}$$

Then $J_{\mathrm{PAM}}(\boldsymbol{y}) \leqslant J_{\mathrm{BVS}}(\boldsymbol{y})$ for every $\boldsymbol{y} \in \mathbb{R}^n$, with $J_{\mathrm{PAM}}(\boldsymbol{y}_k) = J_{\mathrm{BVS}}(\boldsymbol{y}_k)$ at each $k \in \{1 \dots, K\}$. A short calculation gives

$$tJ_{\mathrm{PAM}}(\boldsymbol{y}) = \max_{k \in \{1,\dots,K\}} \left\{ tJ_{\mathrm{BVS}}(\boldsymbol{y}_k) + \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{y}_k\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{y}\|_2^2 \right\}.$$

How good is J_{PAM} as initial condition for the HJ PDE (1.3)? In light of Theorem 3.3, J_{PAM}, unsurprisingly, cannot reconstruct $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. Indeed, a formal calculation yields

$$\inf_{\boldsymbol{y} \in \mathbb{R}^n} \left\{ \frac{1}{2t}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + J_{\mathrm{PAM}}(\boldsymbol{y}) \right\} = \begin{cases} S(\boldsymbol{x}_k, t) & \text{if } \boldsymbol{x} = \boldsymbol{x}_k,\ k \in \{1, \dots, K\}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{4.3}$$

See Appendix A.1 for details. Thus approximating J_{BVS} via its PAM approximation recovers the samples $\{S(\boldsymbol{x}_k, t)\}_{k=1}^K$ but nothing else.

4.1.2. Piecewise quadratic approximation. Here, we assume $\boldsymbol{y} \mapsto tJ_{\mathrm{BVS}}(\boldsymbol{y})$ is semiconvex with constant $1-\alpha$ with $\alpha > 0$, so that $\boldsymbol{y} \mapsto tJ_{\mathrm{BVS}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ is $1-\alpha$ strongly convex. We can then approximate this strongly convex function via its PQMs:

$$tJ_{\mathrm{PQM}}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2 \coloneqq \max_{k\in\{1,\dots,K\}} \left\{ tJ_{\mathrm{BVS}}(\boldsymbol{y}_k) + \frac{1}{2}\|\boldsymbol{y}_k\|_2^2 + \langle \boldsymbol{x}_k, \boldsymbol{y}-\boldsymbol{y}_k\rangle + \frac{\alpha}{2}\|\boldsymbol{y}-\boldsymbol{y}_k\|_2^2 \right\}.$$

Then, $J_{\mathrm{PQM}}(\boldsymbol{y}) \leqslant J_{\mathrm{BVS}}(\boldsymbol{y})$ for every $\boldsymbol{y} \in \mathbb{R}^n$, with $J_{\mathrm{PQM}}(\boldsymbol{y}) = J_{\mathrm{BVS}}(\boldsymbol{y}_k)$ at each $k \in \{1,\dots,K\}$. Moreover, a short calculation gives

(4.4)
$$tJ_{\mathrm{PQM}}(\boldsymbol{y}) = \max_{k\in\{1,\dots,K\}} \left\{ J(\boldsymbol{y}_k) + \frac{1}{2}\|\boldsymbol{x}_k-\boldsymbol{y}_k\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_k-\boldsymbol{y}\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{y}-\boldsymbol{y}_k\|_2^2 \right\}.$$

How good is J_{PQM} as an initial condition for the HJ PDE (1.3)? Again, in light of Theorem 3.3, J_{PQM} cannot reconstruct $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. Nonetheless, a formal calculation yields

$$\inf_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ \frac{1}{2t}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + J_{\mathrm{PQM}}(\boldsymbol{y}) \right\} = \frac{1}{2t}\|\boldsymbol{x}-\boldsymbol{y}_k\|_2^2 + \frac{1}{2t\alpha}\|\boldsymbol{x}-\boldsymbol{x}_k\|_2^2 \tag{4.5}$$

for some $k \in \{1,\dots,K\}$. See Appendix A.2 for more details. Hence J_{PQM} leads to an approximation of $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ that is finite everywhere. In the next section, we describe how max-plus algebra theory [1, 34, 36] can be used to quantify the approximation errors more precisely.

4.2. Max-plus algebra theory for Hamilton–Jacobi PDEs and approximation results. We consider here max-plus algebra techniques for approximating solutions to certain HJ PDEs. Let $\alpha > 0$ and let $\Psi\colon \mathbb{R}^n \to \mathbb{R}$ denote a $(1-\alpha)$-semiconvex function obtained. Following [36, Section III], we approximate Ψ using K vectors $\{\boldsymbol{p}_k\}_{k=1}^K \subset \mathbb{R}^n$ with K semiconvex functions $\boldsymbol{y} \mapsto \langle \boldsymbol{p}_k, \boldsymbol{y}\rangle - \frac{1}{2}\|\boldsymbol{y}\|_2^2$ and a function $a\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$:

$$\Psi_{\mathrm{MP}}(\boldsymbol{y}) \coloneqq \max_{k\in\{1,\dots,K\}} \left\{ \langle \boldsymbol{p}_k, \boldsymbol{y}\rangle - \frac{1}{2}\|\boldsymbol{y}\|_2^2 - a(\boldsymbol{p}_k) \right\}. \tag{4.6}$$

Here, we suppose the vectors $\{\boldsymbol{p}_k\}_{k=1}^K$ and $\boldsymbol{p} \mapsto a(\boldsymbol{p})$ are selected so that $\Psi_{\mathrm{MP}}(\boldsymbol{y}) \leqslant \Psi(\boldsymbol{y})$. As discussed in subsection 4.1, such a selection is possible via the affine piecewise quadratic minorants of the $(1-\alpha)$-strongly convex function $\boldsymbol{y} \mapsto \Psi(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$. Let $\mathcal{Y}$ denote a full dimensional compact, convex subset of $\mathbb{R}^n$ and consider the L_∞ error

$$\epsilon_\infty(\Psi, K, \mathcal{Y}, \Psi_{\mathrm{MP}}) \coloneqq \sup_{\boldsymbol{y}\in\mathcal{Y}} |\Psi(\boldsymbol{y}) - \Psi_{\mathrm{MP}}(\boldsymbol{y})|.$$

Furthermore, we define the corresponding minimal L_∞ error as

$$\delta_\infty(\Psi, K, \mathcal{Y}) = \inf_{\Psi_{\mathrm{MP}} \leqslant \Psi} \epsilon_\infty(\Psi, K, \mathcal{Y}, \Psi_{\mathrm{MP}}).$$

The following result from max-plus algebra theory, proven in [36], stipulates that whatever vectors $\{\boldsymbol{p}_k\}_{k=1}^K$ and function $\boldsymbol{p} \mapsto a(\boldsymbol{p})$ are used to approximate Ψ, the minimal L_∞ error scales as an inverse power law in K and the dimension n in the limit $K \to +\infty$.

THEOREM 4.1 (Gaubert et al. (2011)). *Let $\alpha > 0$, and let $\mathcal{Y}$ denote a full-dimensional compact, convex subset of $\mathbb{R}^n$. If $\Psi \colon \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and $1-\alpha$ semiconvex, then there exists a constant $\beta(n) > 0$ depending only on n such that*

$$\delta_\infty(\Psi, K, \mathcal{Y}) \sim \beta(n) \left(\frac{1}{K} \int_{\mathcal{Y}} \left(\det\left(\nabla^2_{\boldsymbol{y}} \Psi(\boldsymbol{y}) + \boldsymbol{I}_{n\times n}\right)\right)^{\frac{1}{2}} d\boldsymbol{y} \right)^{2/n} \tag{4.7}$$

as $K \to +\infty$.

Thus the minimal L_∞ error is $\Omega(1/K^{2/n})$ as $K \to +\infty$, though the error is smaller the closer the Hessian matrix $\nabla^2_{\boldsymbol{y}} \Psi(\boldsymbol{y})$ is to the identity matrix $\boldsymbol{I}_{n\times n}$.

4.3. Applications to the inverse problem for Hamilton–Jacobi Equations. We consider here the problem of quantifying approximations of the prior function $\boldsymbol{y} \mapsto J_{\mathrm{BVS}}(\boldsymbol{y})$ when the latter is sufficiently regularized and when we have access to the values $\{\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t)\}_{k=1}^K$. Max-plus algebra theory provides us with a first approximation result:

COROLLARY 4.2. *Let $t > 0$ and assume tJ_{BVS} is twice continuously differentiable and $(1-\alpha)$-semiconvex with $\alpha > 0$. Let $\mathcal{Y}$ denote a full-dimensional compact, convex set of $\mathbb{R}^n$. Then there exists a constant $\beta(n)$ depending only on n such that*

$$\delta_\infty(tJ_{BVS}, K, \mathcal{Y}) \sim \beta(n) \left(\frac{1}{K} \int_{\mathcal{Y}} \det\left(t\nabla^2_{\boldsymbol{y}} J_{BVS}(\boldsymbol{y}) + \boldsymbol{I}_{n\times n}\right)^{\frac{1}{2}} d\boldsymbol{y} \right)^{2/n} \tag{4.8}$$

as $K \to +\infty$.

Proof. Immediate from Theorem 4.1 because J_{BVS} satisfies all its assumptions. ∎

Corollary 4.2 provides a lower bound for the approximation error of J_{BVS} relative to J_{PQM}. Indeed, Theorem 4.1 and Corollary 4.2 and the fact that $J_{\mathrm{PQM}}(\boldsymbol{y}) \leqslant J_{\mathrm{BVS}}(\boldsymbol{y})$ for every $\boldsymbol{y} \in \mathbb{R}^n$ imply

$$\delta_\infty(tJ_{\mathrm{BVS}}, K, \mathcal{Y}) \leqslant t \sup_{\boldsymbol{y}\in\mathcal{Y}} |J_{\mathrm{BVS}}(\boldsymbol{y}) - J_{\mathrm{PQM}}(\boldsymbol{y})|. \tag{4.9}$$

Thus in this case J_{PQM} approximates J_{BVS} from below in $\Omega(1/K^{n/2})$ as $K \to +\infty$. We show below a similar upper bound holds using any reachable function $\tilde{J} \in I_t(S)$.

THEOREM 4.3. *Let $t > 0$ and assume tJ_{BVS} is twice continuously differentiable and $(1-\alpha)$-semiconvex with $\alpha > 0$. Let $\mathcal{Y}$ denote a full-dimensional compact, convex set of $\mathbb{R}^n$ and let $\tilde{J} \in I_t(S)$ denote a function that can can reach $\boldsymbol{x} \mapsto S(\boldsymbol{x}, t)$. Then*

$$\delta_\infty(J_{BVS}, K, \mathcal{Y}) \leqslant t \sup_{\boldsymbol{y}\in\mathcal{Y}} |\tilde{J}(\boldsymbol{y}) - J_{PQM}(\boldsymbol{y})|. \tag{4.10}$$

Proof. First, note Theorem 3.3 implies $\tilde{J}(\boldsymbol{y}) \geqslant J_{\mathrm{BVS}}(\boldsymbol{y})$ for every $\boldsymbol{y} \in \mathbb{R}^n$, with equality for every $\boldsymbol{y} \in \mathbb{R}^n$ for which $\boldsymbol{y} = \boldsymbol{x} - t\nabla_{\boldsymbol{x}} S(\boldsymbol{x}, t)$ for some $\boldsymbol{x} \in \mathbb{R}^n$. Thus

$$t\tilde{J}(\boldsymbol{y}) - tJ_{\mathrm{BVS}}(\boldsymbol{y}) = (t\tilde{J}(\boldsymbol{y}) - tJ_{\mathrm{PQM}}(\boldsymbol{y})) + (tJ_{\mathrm{PQM}}(\boldsymbol{y}) - tJ_{\mathrm{BVS}}(\boldsymbol{y})) \geqslant 0,$$

which we rearrange to get

$$tJ_{\mathrm{BVS}}(\boldsymbol{y}) - tJ_{\mathrm{PQM}}(\boldsymbol{y}) \leqslant t\tilde{J}(\boldsymbol{y}) - tJ_{\mathrm{PQM}}(\boldsymbol{y}).$$

Since the set $\mathcal{Y}$ is a compact and convex set, $\sup_{\boldsymbol{y}\in\mathcal{Y}} |tJ_{\mathrm{BVS}}(\boldsymbol{y}) - tJ_{\mathrm{PQM}}(\boldsymbol{y})|$ is finite and attained in $\mathcal{Y}$, say at $\boldsymbol{y}^*$. Combining this with the inequality above yields

$$t \sup_{\boldsymbol{y}\in\mathcal{Y}} |J_{\mathrm{BVS}}(\boldsymbol{y}) - J_{\mathrm{PQM}}(\boldsymbol{y})| \leqslant t\tilde{J}(\boldsymbol{y}^*) - tJ_{\mathrm{PQM}}(\boldsymbol{y}^*) \leqslant t \sup_{\boldsymbol{y}\in\mathcal{Y}} |\tilde{J}(\boldsymbol{y}) - J_{\mathrm{PQM}}(\boldsymbol{y})|.$$

Finally, since J_{BVS} is twice continuously differentiable and $(1-\alpha)$ semiconvex with $\alpha > 0$, we can invoke Theorem 4.1 with $\Psi \equiv J_{\mathrm{BVS}}$ to get

$$\delta_\infty(J_{\mathrm{BVS}}, K, \mathcal{Y}) \leqslant t \sup_{\boldsymbol{y}\in\mathcal{Y}} |\tilde{J}(\boldsymbol{y}) - J_{\mathrm{PQM}}(\boldsymbol{y})|,$$

that is, inequality (4.10) holds. This concludes the proof. ∎

Theorem 4.3 suggests it is possible to learn J_{BVS} via a function $\tilde{J}$ that is twice continuously differentiable and (1-α)-semiconvex and assess the approximation error using the right-hand-side (4.10) as a proxy, in particular by driving $\sup_{\boldsymbol{y}\in\mathcal{Y}} |\tilde{J}(\boldsymbol{y}) - J_{\mathrm{PQM}}(\boldsymbol{y})|$ to zero using sufficiently large enough data by training $\tilde{J}(\boldsymbol{y})$ appropriately.

In the next section, we consider the problem of learning this function using deep neural networks, specifically learned proximal networks [33], to enforce the semiconvexity property required for $\tilde{J}$.

5. Numerical results. We evaluate Learned Proximal Networks (LPNs) for approximating the proximal operators of nonconvex and concave priors. While LPNs [33] are theoretically grounded in convex analysis (parameterizing the proximal operator as the gradient of a convex potential ψ), these experiments investigate their behavior when trained on data generated from fundamentally nonconvex and concave landscapes. All experiments utilize the official LPN implementation. The network is trained via supervised learning, minimizing the mean squared error (MSE) or L1 loss between the network output and the true value. We use an LPN with 2 layers and 256 hidden units using Softplus activation ($\beta = 5$) to ensure C^2 smoothness. The model is trained using the Adam optimizer with a starting learning rate of 10^{-3} and decreased by a factor of 10^{-1} at every 10^5 epochs for a total of 5×10^5 epochs.

The data generation process for all experiments is as follows: N samples (y_i) are drawn uniformly from the hypercube $[-a, a]^d$, where a is chosen to be 4 and d is the dimension, equal $2, 4, 8, 16, 32$ and 64. $N = 3 \times 10^4$ is chosen for $d = 2, 4$, $N = 3 \times 10^4$ is chosen for $d = 8, 16$, and $N = 4 \times 10^4$ is chosen for $d = 32, 64$.

We also trained a second LPN to recover the prior at arbitrary points and compare its performance to the "invert" method (find y such that $f_\theta(y) = x$) used in [33] for recovering the prior from its proximal. Our second LPN is based on the relationship that the non-convex prior $J(x)$ can be approximated using the convex conjugate of the learned potential $\psi(y)$. Specifically, we compute:

$$J(x) \approx G(x) - \frac{1}{2}\|x\|^2 \tag{5.1}$$

where $G(x) = \psi^*(x)$ represents the convex conjugate of the potential $\psi_\theta(y)$ learned by the first LPN. We generate a new dataset $\{(x_k, G_k)\}$ using the trained first LPN

ψ_θ: (i) The gradients of the first network evaluated at the original sample points y_i,

$$x_k = \nabla_y \psi_\theta(y_i), \tag{5.2}$$

and (ii) the values of the Legendre transform corresponding to each point,

$$G_k = \langle x_k, y_i \rangle - \psi_\theta(y_i). \tag{5.3}$$

The network ϕ_G is trained to map the gradients x_k to the conjugate values G_k by minimizing the Mean Squared Error (MSE). The optimization is performed using the Adam optimizer with the same parameters as used in the first LPN. Once the second LPN is trained, the estimated non-convex prior $\hat{J}(x)$ is recovered via

$$\hat{J}(x) = \phi_G(x) - \frac{1}{2}\|x\|^2. \tag{5.4}$$

5.1. Convex prior. We will benchmark our approach with the prior $J(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$. For this example, we have

$$\begin{aligned}\underset{\boldsymbol{y}\in\mathbb{R}^n}{\arg\min}\left\{\frac{1}{2t}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \|\boldsymbol{y}\|_1\right\} &= \cup_{j=1}^n \underset{y_j\in\mathbb{R}}{\arg\min}\left\{\frac{1}{2t}(x_j-y_j)^2 + |y_j|\right\} \\ &= \cup_{j=1}^n \begin{cases} x_j - t, & \text{if } x_j > t, \\ 0, & \text{if } x_j \in [-t,t], \\ x_j + t, & \text{if } x_j < -t. \end{cases}\end{aligned}$$

With this, we can evaluate $S(\boldsymbol{x}, t)$ and the LPN function $\boldsymbol{x} \mapsto \Psi(\boldsymbol{x}) \coloneqq \frac{1}{2} - tS(\boldsymbol{x}, t)$.

Table 1: Mean square errors of LPN ψ and prior J with 2 layers and 256 neurons in the convex L1 prior example.

	Dimension	LPN (ψ)	Prior (J)
Mean Square Errors	2D	$1.04E-5$	$3.33E-5$
	4D	$2.97E-5$	$2.17E-4$
	8D	$1.05E-4$	$7.25E-4$
	16D	$5.27E-3$	$2.11E-3$
	32D	$1.6E-1$	$4.03E-2$
	64D	$2.89E-6$	$2.69E-3$

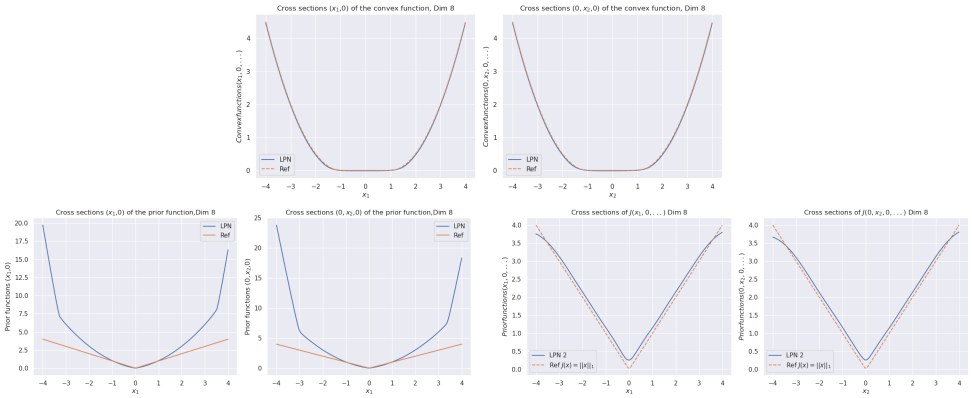

Fig. 1: The cross sections of the convex function $\psi(x)$ for dimension 8 (*top*). The bottom row compares the cross sections of the prior function from "invert LPN" (*left*) and our trained second LPN method (*right*).

5.2. Non-convex prior.

Minplus algebra example. For this example, the prior is

$$J(\boldsymbol{x}) = \min\left(\frac{1}{2\sigma_1}\left\|\boldsymbol{x}-\mu_1\right\|_2^2, \frac{1}{2\sigma_2}\left\|\boldsymbol{x}-\mu_2\right\|_2^2\right).$$

We use $\mu_1 = (1, 0, \dots, 0)$, $\mu_2 = \mathbf{1}/\sqrt{n}$, and $\sigma_1 = \sigma_2 = 1.0$.

Table 2: Mean square errors of LPN ψ and prior J with 2 layers and 256 neurons in the min-plus example.

	Dimension	LPN (ψ)	Prior (J)
Mean Square Errors	2D	$3.33E-6$	$5.73E-7$
	4D	$7.64E-6$	$4.92E-6$
	8D	$3.64E-5$	$1.20E-4$
	16D	$1.99E-4$	$3.44E-4$
	32D	$1.16E-3$	$1.33E-3$
	64D	$2.32E-9$	$5.21E-5$

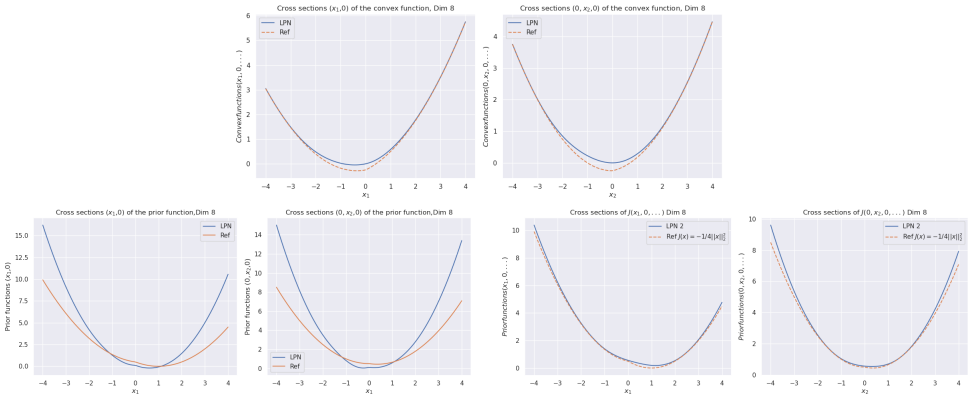

Fig. 2: The cross sections of the convex function $\psi(x)$ for dimension 8 (*top*). The bottom row compares the cross sections of the prior function from "invert LPN" (*left*) and our trained second LPN method (*right*).

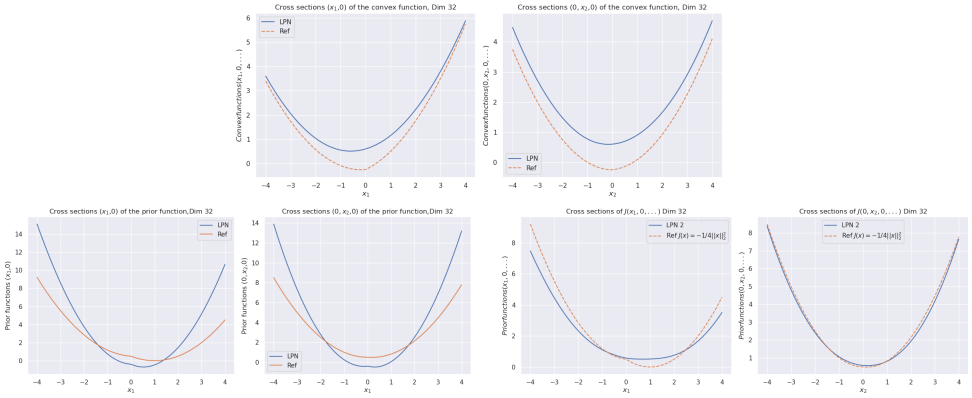

Fig. 3: The cross sections of the convex function $\psi(x)$ for dimension 32 (*top*). The bottom row compares the cross sections of the prior function from "invert LPN" (*left*) and our trained second LPN method (*right*)

5.3. Concave prior. For this example, we use

$$J(\boldsymbol{x}) = -\left\|\boldsymbol{x}\right\|_2^2/4.$$

This is actually challenging example because, technically, J is not uniformly Lipschitz continuous. (Although numerically we can get around this by "Huberizing" the prior.) We use this prior because we have an exact solution for this problem. It's also a bit challenging for a convex LPN network because according to the theory of HJ PDEs, the function $J + \frac{1}{4}\left\|\boldsymbol{x}\right\|_2^2$ is convex, and $J + \frac{1}{2}\left\|\boldsymbol{x}\right\|_2^2$ is strongly convex, so an LPN (which is not inherently S.C.) may not be able to detect the strong convexity and makes this function more challenging to learn.

Table 3: Mean square errors of LPN ψ and prior J with 2 layers and 256 neurons in the concave prior example.

	Dimension	LPN (ψ)	Prior (J)
Mean Square Errors	2D	$7.00E-7$	$1.57E-6$
	4D	$2.74E-5$	$7.70E-5$
	8D	$5.58E-4$	$7.91E-4$
	16D	$3.69E-3$	$3.28E-3$
	32D	$8.70E-2$	$3.01E-2$
	64D	$6.23E-6$	$1.87E-3$

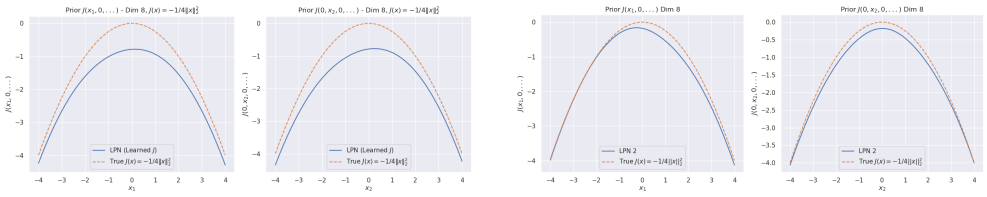

Fig. 4: The cross sections of the prior function for 8 dimension from "invert LPN" (*left*) and our trained second LPN method (*right*)

5.4. Negative ℓ_1 norm. See Equation (5.6) for the minimum value and (5.7) for the proximal value. For this example, we consider $J(\boldsymbol{y}) = -\|\boldsymbol{y}\|_1$. Let $n = 1$ for simplicity and consider the one-dimensional problem

$$S(x,t) = \min_{y\in\mathbb{R}} \left\{ \frac{1}{2t}(x-y)^2 - |y| \right\}. \tag{5.5}$$

The function $y \mapsto (x-y)^2/2t - |y|$ is differentiable everywhere except at $y = 0$. A stationary point of this function satisfies

$$0 \in \frac{y-x}{t} - \partial|y| \iff x \in \begin{cases} x-t, & \text{if } y<0, \\ [x-t, x+t] & \text{if } y=0, \\ x+t, & \text{if } y>0. \end{cases}$$

If $x > t$, the only minimum is $x + t$, in which case we have

$$S(x,t) = \frac{1}{2t}(x+t-x)^2 - |x+t| = \frac{t}{2} - (x+t) = -\frac{t}{2} - x.$$

If $0 < x \leqslant t$, there are two local minimums, 0 and $x + t$, but the global minimum is attained at $x + t$, again yielding $S(x,t) = \frac{t}{2} - x$. If $x = 0$, we have three local minimums: $-t$, 0, and t. The global minimum is attained at either $-t$ or t, yielding $S(0,t) = -\frac{t}{2}$. If $-t \leqslant x < 0$, there are two local minimums, 0 and $x - t$, but the global minimum is attained at $x - t$, yielding $S(x,t) = -\frac{t}{2} + x$. If $x < -t$, the only minimum is $x - t$, in which case we have $S(x,t) - \frac{t}{2} + x$. Hence

$$S(x,t) = -\frac{t}{2} - |x|. \tag{5.6}$$

In particular, its gradient in x is given by

$$\nabla_x S(x,t) \in \begin{cases} 1 & \text{if } x < 0, \\ [-1,1] & \text{if } x = 0, \\ -1 & \text{if } x > 0. \end{cases}$$

Moreover,

$$\arg\min_{y\in\mathbb{R}} \left\{ \frac{1}{2t}(x-y)^2 - |y| \right\} = \begin{cases} x - t & \text{if } x < 0, \\ [-t,t] & \text{if } x = 0, \\ x + t & \text{if } x > 0. \end{cases} \tag{5.7}$$

Table 4: Mean square errors of LPN ψ and prior J with 2 layers and 256 neurons in the negative L_1 norm examples.

	Dimension	LPN (ψ)	Prior (J)
Mean Square Errors	2D	$6.59E-5$	$5.20E-6$
	4D	$3.15E-4$	$3.17E-5$
	8D	$2.12E-3$	$2.94E-4$
	16D	$8.01E-3$	$4.49E-2$
	32D	$1.55E-1$	$2.29E-2$
	64D	$6.42E-4$	$4.49E-3$

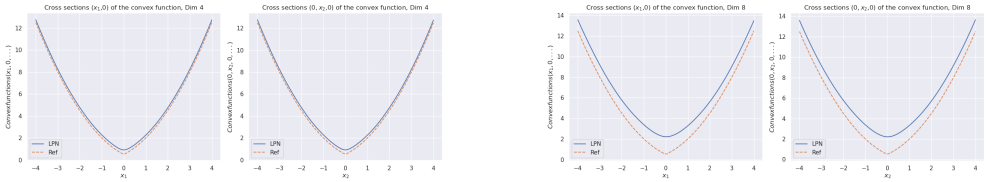

Fig. 5: The cross sections of the convex function $\psi(x)$ for dimension 4 (*left*) and 8 (*right*).

The Tables 1, 2, 3, and 4 quantify the scalability of the LPN approach across dimensions ranging from 2D to 64D. The results indicate that the method performs with high accuracy in lower dimensions (2D through 8D), achieving mean square errors in the range of 10^{-7} to 10^{-4}. However, performance degrades slightly as the dimensionality increases, particularly at 16D and 32D, where the error spikes to a range of 10^{-3} and 10^{-1}. The higher dimensions did not work too well, which might be because we did not train for long enough and also used a simple architecture (which ought to be more intricate for the higher dimensional problems). While the error for the recovered prior J is generally slightly higher than that of LPN ψ, this is expected given the added complexity of recovering the non-convex, non-smooth, or concave functions. However, the errors generally remain low, validating our method's effectiveness even in high-dimensional spaces.

The top rows of Figures 1,2, 3 and Figure 5 demonstrate that the LPN accurately learns the cross sections of the convex function $\psi(x)$ for dimension $4, 8$, and 32, closely

matching the reference function with the most significant variation in Figure 3 corresponding to dimension 32. The bottom row of these figures compares the "invert LPN" (*left*) method and our trained second LPN method (*right*). It is clear that in all cases, our direct method recovers the original non-smooth prior, as indicated by the sharp V-shaped reconstructions in Figure 1, the non-convex prior Figures 2 and 3, and quadratic concave prior Figure 4 despite its challenging nature.

Figure 5 represents the cross-sections of the LPN value function against the ground truth for dimensions 4 (*left*) and 8 (*right*). In both instances, the LPN-approximation curves exhibit a tight fit to the analytical reference, capturing the characteristic V-shape more closely and the non-smooth geometry of the underlying function. The errors for 64 dimensions in all the examples are consistently lower than expected. In contrast, it does not yield a good visual approximation of the cross sections. We are unable to explain the reason for this behaviour in 64D cases yet; however, we think it is probably due to the way we sample the hypercube.

6. Discussion. In this work, we leveraged the theory of viscosity solutions of HJ PDEs to develop novel deep learning numerical methods to learn, from data, the underlying prior of the proximal operator (1.2) yielding $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ defined in (1.1). Our approach built on the existing connections between proximal operators and HJ PDEs, crucially the fact that $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ is obtained from the solution to an HJ PDE, and in particular on the theory for the inverse problem for HJ equations. As discussed in section 3, the theory for the inverse problem for HJ equations show that while there may be infinitely many priors that can recover (1.1), there is a natural choice, obtained by reversing the time in the HJ PDE (3.2) and using the value of the proximal operator $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ as initial condition. The resulting backward viscosity solution yields a prior J_{BVS} that can reconstruct the $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ and also that is semiconvex. We considered the case where only samples of the proximal operators and its values were available in Section section 4, and used techniques from max-plus algebra to derive some characterizations and errors property of J_{BVS} with respect to convex functions approximating it from above. Finally, in section 5 we proposed to learn the prior J_{BVS} by training a convex neural network, specifically a learned proximal network, on a function of the form $\boldsymbol{y} \mapsto \tilde{J}(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y}\|_2^2$ from data $\{\boldsymbol{x}_k, S(\boldsymbol{x}_k, t), \nabla_{\boldsymbol{x}} S(\boldsymbol{x}_k, t)\}_{k=1}^K$ via (4.1). We presented several numerical results that demonstrate the efficiency of our proposed method in high dimensions.

While this work focused on proximal operators, we expect our approach can be extended to a broad class of Bregman divergences, as recent results in the theory of inverse problems for HJ equations suggest [30]. Another potential direction would be in the case where the value of the proximal operator $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ is known to learn the prior J using Monte Carlo sampling strategies, as recently proposed in [49] for the forward problem of HJ equations (i.e., learning $(\boldsymbol{x}, t) \mapsto S(\boldsymbol{x}, t)$ from known J). In the longer term, it would be interesting to devise similar deep learning methods for the inverse problem of HJ equations with possibly time- or state-dependent Hamiltonians, relevant to optimal control problems.

Appendix A. Calculations.

A.1. Formal calculation of (4.3). We have

$$\begin{aligned}
\inf_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2+J_{\mathrm{PAM}}(\boldsymbol{y})\right\} &= \inf_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2+\right.\\
&\qquad\left.\max_{k\in\{1,\dots,K\}}\left\{J_{\mathrm{BVS}}(\boldsymbol{y}_k)+\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}_k\right\|_2^2-\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}\right\|_2^2\right\}\right\}\\
&= \max_{k\in\{1,\dots,K\}}\left\{J_{\mathrm{BVS}}(\boldsymbol{y}_k)+\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}_k\right\|_2^2+\right.\\
&\qquad\left.\inf_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2-\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}\right\|_2^2\right\}\right\}\\
&= \begin{cases} S(\boldsymbol{x}_k,t) & \text{if } \boldsymbol{x}=\boldsymbol{x}_k,\ k\in\{1,\dots,K\},\\ +\infty, & \text{otherwise.}\end{cases}
\end{aligned}$$

A.2. Formal calculation of (4.5). Formally, we have

$$t\nabla_{\boldsymbol{y}}J_{\mathrm{PQM}}(\boldsymbol{y})=\alpha(\boldsymbol{y}-\boldsymbol{y}_k)+\boldsymbol{x}_k-\boldsymbol{y}$$

for some $k\in\{1,\dots,K\}$. Similarly,

$$\begin{aligned}
\hat{\boldsymbol{y}}=\operatorname*{arg\,min}_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2+J_{\mathrm{PQM}}(\boldsymbol{y})\right\} &\iff \boldsymbol{0}=\frac{\hat{\boldsymbol{y}}-\boldsymbol{x}}{t}+\frac{\alpha}{t}(\hat{\boldsymbol{y}}-\boldsymbol{y}_k)+\frac{\boldsymbol{x}_k-\hat{\boldsymbol{y}}}{t}\\
&\iff \hat{\boldsymbol{y}}=\boldsymbol{y}_k+\frac{\boldsymbol{x}-\boldsymbol{x}_k}{\alpha}.
\end{aligned}$$

In addition,

$$\begin{aligned}
&\boldsymbol{x}-\hat{\boldsymbol{y}}=\frac{\boldsymbol{x}_k-(1-\alpha)\boldsymbol{x}}{\alpha}-\boldsymbol{y}_k \implies \frac{1}{2t}\left\|\boldsymbol{x}-\hat{\boldsymbol{y}}\right\|_2^2=\frac{1}{2t\alpha^2}\left\|\boldsymbol{x}_k-\boldsymbol{x}+\alpha(\boldsymbol{x}-\boldsymbol{y}_k)\right\|_2^2,\\
&\boldsymbol{x}_k-\hat{\boldsymbol{y}}=\boldsymbol{x}_k-\boldsymbol{y}_k+\frac{\boldsymbol{x}_k-\boldsymbol{x}}{\alpha},
\end{aligned}$$

and

$$\begin{aligned}
J_{\mathrm{PQM}}(\hat{\boldsymbol{y}}) &= J(\boldsymbol{y}_k)+\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}_k\right\|_2^2-\frac{1}{2t}\left\|\boldsymbol{x}_k-\hat{\boldsymbol{y}}\right\|_2^2+\frac{\alpha}{2t}\left\|\hat{\boldsymbol{y}}-\boldsymbol{y}_k\right\|_2^2\\
&= J(\boldsymbol{y}_k)+\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}_k\right\|_2^2-\frac{1}{2t}\left\|\boldsymbol{x}_k-\boldsymbol{y}_k+\frac{\boldsymbol{x}_k-\boldsymbol{x}}{\alpha}\right\|_2^2+\frac{\alpha}{2t}\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_k}{1-c}\right\|_2^2\\
&= \langle\boldsymbol{x}_k-\boldsymbol{y}_k,\boldsymbol{x}-\boldsymbol{x}_k\rangle/t\alpha-\frac{1}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2+\frac{\alpha}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2\\
&= \langle\boldsymbol{x}_k-\boldsymbol{y}_k,\boldsymbol{x}-\boldsymbol{x}_k\rangle/t\alpha-\frac{c}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2.
\end{aligned}$$

From this, we deduce

$$\begin{aligned}
\inf_{\boldsymbol{y}\in\mathbb{R}^n}\left\{\frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2+J_{\mathrm{PQM}}(\boldsymbol{y})\right\} &= \frac{1}{2t\alpha^2}\left\|\boldsymbol{x}_k-\boldsymbol{x}+\alpha(\boldsymbol{x}-\boldsymbol{y}_k)\right\|_2^2 \\
&\quad+\langle\boldsymbol{x}_k-\boldsymbol{y}_k,\boldsymbol{x}-\boldsymbol{x}_k\rangle/t\alpha-\frac{c}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2, \\
&= \frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}_k\right\|_2^2+\frac{1}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2 \\
&\quad-\langle\boldsymbol{x}-\boldsymbol{x}_k,\boldsymbol{x}-\boldsymbol{y}_k\rangle/t\alpha \\
&\quad+\langle\boldsymbol{x}_k-\boldsymbol{y}_k,\boldsymbol{x}-\boldsymbol{x}_k\rangle/t\alpha-\frac{c}{2t\alpha^2}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2 \\
&= \frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}_k\right\|_2^2+\frac{1}{2t(\alpha}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2 \\
&\quad-\frac{1}{t\alpha}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2 \\
&= \frac{1}{2t}\left\|\boldsymbol{x}-\boldsymbol{y}_k\right\|_2^2+\frac{1}{2t\alpha}\left\|\boldsymbol{x}-\boldsymbol{x}_k\right\|_2^2,
\end{aligned}$$

for some $k\in\{1,\ldots,K\}$.

REFERENCES

[1] M. Akian, R. Bapat, S. Gaubert, et al., *Max-plus algebra*, Handbook of linear algebra, 39 (2006), pp. 10–14.

[2] B. Amos, L. Xu, and J. Z. Kolter, *Input convex neural networks*, in International conference on machine learning, PMLR, 2017, pp. 146–155.

[3] H. Antil, H. C. Elman, A. Onwunta, and D. Verma, *A deep neural network approach for parameterized PDEs and Bayesian inverse problems*, Machine Learning: Science and Technology, 4 (2023).

[4] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numerica, 28 (2019), pp. 1–174.

[5] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Systems & Control: Foundations & Applications, Birkhäuser Boston, Inc., Boston, MA, 1997.

[6] M. Bardi and S. Faggian, *Hopf-type estimates and formulas for nonconvex nonconcave hamilton–jacobi equations*, SIAM J. Math. Anal., 29 (1998), pp. 1067–1086.

[7] E. Barron, P. Cannarsa, R. Jensen, and C. Sinestrari, *Regularity of Hamilton–Jacobi equations when forward is backward*, Indiana Univ. Math. J., (1999), pp. 385–409.

[8] E. Barron, L. Evans, and R. Jensen, *Viscosity solutions of Isaacs' equations and differential games with Lipschitz controls*, J. Differential Equations, 53 (1984), pp. 213 – 233.

[9] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, *An overview on deep learning-based approximation methods for partial differential equations*, arXiv preprint arXiv:2012.12348, (2020).

[10] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*, CRC press, 2021.

[11] P. Cannarsa and C. Sinestrari, *Semiconcave Functions, Hamilton—Jacobi Equations, and Optimal Control*, Springer, 2004.

[12] C. Carathéodory, *Calculus of variations and partial differential equations of the first order. Part I: Partial differential equations of the first order*, Holden-Day, Inc., San Francisco-London-Amsterdam, 1965.

[13] C. Carathéodory, *Calculus of variations and partial differential equations of the first order. Part II: Calculus of variations*, Holden-Day, Inc., San Francisco-London-Amsterdam, 1967.

[14] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier, *Deep relaxation: partial differential equations for optimizing deep neural networks*, Research in the Mathematical Sciences, 5 (2018), p. 30.

[15] P. Chen, T. Meng, Z. Zou, J. Darbon, and G. E. Karniadakis, *Leveraging multitime Hamilton–Jacobi PDEs for certain scientific machine learning problems*, SIAM J. Sci. Comput., 46 (2024), pp. C216–C248.

[16] C. G. Claudel and A. M. Bayen, *Convex formulations of data assimilation problems for a class of Hamilton–Jacobi equations*, SIAM J. Optim., 49 (2011), pp. 383–402.
[17] R. M. Colombo and V. Perrollaz, *Initial data identification in conservation laws and Hamilton–Jacobi equations*, J. Math. Pures Appl. (9), 138 (2020), pp. 1–27.
[18] M. G. Crandall, H. Ishii, and P.-L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. (N.S.), 27 (1992), pp. 1–67.
[19] M. G. Crandall and P.-L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American mathematical society, 277 (1983), pp. 1–42.
[20] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, *Scientific machine learning through physics–informed neural networks: Where we are and what's next*, J. Sci. Comput., 92 (2022), p. 88.
[21] J. Darbon, *On convex finite-dimensional variational methods in imaging sciences and Hamilton–Jacobi equations*, SIAM J. Imaging Sci., 8 (2015), pp. 2268–2293.
[22] J. Darbon, P. M. Dower, and T. Meng, *Neural network architectures using min-plus algebra for solving certain high-dimensional optimal control problems and Hamilton–Jacobi pdes*, Math. Control Signals Systems, 35 (2023), pp. 1–44.
[23] J. Darbon and G. P. Langlois, *On Bayesian posterior mean estimators in imaging sciences and Hamilton–Jacobi partial differential equations*, J. Math. Imaging Vision, 63 (2021), pp. 821–854.
[24] J. Darbon, G. P. Langlois, and T. Meng, *Overcoming the curse of dimensionality for some Hamilton–Jacobi partial differential equations via neural network architectures*, Res. Math. Sci., 7 (2020), p. 20.
[25] J. Darbon, G. P. Langlois, and T. Meng, *Connecting Hamilton-Jacobi partial differential equations with maximum a posteriori and posterior mean estimators for some non-convex priors*, Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision, (2021), pp. 1–25.
[26] J. Darbon and T. Meng, *On decomposition models in imaging sciences and multi-time Hamilton-Jacobi partial differential equations*, arXiv preprint arXiv:1906.09502, (2019).
[27] J. Darbon and T. Meng, *On some neural network architectures that can represent viscosity solutions of certain high-dimensional Hamilton–Jacobi partial differential equations*, J. Comput. Phys., 425 (2021), p. 109907.
[28] J. Darbon, T. Meng, and E. Resmerita, *On Hamilton-Jacobi pdes and image denoising models with certain nonadditive noise*, J. Math. Imaging Vision, 64 (2022), pp. 408–441.
[29] J. Darbon and S. Osher, *Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere*, Res. Math. Sci. Sciences, 3 (2016), p. 19.
[30] C. Esteve and E. Zuazua, *The inverse problem for Hamilton–Jacobi equations and semiconcave envelopes*, SIAM J. Math. Anal., 52 (2020), pp. 5627–5657.
[31] L. C. Evans, *Partial differential equations*, vol. 19, American Mathematical Society, 2022.
[32] L. C. Evans and P. E. Souganidis, *Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations*, Indiana Univ. Math. J., 33 (1984), pp. 773–797.
[33] Z. Fang, S. Buchanan, and J. Sulam, *What's in a prior? Learned proximal networks for inverse problems*, in The Twelfth International Conference on Learning Representations, 2024.
[34] W. H. Fleming and W. M. McEneaney, *A max-plus-based algorithm for a Hamilton–Jacobi–Bellman equation of nonlinear filtering*, SIAM J. Optim., 38 (2000), pp. 683–710.
[35] W. H. Fleming and H. M. Soner, *Controlled Markov processes and viscosity solutions*, vol. 25, Springer Science & Business Media, 2006.
[36] S. Gaubert, W. McEneaney, and Z. Qu, *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, IEEE, 2011, pp. 1054–1061.
[37] R. Gribonval and M. Nikolova, *A characterization of proximity operators*, J. Math. Imaging Vision, 62 (2020), pp. 773–789.
[38] J. Hu, W. Gan, Z. Sun, H. An, and U. S. Kamilov, *A plug-and-play image registration network*, arXiv preprint arXiv:2310.04297, (2023).
[39] V. Isakov, *Inverse Problems for Partial Differential Equations*, vol. 127, Springer, 2017.
[40] F. Jia, Y. Huang, S.-H. Wang, C. Garcia-Cardona, A. L. Bertozzi, and B. Wang, *Plug-and-play image restoration with flow matching: A continuous viewpoint*, arXiv preprint arXiv:2512.04283, (2025).
[41] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Physics-informed machine learning*, Nat. Rev. Phys., 3 (2021), pp. 422–440.

[42] W. McEneaney, *Max-plus methods for nonlinear control and estimation*, Springer Science & Business Media, 2006.
[43] T. Meng, S. Liu, S. W. Fung, and S. Osher, *Recent advances in numerical solutions for Hamilton-Jacobi PDEs*, arXiv preprint arXiv:2502.20833, (2025).
[44] T. Meng, S. Liu, W. Li, and S. Osher, *A primal-dual hybrid gradient method for solving optimal control problems and the corresponding Hamilton-Jacobi PDEs*, arXiv preprint arXiv:2403.02468, (2024).
[45] T. Meng, Z. Zhang, J. Darbon, and G. Karniadakis, *Sympocnet: Solving optimal control problems with applications to high-dimensional multiagent path planning problems*, SIAM Journal on Scientific Computing, 44 (2022), pp. B1341–B1368.
[46] A. Misztela and S. Plaskacz, *An initial condition reconstruction in Hamilton–Jacobi equations*, Nonlinear Anal., 200 (2020), p. 112082.
[47] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, *Ot-flow: Fast and accurate continuous normalizing flows via optimal transport*, in Proc. AAAI Conf. Artif. Intell., vol. 35, 2021, pp. 9223–9232.
[48] S. Osher, H. Heaton, and S. Wu Fung, *A Hamilton–Jacobi-based proximal operator*, Proc. Natl. Acad. Sci., 120 (2023), p. e2220469120.
[49] Y. Park and S. Osher, *Neural implicit solution formula for efficiently solving Hamilton-Jacobi equations*, arXiv preprint arXiv:2501.19351, (2025).
[50] C. Parkinson, D. Arnold, A. L. Bertozzi, Y. T. Chow, and S. Osher, *Optimal human navigation in steep terrain: a Hamilton-Jacobi-Bellman approach*, arXiv preprint arXiv:1805.04973, (2018).
[51] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317, Springer Science & Business Media, 2009.
[52] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung, *A machine learning framework for solving high-dimensional mean field game and mean field control problems*, Proc. Natl. Acad. Sci., 117 (2020), pp. 9183–9193.
[53] Z. Zou, T. Meng, P. Chen, J. Darbon, and G. E. Karniadakis, *Leveraging viscous Hamilton–Jacobi PDEs for uncertainty quantification in scientific machine learning*, SIAM/ASA J. Uncertain. Quantif., 12 (2024), pp. 1165–1191.