A SUPERVISED LEARNING SCHEME FOR COMPUTING HAMILTON–JACOBI EQUATION VIA DENSITY COUPLING*

JIANBO CUI†, SHU LIU‡, AND HAOMIN ZHOU§

Abstract. We propose a supervised learning scheme for the first order Hamilton–Jacobi PDEs in high dimensions. The scheme is designed by using the geometric structure of Wasserstein Hamiltonian flows via a density coupling strategy. It is equivalently posed as a regression problem using the Bregman divergence, which provides the loss function in learning while the data is generated through the particle formulation of Wasserstein Hamiltonian flow. We prove a posterior estimate on L^1 residual of the proposed scheme based on the coupling density. Furthermore, the proposed scheme can be used to describe the behaviors of Hamilton–Jacobi PDEs beyond the singularity formations on the support of coupling density. Several numerical examples with different Hamiltonians are provided to support our findings.

Key words. Hamilton–Jacobi PDE, high dimension, supervised learning, density coupling, Wasserstein Hamiltonian flow

MSC codes. 65M75, 65P10, 49Q22, 68T07

1. Introduction.

In this paper, we are concerned with solving the following Hamilton–Jacobi equation numerically,

$$\frac{\partial u(x,t)}{\partial t} + H(x, \nabla u(x,t)) = 0, \quad u(x,0) = g(x), \tag{1.1}$$

where $t \in [0,T], x \in \mathbb{R}^d$ with $d \in \mathbb{N}^+$, and the Hamiltonian H is convex with respect to the second variable. Hamilton-Jacobi partial differential equations (HJ PDEs) (1.1) arise in many areas of applications, including the calculus of variations, control theory, and differential games [1]. However, obtaining their analytical solutions, if at all possible, is often challenging, especially in high dimensions. As indispensable tools, numerical methods such as finite difference [12, 34], fast sweeping [41] and level set methods [25, 23] have been developed and refined over the years to approximate the solutions and predict their longtime dynamics. Those traditional algorithms involve discretizing the equation on grids and approximating the derivatives by using either finite difference or finite element techniques, and thus their applicability is limited by the so-called curse of dimensionality, namely the computational cost grows exponentially with respect to the problem dimension d [4].

In recent years, several strategies are proposed to mitigate the challenges caused by the curse of dimensionality when solving HJ PDEs numerically[1], including the optimization method [16, 10], sparse grids [6], neural networks [15, 21], etc. For

*

Funding: The research is partially supported by research grants NSF DMS-2307465 and ONR N00014-21-1-2891. The research of the first author is partially supported by the Hong Kong Research Grant Council ECS grant 25302822, GRF grant 15302823, NSFC grant 12301526, the internal grants (P0039016, P0045336, P0046811) from Hong Kong Polytechnic University and the CAS AMSS-PolyU Joint Laboratory of Applied Mathematics.

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. (jianbo.cui@polyu.edu.hk).

‡Department of Mathematics, University of California, Los Angeles, CA 90095, USA. (shuliu@math.ucla.edu).

§School of Mathematics, Georgia Tech, Atlanta, GA 30332, USA (hmzhou@math.gatech.edu).

[1]For more related topics and research problems on high dimensional HJ equations, we refer to http://www.ipam.ucla.edu/programs/long-programs/high-dimensional-hamilton-jacobi-pdes/?tab=activities.

instance, the authors in [18] proposed a probabilistic method based on the 2nd-order backward stochastic differential equation (SDE) to solve second-order HJ equations. A deep learning approach was then developed in [21] for Hamilton-Jacobi-Bellman (HJB) equations with the gradient acting as the policy function. The work [16] used the Hopf-Lax formula and split Bregman algorithm to solve HJ equation. For general state-dependent HJ equations, we refer to [9] for the numerical treatments via the coordinate descent algorithm and a generalized version of Hopf-Lax formula. In [35], the authors focused on the stationary HJ equation on bounded region via a special kind of Hopf-Lax formula and neural networks. In [15], the authors designed an architecture of deep neural network by imitating the structure of Hopf-Lax formula and then optimized its network parameters to acquire the solution of HJ equation. In [33], the authors focused on solving the high-dimensional HJB equation with quadratic kinetic energy. They reformulated the equation as an equivalent variational problem aiming to minimize discrepancies between the path measures of the controlled diffusion processes and the uncontrolled diffusion processes. In [31, 32], the authors proposed a causality-free algorithm to deal with the HJB equation originating from the optimal feedback control. The numerical solution is computed via minimizing the L^2 loss between the neural network approximation and the benchmark solution obtained by computing the optimal trajectories on randomly generated data points. In [30], by coupling with a continuity equation, the authors proposed a saddle point problem regarding the HJ equation, which is further solved via the primal-dual hybrid gradient algorithm.

In this paper, we introduce an alternative supervised learning method to solve HJ PDEs in high dimensions. Our study stems from some recent advancements in Wasserstein Hamiltonian flow (WHF) [8], which describes a family of PDEs defined on the Wasserstein manifold, the probability density set equipped with the optimal transport (OT) metric. Examples of WHFs include the Wasserstein geodesic [13], Schrödinger equation [14], and mean field control [28]. A typical WHF consists of a transport (or Fokker-Planck) equation and a HJ equation. Coupling two equations together, they form a geometric flow with symplectic and Hamiltonian structures on the Wasserstein manifold (we refer to section 2 for more details). This inspires us to design a numerical scheme that can preserve the geometric properties of the original equation and mitigate the curse of dimensionality at the same time.

To achieve this goal, we must confront several difficulties. First, the classical structure-preserving methods are often implicit in time and they become intractable when the spatial dimension grows high. Second, it is well-known that the characteristics of HJ equation may intersects and its classical solution may only exist up to a finite time. Third, the state-of-the-art numerical methods mainly focus on solving the viscosity solution, and may not capture the geometric structure on the Wasserstein density manifold. Last but not least, in some applications like the geometric optics, seismic waves and semi-classical limits of quantum dynamics, one may be more interested in other physical solutions, like the multi-valued solution and its statistical information [25].

To overcome these challenges, we leverage the geometric structure of WHF and the approximation power of deep neural networks (DNNs) to design a supervised learning procedure. More precisely, we propose an approach consisting of the following steps.

1. Coupling the given HJ equation with a continuity equation that transports a probability density function to form a WHF on Wasserstein manifold. The transport velocity field is provided by the solution of the HJ equation. According to the theory of WHF, a particle version corresponding to the coupled

system can be constructed leading to a system of Hamiltonian ordinary differential equations (ODEs).

2. Formulating a regression problem based on the Bregmann divergence following the OT theory. Its critical point satisfies the coupled system of WHF. This regression or its equivalent least squares expression are then used as the loss function in the learning process.
3. Generating the training data $(\{\boldsymbol{X}_t\}, \{\boldsymbol{P}_t\})$ by applying a symplectic integrator to the particle version of WHF, which is the Hamiltonian ODE system constructed in the first step.
4. Learning the solution HJ equation by reducing the loss function evaluated on the training data $(\{\boldsymbol{X}_t\}, \{\boldsymbol{P}_t\})$ via minimization algorithms such as Adam [26].

Details on the first and second steps will be given in section 2, and about the third and fourth steps in section 3.

The proposed method eases the computation burden of high dimensional HJ equation from three different aspects. (i) The loss function is expressed in term of expectation, which can be evaluated by employing the Monte Carlo integral methods and auto differentiation in DNNs. This allows us to carry out the calculation in higher dimensions without limiting the number of unknowns as the classical finite difference and finite element methods do. (ii) The training data $(\{\boldsymbol{X}_t\}, \{\boldsymbol{P}_t\})$ is generated by solving ODEs, which can be scaled up to higher dimensions. (iii) The density function can be selected (supervised) so that its support covers the region of interest. This provides a mechanism to only generate training data concentrated at the place where the solution of HJ equation is needed, and it is different from many existing DNN based methods for high dimensional PDEs, like physics-informed neural network (PINN) [36], deep-Ritz [17], or weak adversarial network [43], in which samples are usually taken everywhere in the domain. An added benefit is that the training data is computed by symplectic structure preserving schemes so that better geometric properties of the HJ equation can be retained in the learning procedure.

More importantly, we would like to advocate two new features of the proposed method for theoretical analysis. The coupling strategy enables us to develop a novel error bound using the residual estimate with respect to the density controlling where and how the training data is sampled. In other words, the error estimate may vary depending on the chosen density. This is different from the traditional error estimates, and it is more suitable for machine learning-based methods in which random samples are used for the training. We establish the rigorous error estimate for the proposed method in section 3. In a special case when the initial density is selected as the uniform distribution, the proposed method generates training data using ODEs that resemble the bi-characteristic formulation. According to the uniqueness theorem of ODEs, the training data can be generated beyond the blow-up time that the classic solution of HJ equation doesn't exist anymore, for example, the characteristics intersect. In this sense, the supervised learning method may compute the solution of HJ equation after the blow-up time. We show several such examples along with other numerical experiments in section 4.

Although our proposed approach shares some similarities with the supervised learning formulation presented in [31, 32], they have major differences too. The algorithm in [31, 32] is designed for the “backward” HJ equations originated from control with desirable terminal conditions, and the training data is generated by solving boundary value problems following the Pontryagin maximal principle. While our scheme is proposed for the “forward” HJ equation with given initial condition, and the training data is created by solving initial value Hamiltonian ODEs following particle formulation of WHF. More importantly, our derivation is conducted on the Wasserstein manifold, and it reveals the connection between the supervised learning scheme and a sup-inf problem originated from the mean-field control, which further provides a formulation for error analysis. It is also worth mentioning that the coupling idea is also used in [30], in which the solution of HJ equation is reformulated as a saddle point problem and further solved by the primal-dual hybrid gradient algorithm. In our scheme, we introduce a swarm of particles governed by the Hamiltonian ODEs corresponding to the WHF, and their trajectories are used as the data in the supervised learning. This leads to a minimization problem whose loss function can be computed by the Monte–Carlo method, and it is scalable to high-dimensional problems.

2. Density coupling strategy.

In this section, we introduce two key ingredients for designing the supervised learning scheme of HJ equations. One is coupling the HJ equation with a transport equation for the probability density to form a WHF on the Wesserstein manifold and its particle formulation. Another is connecting the coupled system to the critical point of a regression problem via the Bregman divergence.

2.1. Coupled Wasserstein Hamiltonian flow.

In this part, we introduce the density coupling strategy for (1.1). To explain it clearly, let us assume that the Hamiltonian $H:(x,p)\mapsto H(x,p)$ belongs to $\mathcal{C}^2(\mathbb{R}^d\times\mathbb{R}^d)$ and being strictly convex with respect to the second variable p for arbitrary fixed first variable x.

Suppose that the solution u of (1.1) exists and is smooth in time and space. Consider a random particle system $\{\boldsymbol{X}_t(\omega)\}_{t\in[0,T],\omega\in\Omega}$ defined on a complete probability space $(\Omega,\mathcal{F},P)$, satisfying the following ODE

$$\dot{\boldsymbol{X}}_t=\nabla_p H(\boldsymbol{X}_t,\nabla_x u(\boldsymbol{X}_t,t)),$$

where the initial value $\boldsymbol{X}_0$ obeys the probability distribution with the density function ρ_0 (denote $\boldsymbol{X}_0\sim\rho_0$ for simplicity). Then the probability density function $\rho(\cdot,t)$ of $\boldsymbol{X}_t$ satisfies

$$\partial_t\rho(x,t)+\nabla\cdot(\rho(x,t)\nabla_p H(x,\nabla u(x,t)))=0,\quad \rho(\cdot,0)=\rho_0, \tag{2.1}$$

which is a transport (continuity) equation. Let us consider the dynamics of the momentum defined by $\boldsymbol{P}_t(\omega)=\nabla_x u(\boldsymbol{X}_t(\omega),t)$. By taking the time derivative of $\boldsymbol{P}_t$, we get

$$\dot{\boldsymbol{P}}_t=\frac{\partial}{\partial t}\nabla_x u(\boldsymbol{X}_t,t)+\nabla_x^2 u(\boldsymbol{X}_t,t)\dot{\boldsymbol{X}}_t=\frac{\partial}{\partial t}\nabla_x u(\boldsymbol{X}_t,t)+\nabla_x^2 u(\boldsymbol{X}_t,t)\nabla_p H(\boldsymbol{X}_t,\nabla_x u(\boldsymbol{X}_t,t)), \tag{2.2}$$

where $\nabla_x^2 u(x,t)$ is the Hessian matrix of $u(x,t)$. If we differentiate (1.1) on both sides with respect to x, we have

$$\frac{\partial}{\partial t}\nabla_x u(x,t)+\nabla_x H(x,\nabla u(x,t))+\nabla_x^2 u(x,t)\nabla_p H(x,\nabla_x u(x,t))=0,\ \ \nabla_x u(\cdot,0)=\nabla g(x). \tag{2.3}$$

By setting $x=\boldsymbol{X}_t$ in (2.3) and substituting back into (2.2), we obtain that

$$\dot{\boldsymbol{P}}_t=-\nabla_x H(\boldsymbol{X}_t,\nabla_x u(\boldsymbol{X}_t,t))=-\nabla_x H(\boldsymbol{X}_t,\boldsymbol{P}_t).$$

To sum up, the coupled time-evolving probability density $\rho(\cdot,t)$ can be viewed as the probability density of the random particle $\boldsymbol{X}_t$ satisfying the Hamiltonian system

$$\begin{cases}\dot{\boldsymbol{X}}_t=\nabla_p H(\boldsymbol{X}_t,\boldsymbol{P}_t), & \boldsymbol{X}_0\sim\rho_0,\\ \dot{\boldsymbol{P}}_t=-\nabla_x H(\boldsymbol{X}_t,\boldsymbol{P}_t), & \boldsymbol{P}_0=\nabla g(\boldsymbol{X}_0).\end{cases} \tag{2.4}$$

Meanwhile, this density coupling strategy is related to the WHF introduced in [8]. More precisely, following the derivation provided in [14], we obtain a coupled system of PDEs corresponding to the particle system (2.4),

$$\partial_t \rho(x,t) + \nabla \cdot (\rho(x,t)\nabla_p H(x, \nabla \widehat{u}(x,t))) = 0, \quad \rho(\cdot,0) = \rho_0; \tag{2.5}$$

$$(\partial_t \widehat{u}(x,t) + H(x, \nabla \widehat{u}(x,t)))\rho(x,t) = 0, \ \ \widehat{u}(\cdot,0) = g(\cdot), \tag{2.6}$$

where $\widehat{u}(x,t) = u(x,t) + c(t)$ for any arbitrary $c(\cdot) \in \mathcal{C}^1([0,T])$. When $\rho(\cdot,t) > 0, t \in [0,T]$, (2.5)-(2.6) becomes a WHF. In particular, when $H(x,p) = |p|^2$, the coupled system (2.5)-(2.6) is the Wasserstein geodesic equation [42], which is the critical point of the Benamou-Brenier formula defining the OT distance on Wasserstein manifold [5].

This approach of coupling offers additional freedom in choosing the initial density ρ_0 which ultimately controls the support of the coupled density $\mathrm{Spt}(\rho(\cdot,t))$, hence where and how the samples $(\{\boldsymbol{X}_t\}, \{\boldsymbol{P}_t\})$ are drawn. At the same time, the Hamiltonian system (2.4) and Wasserstein Hamiltonian system (2.5)-(2.6) preserve the corresponding symplectic and Hamiltonian structures. As a by-product, solving (2.5)-(2.6) on $\mathrm{Spt}(\rho(\cdot,t))$, can recover the solution of original Hamiltonian–Jacobi equation (1.1) up to a spatial constant function. It should be noticed that the solution solved by (2.5)-(2.6) is consistent with the classical solution of (2.6) when $T < T_*$ with T_* being the first time that (2.6) develops a singularity. On the other hand, the Hamiltonian system (2.4) is always well-posed even if the PDE (2.6) does not admit classical solutions. This inspires us to design a new way to learn the solution of (1.1) even beyond the singularity.

2.2. Regression problem via Bregman divergence. To facilitate the learning process, we propose a minimization problem whose minimizer coincides with the solution of (1.1) up to a spatial constant function. A key observation as reported in [5, 42, 2, 8] and many more references therein indicates that if (2.5) and (2.6) admit the classical solution $\rho, \hat{u}$ on $[0,T]$, then $\rho, \hat{u}$ can be treated as the critical point of sup-inf problem given as

$$\sup_{\psi \in \mathcal{C}^1} \inf_{\widetilde{\rho} \in \mathcal{C}^1} \{\mathscr{I}_{\rho_0,\rho_T,T}(\widetilde{\rho}, \psi)\}, \tag{2.7}$$

where

$$\begin{aligned}\mathscr{I}_{\rho_0,\rho_b,T}(\widetilde{\rho},\psi) = &\int_0^T \int_{\mathbb{R}^d} -(\partial_t \psi(x,t) + H(x,\nabla\psi(x,t)))\widetilde{\rho}(x,t)\ dxdt \\ &+ \int_{\mathbb{R}^d} \psi(x,T)\rho_T(x)\ dx - \int_{\mathbb{R}^d} \psi(x,0)\rho_0(x)\ dx.\end{aligned} \tag{2.8}$$

This formulation originates from the optimal transport associated with the initial density $\rho_0 = \rho(\cdot,0)$ and target $\rho_T = \rho(\cdot,T)$. Here we use $\widetilde{\rho}$ as variable of the functional so as to distinguish it from the solution ρ to the continuity equation (2.5).

Consequently, we can solve (2.7) instead of directly dealing with the PDE system (2.5) and (2.6). We recall that the optimal density $\widetilde{\rho}$ of (2.7) is exactly the classical solution in (2.5). This suggests that (2.7) can be rewrite as the following optimization only associated with the variable ψ if we directly replace $\widetilde{\rho}$ in (2.8) by the optimal density ρ,

$$\sup_{\psi \in \Psi} \{\mathscr{L}_{\rho_0,g,T}(\psi)\}, \tag{2.9}$$

where

$$\mathscr{L}_{\rho_0,g,T}(\psi) = \int_0^T \int_{\mathbb{R}^d} -\left(\partial_t \psi(x,t) + H(x, \nabla\psi(x,t))\right)\rho_t(x)\ dxdt$$
$$+ \int_{\mathbb{R}^d} \psi(x,T)\rho_T(x)\ dx - \int_{\mathbb{R}^d} \psi(x,0)\rho_0(x)\ dx. \quad (2.10)$$

We want to point out that in the standard OT formulation, the terminal density ρ_T is given independently. This is different in the coupled system considered here. Since $\rho(x,t)$ is the probability density of $\boldsymbol{X}_t$ given by the Hamiltonian system (2.4) on $[0,T]$, which is uniquely determined by the initial conditions ρ_0 and g. It implies that $\rho_T = \rho(x,T)$ is also determined by ρ_0 and g. For this reason, we use notation $\mathscr{L}_{\rho_0,g,T}(\psi)$ in (2.10) to emphasize the dependence on g. It can be checked that $\mathscr{L}_{\rho_0,g,T}(\psi+c) = \mathscr{L}_{\rho_0,g,T}(\psi)$ for any continuous in time and constant in space function $c \in \mathcal{C}^1([0,T]\times\mathbb{R}^d)$. Thus it suffices to consider (2.10) over the equivalent class $[\psi]$ of $\psi \in \mathcal{C}^1([0,T]\times\mathbb{R}^d)$ up to a spatial constant function. We denote this set of equivalent class by Ψ. In addition, if we denote μ_t as the joint probability distribution of $(\boldsymbol{X}_t, \boldsymbol{P}_t)$ solved from the Hamiltonian system (2.4) for $0 \le t \le T$, $\rho(\cdot,t)$ is the density of the $\boldsymbol{X}$-marginal distribution of μ_t. To further simplify the expression of (2.10), we use the concept of Bregman divergence.

DEFINITION 2.1 (Bregman divergence [7]). *Suppose $f \in \mathcal{C}^1(\mathbb{R}^d)$ is a strict convex function. We define the Bregman divergence $D_f(\cdot:\cdot): \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ induced by f as*

$$D_f(q_1 : q_2) = f(q_1) - f(q_2) - \nabla f(q_2)\cdot(q_1 - q_2).$$

It is known that the Bregman divergence is positive and $D_f(q_1 : q_2) = 0$ if and only if $q_1 = q_2$. Denote H^* as the *Legendre Transform* of the given Hamiltonian $H(x,p)$ with respect to p, i.e., $H^*(x,v) \triangleq \sup_{p\in\mathbb{R}^d}\{v\cdot p - H(x,p)\}$ for any fixed $x \in \mathbb{R}^d, v \in \mathbb{R}^d$. Since $H \in \mathcal{C}^1(\mathbb{R}^{2d})$ is strictly convex with respect to p for arbitrary x, $H^*(x,v)$ is also strictly convex with respect to v. And both $\nabla_p H(x,\cdot)$ and $\nabla_v H^*(x,\cdot)$ are invertible for arbitrary $x \in \mathbb{R}^d$.

LEMMA 2.1. *Suppose $f \in \mathcal{C}^2(\mathbb{R}^d)$ is α-strongly convex and L-strongly smooth ($\alpha, L > 0$), i.e., $\alpha I_d \preceq \nabla^2 f(q) \preceq L I_d$ for any $q \in \mathbb{R}^d$. Then, the Legendre transform f^* of f belongs to $\mathcal{C}^2(\mathbb{R}^d)$, and is $\frac{1}{L}-$strongly convex and $\frac{1}{\alpha}$-strongly smooth on $\mathbb{R}^d$. Furthermore, it holds that*

$$f(q) + f^*(p) - q\cdot p = D_f(q : \nabla f^*(p)) = D_{f^*}(p : \nabla f(q)).$$

LEMMA 2.2. *Suppose that $T > 0$ is the given terminal time, and that the Hamiltonian $H \in \mathcal{C}^1(\mathbb{R}^d\times\mathbb{R}^d)$ is strongly convex with respect to the momentum p for any $x \in \mathbb{R}^d$. Assume $\rho_0 \in \mathcal{C}^1(\mathbb{R}^d)$ and $g \in \mathcal{C}^1(\mathbb{R}^d)$. Then*

$$\mathscr{L}_{\rho_0,g,T}(\psi) = -\int_0^T \int_{\mathbb{R}^{2d}} D_{H,x}(\nabla\psi(x,t) : p)\ d\mu_t(x,p)dt + \int_0^T \int_{\mathbb{R}^{2d}} H^*(x, \nabla_p H(x,p))\ d\mu_t(x,p)dt, \quad (2.11)$$

where we denote $D_{H,x}(q_1 : q_2) = D_{H(x,\cdot)}(q_1 : q_2)$, i.e., $D_{H,x}$ is the x-dependent Bregman divergence regarding $H(x,\cdot)$.

The proof of Lemma 2.1 uses some standard arguments which are common in the convex optimization. The proof of Lemma 2.2 is done by direct calculation of $\mathscr{L}_{\rho_0,g,T}(\psi)$ and Lemma 2.1. For completeness, we provide the proofs in the supplementary material. The second term on the right-hand side of (2.11) does not involve

ψ, and thus can be treated as a constant, which implies that the original optimization (2.10) is equivalent to the following regression,

$$\min_{\psi\in\Psi}\left\{\int_0^T \int_{\mathbb{R}^{2d}} D_{H,x}(\nabla\psi(x,t) : p)\ d\mu_t(x,p)dt\right\}. \tag{2.12}$$

As we know that $\mu_t(x,p)$ can be conveniently sampled according to the Hamiltonian ODEs (2.4), and by the Fubini's theorem, we can reformulate (2.12) as

(2.13)

$$(D_H\text{-Regression})\ \min_{[\psi]\in\Psi}\left\{\mathscr{L}^{D_{H,x}}_{\rho_0,g,T}(\psi)\right\}, \mathscr{L}^{D_{H,x}}_{\rho_0,g,T}(\psi) \triangleq \mathbb{E}_\omega\left[\int_0^T D_{H,x}(\nabla\psi(\boldsymbol{X}_t(\omega),t) : \boldsymbol{P}_t(\omega))dt\right].$$

This functional matches the gradient $\nabla\psi(\boldsymbol{X}_t,t)$ to the momentum $\boldsymbol{P}_t$ with respect to the Bregman divergence induced by the Hamiltonian H. And it can be approximated by the Monte–Carlo method once the samples are available. We use it as the loss in the supervised learning and discuss its details in section 3.1.

We may also replace the $D_{H,x}$ by the quadratic distance $|\cdot|^2$. This does not weaken the performance of the original problem (2.12) since $D_{H,x}(q_1 : q_2) \approx \frac{1}{2}(q_1 - q_2)^\top\nabla^2 H(q_2)(q_1-q_2)$ for sufficiently close q_1, q_2. For this reason, we also propose the following least squares problem as the loss function in our algorithm, which may make the training easier.

(2.14)

$$(\text{Least Squares})\ \min_{[\psi]\in\Psi}\left\{\mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}(\psi)\right\}, \mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}(\psi) \triangleq \mathbb{E}_\omega\left[\int_0^T |\nabla\psi(\boldsymbol{X}_t(\omega),t) - \boldsymbol{P}_t(\omega)|^2\ dt\right].$$

PROPOSITION 2.1. *Suppose $H(x,p) = \frac{1}{2}|p|^2 + V(x)$. Then $D_{H,x}(q_1 : q_2) = \frac{1}{2}|q_1-q_2|^2$, and the corresponding regression* (2.13) *is equivalent to the least squares formulation* (2.14)*.*

Further discussion regarding this least squares problem and its related algorithm is provided in section 3.1. Next, we give a consistency result on the regression problem (2.13) whose proof can be found in the supplementary material.

THEOREM 2.1 (Consistency). *Suppose the Hamiltonian $H \in \mathcal{C}^1(\mathbb{R}^d\times\mathbb{R}^d)$ satisfies the conditions that $\nabla_x H, \nabla_p H$ are Lipschitz, and that H is strictly convex with respect to p for any fixed $x\in\mathbb{R}^d$. Assume that $\widehat{\psi}\in\mathcal{C}^2(\mathbb{R}^d\times[0,T])$ satisfies $\mathscr{L}^{D_{H,x}}_{\rho_0,g,T}(\widehat{\psi}) = 0$, then $\widehat{\psi}$ solves the following gradient-version of the Hamilton-Jacobi equation*

(2.15)

$$\nabla\left(\frac{\partial}{\partial t}\widehat{\psi}(x,t) + H(x,\nabla\widehat{\psi}(x,t))\right) = 0, \quad \textit{at } (x,t)\in\mathbb{R}^d\times(0,t] \textit{ with } x\in \mathrm{Spt}(\rho_t);$$

$$\textit{and } \nabla\widehat{\psi}(x,0) = \nabla g(x) \quad \textit{with any } x\in Spt(\rho_0).$$

Similarly, $\widehat{\psi}$ also solves (2.15) *if $\mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}(\widehat{\psi}) = 0$.*

Proof. Given the Lipschitz condition on the vector field $(\nabla_x H^\top, \nabla_p H^\top)^\top$, it is known that the underlying Hamiltonian system considered admits a unique solution with continuous trajectories a.s. for arbitrary initial condition $(\boldsymbol{X}_0, \nabla u(\boldsymbol{X}_0))$.

Let us recall the probability space $(\Omega, \mathcal{F}, P)$ used to describe the randomness of the Hamiltonian system. Since

$$\mathbb{E}_\omega \left[\int_0^T D_H(\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega), t) : \boldsymbol{P}_t(\omega)) \; dt \right] = 0,$$

then by the fact that Bregman divergence D_H is always non-negative, we obtain

$$\int_0^T D_H(\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega), t) : \boldsymbol{P}_t(\omega)) \; dt = 0, \quad P - \text{almost surely}.$$

Thus, there exists a measurable subset $\Omega' \subset \Omega$ with $P(\Omega') = 1$ such that

$$\int_0^T D_H(\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) : \boldsymbol{P}_t(\omega')) \; dt = 0, \quad \forall \; \omega' \in \Omega'.$$

By using the continuity and non-negativity (Definition 2.1) of $D_H(\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) : \boldsymbol{P}_t(\omega'))$ with respect to t, we have

$$\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) = \boldsymbol{P}_t(\omega') \quad \text{for} \;\; 0 \le t \le T. \tag{2.16}$$

When $t = 0$, we have $\nabla\widehat{\psi}(\boldsymbol{X}_0(\omega'), 0) = \boldsymbol{P}_0(\omega')$. Recall the initial condition of the Hamiltonian System, we have $\boldsymbol{P}_0(\omega') = \nabla g(\boldsymbol{X}_0(\omega'))$. This yields $\nabla\widehat{\psi}(\boldsymbol{X}_0(\omega'), 0) = \nabla g(\boldsymbol{X}_0(\omega'))$ for any $\omega' \in \Omega'$, which yields

$$\nabla\widehat{\psi}(x, 0) = \nabla g(x) \quad \text{for all } x \in \mathrm{Spt}(\rho_0). \tag{2.17}$$

On the other hand, for $t \in (0, T]$, by differentiating on both sides of (2.16) w.r.t. t, we obtain

$$\frac{\partial}{\partial t}\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) + \nabla^2\widehat{\psi}(\boldsymbol{X}_t(\omega'), t)\dot{\boldsymbol{X}}_t(\omega') = \dot{\boldsymbol{P}}_t(\omega'). \tag{2.18}$$

Recall that we have

$$\begin{aligned} \dot{\boldsymbol{X}}_t &= \nabla_p H(\boldsymbol{X}_t, \boldsymbol{P}_t) = \nabla_p H(\boldsymbol{X}_t, \nabla\widehat{\psi}(\boldsymbol{X}_t, t)), \\ \dot{\boldsymbol{P}}_t &= -\nabla_x H(\boldsymbol{X}_t, \boldsymbol{P}_t) = -\nabla_x H(\boldsymbol{X}_t, \nabla\widehat{\psi}(\boldsymbol{X}_t, t)). \end{aligned}$$

Plugging these into (2.18) yields

$$\begin{aligned} \frac{\partial}{\partial t}\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) + \nabla^2\widehat{\psi}(\boldsymbol{X}_t(\omega'), t)\nabla_p H(\boldsymbol{X}_t(\omega'), \nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t)) \\ = -\nabla_x H(\boldsymbol{X}_t(\omega'), \nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t)), \end{aligned}$$

which leads to

$$\nabla \left(\frac{\partial}{\partial t}\widehat{\psi}(\boldsymbol{X}_t(\omega'), t) + H(x, \nabla\widehat{\psi}(\boldsymbol{X}_t(\omega'), t)) \right) = 0, \quad \forall \; \omega' \in \Omega'.$$

Since the probability density distribution of $\boldsymbol{X}_t$ is ρ_t, we have proved that

$$\nabla \left(\frac{\partial}{\partial t}\widehat{\psi}(x, t) + H(x, \nabla\widehat{\psi}(x, t)) \right) = 0, \quad \forall \;\; x \in \mathrm{Spt}(\rho_t). \tag{2.19}$$

Combining (2.17) and (2.19) proves this theorem.

On the other hand, if $\mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}(\widehat{\psi}) = 0$. By using the fact that $|\nabla\widehat{\psi}(\boldsymbol{X}_t(\omega),t) - \boldsymbol{P}_t(\omega)|^2$ is continuous and non-negative for a.s. $\omega \in \Omega$, we can repeat the previous proof to show the same assertion still holds. ∎

REMARK 2.1. *We would like to point out that the solution of dynamical ODEs* (2.4)*, and both definitions of the regression* (2.13) *and least square problems* (2.14) *can exist even after the singularity formation in the solution of HJ equation* (1.1)*. This means that we can use the proposed method to compute the minimizers beyond the singularity time. An interesting question is what solution the proposed method computes. To answer it, Theorem 2.1 may give us some hints as it can be used to define a weak solution of HJ equation in the following sense. By swapping the integrals in $\mathscr{L}^{D_{H,x}}_{\rho_0,g,T}$, it holds that*

$$\begin{aligned}\mathscr{L}^{D_{H,x}}_{\rho_0,g,T}(\psi) &= \int_0^T \int_{\mathbb{R}^d} D_{H,x}(\nabla\psi(x,t) : p)\ d\mu_t(x,p)dt \\ &= \int_0^T \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} D_{H,x}(\nabla\psi(x,t) : p)\ d\mu_t(p|x) \right)\ \rho_t(x)dx\ dt.\end{aligned}$$

The minimizer $\widehat{\psi}$ of $\mathscr{L}^{D_{H,x}}_{\rho_0,g,T}$ can be viewed as a weak solution of the HJ equation since taking the first variation on ψ leads to

$$-\nabla \cdot \left(\rho_t(x) \left(\int_{\mathbb{R}^d} \nabla_{q_1} D_{H,x}(\nabla\widehat{\psi}(x,t) : p)\ d\mu_t(p|x) \right) \right) = 0.$$

Here $\nabla_{q_1} D_{H,x}(\cdot : \cdot)$ is the partial derivative with respect to the first variable q_1 of $D_{H,x}(q_1 : q_2)$. In particular, if $H(x,p) = \frac{1}{2}|p|^2 + V(x)$, the minimizer of $\mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}$ solves the following elliptic equation

$$-\nabla \cdot (\rho_t(x)(\nabla\widehat{\psi}(x,t) - \bar{p}(x,t))) = 0. \quad \textit{where } \bar{p}(x,t) = \int_{\mathbb{R}^d} p\ d\mu_t(p|x). \quad \textit{for } t \in [0,T]. \tag{2.20}$$

To sum up, in the proposed regression problem, $\nabla\widehat{\psi}$ can be viewed as the orthogonal (with respect to the $L^2(\rho_t)$ inner product) projection of the $\mu_t(\cdot|x)$-weighted momentum $\bar{p}(x,t)$ to the space of gradient fields.

This definition comes with several benefits. On the one hand, Theorem 2.1 verifies that the minimizer $\widehat{\psi}$ solves the HJ equation (2.15) *in the strong sense (in the gradient form) before the time T_* that the classical solution develops caustics. On the other hand, the lifetime of the minimizer $\widehat{\psi}$ of $\mathscr{L}^{|\cdot|^2}_{\rho_0,g,T}$ goes beyond T_* since the conditional distribution $\mu_t(\cdot|x)$ on momentum is not based on the Dirac type function centered at certain positions x. Although the minimizer may be multi-valued and has information about which mono-momentum to match with, we treat $\widehat{\psi}$ as the $\mu_t(\cdot|x)$-weighted "solution" associated with the Hamilton-Jacobi equation* (1.1) *in this paper. However, how to theoretically understand the numerical solution after the singularity remains as an open question, which is beyond the scope of this paper. Furthermore, by modifying the cost functional in the regression problem, one may construct different types of weak solutions of HJ equations. This is another topic that deserves further investigation and careful discussion.*

3. Supervised learning scheme via density coupling.

In this section, we present the supervised learning scheme based on the density coupling strategy and the regression formulation (2.13).

3.1. Algorithm.

Our method for computing the Hamilton-Jacobi equation (1.1) associated with the probability density distribution ρ_0 consists of the following two main steps.

- (Generating sample trajectories on phase space) Sample N particles $\{x_0^{(k)}\}_{k=1}^N$ from ρ_0 with momentum $p_0^{(k)} = \nabla g(x_0^{(k)})$, and apply a suitable geometric integrator to solve the Hamiltonian system

$$\begin{aligned} \dot{x}_t^{(k)} &= \nabla_p H(x_t^{(k)}, p_t^{(k)}) \\ \dot{p}_t^{(k)} &= -\nabla_x H(x_t^{(k)}, p_t^{(k)}) \end{aligned} \quad \text{with initial condition } (x_0^{(k)}, \nabla g(x_0^{(k)})). \tag{3.1}$$

  at time steps $t_i = ih$, with $h = \frac{T}{M}$, $1 \le i \le M$ for each $k \in \{1, 2, ..., N\}$. We denote the numerical solutions at t_i as $\{(\tilde{x}_{t_i}^{(k)}, \tilde{p}_{t_i}^{(k)})\}$, $1 \le k \le N$.
- (Compute ψ via supervised learning) Set up the neural network $\psi_\theta : \mathbb{R}^d \times [0, T] \to \mathbb{R}$, and minimize the sum of average discrepancies between each $\nabla_x \psi_\theta(\tilde{x}_{t_i}^{(k)}, t_i)$ and $\tilde{p}_{t_i}^{(k)}$ at each time step t_i evaluated on a random batch $\{\widetilde{x}^{(k_j)}\}_{j=1}^{N_0} \subset \{\widetilde{x}^{(k)}\}$ with batchsize N_0. More precisely, we denote

$$\text{Loss}(\theta) = \frac{1}{M} \sum_{i=1}^{M} \left(\frac{1}{N_0} \sum_{k=1}^{N_0} D_{H, \tilde{x}_{t_i}^{(k)}} (\nabla_x \psi_\theta(\tilde{x}_{t_i}^{(k)}, t_i) : \tilde{p}_{t_i}^{(k)}) \right). \tag{3.2}$$

  We apply stochastic gradient descent algorithms such as Adam's method [26] to minimize $\text{Loss}(\theta)$ with respect to the parameter θ in ψ_θ. We summarize our method in Algorithm 3.1.

Algorithm 3.1 Computing the gradient field of Hamilton-Jacobi equation (1.1) associated with initial density function ρ_0.

Set up neural network $\psi_\theta : \mathbb{R}^d \times [0, T] \to \mathbb{R}$;
Sample $\{x_0^{(k)}\}_{k=1}^N$ from ρ_0;
Apply a suitable geometric integrator to solve the Hamiltonian system (3.1) with initial condition $x_0 = x_0^{(k)}, p_0 = \nabla g(x_0^{(k)})$ to obtain the trajectory $(\tilde{x}_{t_i}^{(k)}, \tilde{p}_{t_i}^{(k)})$ at time steps $0 \le t_1 \le \cdots \le t_M = T$ for each k, $1 \le k \le N$.
for $\text{Iter} = 0$ to N_{Iter} **do**
    Pick random batch with size $N_0 \le N$ from $\{\widetilde{x}^{(k)}\}$;
    Evaluate $\text{Loss}(\theta)$ defined as in (3.3);
    Apply Adam's method with learning rate lr to perform gradient descent $\theta \leftarrow \theta - lr\ \nabla_\theta \text{Loss}(\theta)$;
    if $\text{Loss}(\theta) \le err_0$ **then**
        break;
    end if
end for
$\nabla_x \psi_\theta(\cdot, t)$ $(0 \le t \le T)$ is the computed gradient field of the Hamilton-Jacobi equation (1.1).

In our algorithm, we have the freedom to choose the geometric integrator to discretize the Hamiltonian system (2.4). There are various choices such as symplectic Runge–Kutta schemes, symplectic partitioned Runge–Kutta Methods, Strömer–Verlet scheme, etc. We refer interested readers to [20] and references therein for further details. Such structure-preserving methods could preserve the properties, such as symplectic structure and quadratic conservative quantities, of the original system as much as possible [14].

A few observations have been made during our implementation of the proposed algorithm.

First, Theorem 2.1 suggests that both the regression problem (2.13) and (2.14) are consistent with respect to equation (2.15). However, in practice, to perform the supervised learning in an efficient and stable way, one needs to avoid the case in which the Hessian (with respect to p) of the Hamiltonian H possesses a large conditional number. We adopt the least squares regression (2.14) and use the quadratic loss (3.3) instead of D_H loss in (3.2) in our implementation,

$$\text{Loss}(\theta) = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N_0}\sum_{k=1}^{N_0}|\nabla_x\psi_\theta(\tilde{x}_{t_i}^{(k)}, t_i) - \tilde{p}_{t_i}^{(k)}|^2\right). \tag{3.3}$$

Second, it may be difficult for a single neural network to learn the solution on the entire time interval $[0, T]$, especially when T is large or when the solution experiences large-scale oscillations. In such cases, in order to improve the performance of our method, we split the time interval $[0, T]$ into smaller sub-intervals, train different ψ_θ on each sub-interval respectively, and then concatenate the solution together. We refer the reader to section 4.2.1 for further details.

Third, we may re-sample the points $\{x_0^{(k)}\}_{1\leq k\leq N}$ from ρ_0 and repeat the procedure in each training iteration to update θ. According to our experience, such a strategy produces numerical solutions with similar quality compared to that computed by the method with fixed samples throughout the simulations.

3.2. Bound on the residual . In this part, we estimate the density weighted residual of the numerical solution ψ_θ produced from the proposed algorithm. Let us denote $\tilde{\Phi}_h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ as the solution map of the chosen geometric integrator for (2.4), and

$$(\tilde{x}_{t_i}, \tilde{p}_{t_i}) = \tilde{\Phi}_h^{(i)}(x_0, \nabla g(x_0)) \triangleq \underbrace{\tilde{\Phi}_h \circ \cdots \circ \tilde{\Phi}_h}_{i\ \tilde{\Phi}_h\text{s composing together}} (x_0, \nabla g(x_0)),$$

where the stepsize $h = \frac{T}{M}$, $(\tilde{x}_{t_i}, \tilde{p}_{t_i})$ is the *numerical solution* solved at time $t_i = ih$ with initial condition x_0 and $p_0 = \nabla g(x_0)$. We denote $\tilde{\rho}_{t_i}$ the probability density of random variable $\tilde{x}_{t_i}$. Let $r \geq 2$ be the order of the local truncation error of numerical solver $\tilde{\Phi}_h$[2]. Correspondingly, we denote $\Phi_t : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ as the flow map of the

[2] i.e., suppose (x_h, p_h) is the exact solution of (3.5) with initial condition (x_0, p_0) after one time step h, then

$$|\tilde{\Phi}_h(x_0, p_0) - (x_h, p_h)| = C_{\tilde{\Phi}_h}(x_0, p_0)h^r, \tag{3.4}$$

where $C_{\tilde{\Phi}_h}((x_0, p_0))$ is a constant only depending on the Hamiltonian H, the initial condition (x_0, p_0), and the numerical scheme.

Hamiltonian system

$$\dot{x}_t = \nabla_p H(x_t, p_t), \quad \dot{p}_t = -\nabla_x H(x_t, p_t), \tag{3.5}$$

i.e., $\Phi_t((x_0, p_0)) = (x_t, p_t)$ for $t \in [0, T]$.

For the given approximation ψ_θ to the solution of the Hamilton–Jacobi equation, we consider the loss vector of the supervised learning at each sample point as

$$e_{t_i}^{(k)} = \nabla \psi_\theta(\tilde{x}_{t_i}^{(k)}, t_i) - \tilde{p}_{t_i}^{(k)}. \tag{3.6}$$

Let us set

$$\varepsilon_i^N = \frac{1}{N} \sum_{k=1}^{N} |e_{t_i}^{(k)}| \quad \text{and} \quad \delta_i^{N,h} = \frac{1}{N} \sum_{k=1}^{N} \frac{|e_{t_{i+1}}^{(k)} - e_{t_i}^{(k)}|}{h} \tag{3.7}$$

as the empirical average of the training loss and its difference quotient at time node t_i, respectively. We note that when $\nabla\psi_\theta$ is Lipschitz on the support of the probability density function, $e_{t_i}^{(k)}$ is continuous with respect to t_i along (3.5). In particular, if there is no training error (i.e., $e_{t_i}^{(k)} = 0$), we have $\varepsilon_i^N = \delta_i^{N,h} = 0$. Our estimate on the L^1-residual of $\nabla\psi_\theta$ is presented in the next theorem.

THEOREM 3.1 (Posterior estimation on L^1 residual of Hamilton-Jacobi equation). *Suppose that $\frac{\partial H}{\partial p}$ and $\frac{\partial H}{\partial x}$ are Lipschitz with constants L_1 and L_2 respectively, the initial distribution ρ_0 has a compact support, $\epsilon \in (0,1)$ is a given constant, M is large enough such that $M \geq \max\{T, \frac{T}{2}(L_1 + L_2)e^{L_1+L_2}\}$, and the time stepsize is taken as $h = \frac{T}{M}$. Assume that the neural network ψ_θ is trained by minimizing the loss* (3.3) *with data generated by a numerical integrator of order r for* (3.1) *with initial samples $\{x_{t_0}^{(k)}\}_{k=1}^N$ drawn from ρ_0. Then with probability $1-\epsilon$, ψ_θ satisfies*

$$\begin{aligned}
&\int_{\mathbb{R}^d} \left| \nabla \left(\frac{\partial}{\partial t} \psi_\theta(x, t_i) + H(x, \nabla\psi_\theta(x, t_i)) \right) \right| \tilde{\rho}_{t_i}(x) dx \\
&\leq \frac{1}{2}\lambda(\theta, i)h + \eta(\theta, i)h^{r-1} + \delta_i^{N,h} + \nu(\theta, i)\varepsilon_i^N + R(\theta, i)\sqrt{\frac{\ln M + \ln\frac{2}{\epsilon}}{2N}}, \qquad (3.8)
\end{aligned}$$

at $t_i = ih$, $i = 1, \ldots, M$. Here, $\lambda(\theta, i), \eta(\theta, i), \nu(\theta, i), R(\theta, i)$ are non-negative constants depending on the parameter θ, time node t_i, Hamiltonian H, initial distribution ρ_0, and numerical scheme $\tilde{\Phi}_h$.

Proof. Let us focus on the k-th trajectory $\{(\tilde{x}_{t_i}^{(k)}, \tilde{p}_{t_i}^{(k)})\}_{i=0}^M$. At time node t_i, $i \leq M-1$, we denote

$$(\widehat{x}_\tau^{(k)}, \widehat{p}_\tau^{(k)}) = \Phi_\tau(\tilde{x}_{t_i}^{(k)}, \tilde{p}_{t_i}^{(k)}), \quad \tau \geq 0.$$

For simplicity, we omit the superscript (k) of each $(\tilde{x}_{t_i}^{(k)}, \tilde{p}_{t_i}^{(k)})$, $(x_t^{(k)}, p_t^{(k)})$, $(\widehat{x}_\tau^{(k)}, \widehat{p}_\tau^{(k)})$ and $e_i^{(k)}$. We start by considering

$$\nabla\psi_\theta(\tilde{x}_{t_{i+1}}, t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_i}, t_i) = \tilde{p}_{t_{i+1}} - \tilde{p}_{t_i} + (e_{i+1} - e_i) \tag{3.9}$$

The left-hand side of (3.9) can be recast as

$$(\nabla\psi_\theta(\widehat{x}_h, t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_i}, t_i)) + (\nabla\psi_\theta(\tilde{x}_{t_{i+1}}, t_{i+1}) - \nabla\psi_\theta(\widehat{x}_h, t_{i+1})),$$

where the first term can be formulated as

$$\nabla\psi_\theta(\widehat{x}_h, t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_i}, t_i) = \int_0^h \frac{d}{d\tau}\nabla\psi_\theta(\widehat{x}_\tau, t_i+\tau)\, d\tau$$
$$= \int_0^h \nabla^2\psi_\theta(\widehat{x}_\tau, t_i+\tau)\frac{\partial}{\partial p}H(\widehat{x}_\tau, \widehat{p}_\tau) + \frac{\partial}{\partial t}\nabla\psi_\theta(\widehat{x}_\tau, t_i+\tau)\, d\tau.$$

For the second equality, we recall that $\dot{\widehat{x}}_\tau = \frac{\partial}{\partial p}H(\widehat{x}_\tau, \widehat{p}_\tau)$.

On the other hand, the right-hand side of (3.9) can be formulated as

$$(\widehat{p}_h - \tilde{p}_{t_i}) + (\tilde{p}_{t_{i+1}} - \widehat{p}_h) + (e_{i+1} - e_i),$$

where the first term can be rewritten as

$$\widehat{p}_h - \tilde{p}_{t_i} = \int_0^h \dot{\widehat{p}}_\tau\, d\tau = \int_0^h -\frac{\partial}{\partial x}H(\widehat{x}_\tau, \widehat{p}_\tau)\, d\tau.$$

Combining the previous calculations, we obtain

$$\int_0^h \frac{\partial}{\partial t}\nabla\psi_\theta(\widehat{x}_\tau, t_i+\tau) + \nabla^2\psi_\theta(\widehat{x}_\tau, t_i+\tau)\frac{\partial}{\partial p}H(\widehat{x}_\tau, \widehat{p}_\tau) + \frac{\partial}{\partial x}H(\widehat{x}_\tau, \widehat{p}_\tau)\, d\tau$$
$$= (\nabla\psi_\theta(\widehat{x}_h, t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_{i+1}}, t_{i+1})) + (\tilde{p}_{t_{i+1}} - \widehat{p}_h) + (e_{i+1} - e_i). \tag{3.10}$$

We estimate the distance between $\widehat{x}_\tau$ and $\widehat{x}_0 = \tilde{x}_{t_i}$ by considering

$$|\widehat{x}_\tau - \widehat{x}_0| \le \int_0^\tau |\frac{\partial}{\partial p}H(\widehat{x}_s, \widehat{p}_s)|\, ds \le \int_0^\tau |\frac{\partial}{\partial p}H(\widehat{x}_0, \widehat{p}_0)| + |\frac{\partial}{\partial p}H(\widehat{x}_0, \widehat{p}_0) - \frac{\partial}{\partial p}H(\widehat{x}_s, \widehat{p}_s)|\, ds$$
$$\le \tau|\frac{\partial}{\partial p}H(\widehat{x}_0, \widehat{p}_0)| + L_1\int_0^\tau |\widehat{x}_s - \widehat{x}_0| + |\widehat{p}_s - \widehat{p}_0|\, ds, \tag{3.11}$$

where the second inequality is due to the Lipschitz property of $\frac{\partial H}{\partial p}$. Similarly, for $\widehat{p}_\tau$ and $\widehat{p}_0 = \tilde{p}_{t_i}$, we have

$$|\widehat{p}_\tau - \widehat{p}_0| \le \int_0^\tau |-\frac{\partial}{\partial x}H(\widehat{x}_s, \widehat{p}_s)|\, ds \le \tau|\frac{\partial}{\partial x}H(\widehat{x}_0, \widehat{p}_0)| + L_2\int_0^\tau |\widehat{x}_s - \widehat{x}_0| + |\widehat{p}_s - \widehat{p}_0|\, ds \tag{3.12}$$

By adding (3.11) and (3.12) and applying the Grönwall's inequality, we obtain

$$|\widehat{x}_\tau - \tilde{x}_{t_i}| + |\widehat{p}_\tau - \tilde{p}_{t_i}| \le (|\frac{\partial}{\partial p}H(\tilde{x}_{t_i}, \tilde{p}_{t_i})| + |\frac{\partial}{\partial x}H(\tilde{x}_{t_i}, \tilde{p}_{t_i})|) \tag{3.13}$$
$$\times\left(\tau + \frac{e^{(L_1+L_2)\tau} - (L_1+L_2)\tau - 1}{L_1+L_2}\right),$$

From the Lipschitz property and the inequality $e^x \le 1 + x + \frac{1}{2}e^x x^2$ for $x \ge 0$, the right hand side of (2) can be further bounded by

$$\Big((L_1+L_2)(|\tilde{x}_{t_i}| + |\tilde{p}_{t_i}|) + (|\partial_p H(0,0)| + |\partial_x H(0,0)|)\Big)\left(\tau + \frac{1}{2}e^{(L_1+L_2)\tau}(L_1+L_2)\tau^2\right).$$

Let us denote $R_{t_i} = \max_{1\le k\le N}\{|\tilde{x}_{t_i}^{(k)}| + |\tilde{p}_{t_i}^{(k)}|\}$, $L = L_1 + L_2$ and $C = |\partial_p H(0,0)| + |\partial_x H(0,0)|$. Since we assume that

$$M \ge \max\{T, \frac{T}{2}(L_1+L_2)e^{L_1+L_2}\},$$

the time stepsize

$$h \le \frac{T}{M} \le \min\{1, \frac{2}{L_1+L_2} e^{-(L_1+L_2)}\}.$$

Then for $0 \le \tau \le h$, we have $\frac{1}{2} e^{(L_1+L_2)\tau}(L_1+L_2)\tau^2 \le \frac{1}{2} e^{Lh} Lh \cdot \tau \le \tau$. Thus, (2) can be bounded by

$$|\widehat{x}_\tau - \tilde{x}_{t_i}| + |\widehat{p}_\tau - \tilde{p}_{t_i}| \le 2(LR_{t_i} + C + 1)\tau.$$

Denote the time-space region $E_i \subset \mathbb{R}^d \times \mathbb{R}_+$ as

$$E_i = \{(y,s) \mid |y| \le R_{t_i} + (LR_{t_i} + C + 1)h,\ t_i \le s \le t_{i+1}\}.$$

Notice that $(\widehat{x}_\tau, t_i + \tau) \in E_i$ for any $0 \le \tau \le h$. We define

$$L^A_{\theta,i} = \mathrm{Lip}_{E_i}(\partial_t \nabla \psi_\theta) \triangleq \sup_{(y,s),(y',s')\in E_i} \frac{|\partial_t \nabla \psi_\theta(y,s) - \partial_t \nabla \psi_\theta(y',s')|}{|y-y'| + |s-s'|}, \tag{3.14}$$

i.e., $L^A_{\theta,i}$ as the Lipschitz constant of vector function $\partial_t \nabla \psi_\theta(x,t)$ on E_i. Then we have

$$|\partial_t \nabla \psi_\theta(\widehat{x}_\tau, t_i+\tau) - \partial_t \nabla \psi_\theta(\tilde{x}_{t_i}, t_i)| \le L^A_{\theta,i}(|\widehat{x}_\tau - \tilde{x}_{t_i}| + \tau) \le L^A_{\theta,i}\, 3(LR_{t_i} + C + 1)h. \tag{3.15}$$

Let us denote

$$M_{\theta,i} = \sup_{x \in \mathrm{supp}(\tilde{\rho}_{t_i})} \|\nabla^2 \psi_\theta(x,t_i)\|, \tag{3.16}$$

and

$$L^B_{\theta,i} = \mathrm{Lip}_{E_i}(\nabla^2 \psi_\theta) \triangleq \sup_{(y,s),(y',s')\in E_i} \frac{\|\nabla^2 \psi_\theta(y,s) - \nabla^2 \psi_\theta(y',s')\|}{|y-y'| + |s-s'|}, \tag{3.17}$$

here $\|\cdot\|$ is the 2-norm of the square matrix.

Direct calculation yields that

$$\begin{aligned}
&\left|\nabla^2 \psi_\theta(\widehat{x}_\tau, t_i+\tau) \frac{\partial}{\partial p} H(\widehat{x}_\tau, \widehat{p}_\tau) - \nabla^2 \psi_\theta(\tilde{x}_{t_i}, t_i) \frac{\partial}{\partial p} H(\tilde{x}_{t_i}, \tilde{p}_{t_i})\right| \\
=& \left|(\nabla^2 \psi_\theta(\widehat{x}_\tau, t_i+\tau) - \nabla^2 \psi_\theta(\tilde{x}_{t_i}, t_i)) \frac{\partial}{\partial p} H(\widehat{x}_\tau, \widehat{p}_\tau) + \nabla^2 \psi_\theta(\tilde{x}_{t_i}, t_i)(\frac{\partial}{\partial p} H(\widehat{x}_\tau, \widehat{p}_\tau) - \frac{\partial}{\partial p} H(\tilde{x}_{t_i}, \tilde{p}_{t_i}))\right| \\
\le& L^B_{\theta,i}(|\widehat{x}_\tau - \tilde{x}_{t_i}| + \tau)\left|\frac{\partial}{\partial p} H(\widehat{x}_\tau, \widehat{p}_\tau)\right| + \|\nabla^2 \psi_\theta(\tilde{x}_{t_i}, t_i)\| L_1(|\widehat{x}_\tau - \tilde{x}_{t_i}| + |\widehat{p}_\tau - \tilde{p}_{t_i}|) \\
\le& L^B_{\theta,i}(2(LR_{t_i} + C + 1)\tau + \tau)(|\partial_p H(0,0)| + L_1(R_{t_i} + 2(LR_{t_i} + C + 1)\tau)) \\
&+ M_{\theta,i} 2L_1(LR_{t_i} + C + 1)\tau,
\end{aligned} \tag{3.18}$$

and that

$$\left|\frac{\partial}{\partial x} H(\widehat{x}_\tau, \widehat{p}_\tau) - \frac{\partial}{\partial x} H(\tilde{x}_{t_i}, \tilde{p}_{t_i})\right| \le L_2(|\widehat{x}_\tau - \tilde{x}_{t_i}| + |\widehat{p}_\tau - \tilde{p}_{t_i}|) \le 2L_2(LR_{t_i} + C + 1)\tau \tag{3.19}$$

For convenience, we introduce

$$\mathscr{D}\psi_\theta(x,p,t) = \frac{\partial}{\partial t} \nabla \psi_\theta(x,t) + \nabla^2 \psi_\theta(x,t) \frac{\partial}{\partial p} H(x,p) + \frac{\partial}{\partial x} H(x,p).$$

Combining (3.15),(3.18) and (3.19), and denoting

$$\lambda(\theta,i) = 3L^A_{\theta,i}(LR_{t_i}+C+1) + L^B_{\theta,i}3(LR_{t_i}+C+1)(|\partial_p H(0,0)| + L_1(R_{t_i}+2(LR_{t_i}+C+1)h)) + 2L_1M_{\theta,i}(LR_{t_i}+C+1) + 2L_2(LR_{t_i}+C+1), \tag{3.20}$$

we can bound

$$|\mathscr{D}\psi_\theta(\widehat{x}_\tau,\widehat{p}_\tau,t_i+\tau) - \mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i)| \le \lambda(\theta,i)\tau. \tag{3.21}$$

We reformulate (3.10) as

$$h\mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i) = \int_0^h \mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i) - \mathscr{D}\psi_\theta(\widehat{x}_\tau,\widehat{p}_\tau,t_i+\tau)\, d\tau + (\nabla\psi_\theta(\widehat{x}_h,t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_{i+1}},t_{i+1})) + (\tilde{p}_{t_{i+1}} - \widehat{p}_h) + (e_{i+1}-e_i).$$

We have the following estimate

$$|\mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i)| \le \frac{1}{h}\int_0^h |\mathscr{D}\psi_\theta(\widehat{x}_\tau,\widehat{p}_\tau,t_i+\tau) - \mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i)|\, d\tau + \frac{1}{h}|\nabla\psi_\theta(\widehat{x}_h,t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_{i+1}},t_{i+1})| + \frac{|\tilde{p}_{t_{i+1}} - \widehat{p}_h|}{h} + \frac{|e_{i+1}-e_i|}{h}. \tag{3.22}$$

Using (3.21), the first term on the right hand side of (3.22) is upper bounded by $\frac{1}{2}\lambda(\theta,i)h$.

Let us define

$$D_i = \{x \mid |x| \le R_{t_i} + 3(LR_{t_i}+C+1)h\ \}.$$

and

$$L^C_{\theta,i} = \mathrm{Lip}(\nabla\psi_\theta(\cdot,t_i)) \triangleq \sup_{y,y'\in D_i} \frac{|\nabla\psi_\theta(y,t_i) - \nabla\psi_\theta(y',t_i)|}{|y-y'|}. \tag{3.23}$$

Recall the notation used in (3.4). Since we assume that the numerical scheme for integrating the Hamiltonian system has local truncation error of order r, the second term can be bounded by

$$\frac{1}{h}|\nabla\psi_\theta(\widehat{x}_h,t_{i+1}) - \nabla\psi_\theta(\tilde{x}_{t_{i+1}},t_{i+1})| \le L^C_{\theta,i+1}\frac{|\widehat{x}_h - \tilde{x}_{t_{i+1}}|}{h} \le L^C_{\theta,i+1}C_{\tilde{\Phi}_h}(\tilde{x}_{t_i},\tilde{p}_{t_i})h^{r-1}. \tag{3.24}$$

Similarly, the last two terms in (3.22) can be bounded by $C_{\tilde{\Phi}_h}(\tilde{x}_{t_i},\tilde{p}_{t_i})h^{r-1}$.

The left hand side of (3.22) can be recast as

$$|\mathscr{D}\psi_\theta(\tilde{x}_{t_i},\nabla\psi_\theta(\tilde{x}_{t_i}),t_i) + (\mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i) - \mathscr{D}\psi_\theta(\tilde{x}_{t_i},\nabla\psi_\theta(\tilde{x}_{t_i}),t_i))|.$$

Since $\nabla\psi_\theta(\tilde{x}_{t_i},t_i) = \tilde{p}_{t_i} + e_i$, we have

$$\begin{aligned} &|\mathscr{D}\psi_\theta(\tilde{x}_{t_i},\tilde{p}_{t_i},t_i) - \mathscr{D}\psi_\theta(\tilde{x}_{t_i},\nabla\psi_\theta(\tilde{x}_{t_i}),t_i)| \\ &\le \|\nabla^2\psi_\theta(\tilde{x}_{t_i},t_i)\|L_1|\tilde{p}_{t_i} - \nabla\psi_\theta(\tilde{x}_{t_i})| + L_2|\tilde{p}_{t_i} - \nabla\psi_\theta(\tilde{x}_{t_i})| \\ &\le (M_{\theta,i}L_1 + L_2)e_i. \end{aligned}$$

Let us recall

$$\mathscr{D}\psi_\theta(\tilde{x}_{t_i}, \nabla\psi_\theta(\tilde{x}_{t_i}), t_i) = \nabla\left(\frac{\partial}{\partial t}\psi_\theta(\tilde{x}_{t_i}, t_i) + H(\tilde{x}_{t_i}, \nabla\psi_\theta(\tilde{x}_{t_i}))\right),$$

thus, (3.22) leads to

$$\begin{aligned}&\left|\nabla\left(\frac{\partial}{\partial t}\psi_\theta(\tilde{x}_{t_i}, t_i) + H(\tilde{x}_{t_i}, \nabla\psi_\theta(\tilde{x}_{t_i}))\right)\right| \\ &\le \frac{1}{2}\lambda(\theta,i)h + (L^C_{\theta,i+1}+1)C_{\tilde{\Phi}_h}(\tilde{x}_{t_i}, \tilde{p}_{t_i})h^{r-1} + \frac{|e_{i+1}-e_i|}{h} + (M_{\theta,i}L_1 + L_2)e_i.\end{aligned}$$

We finally take average over the sample points $\{\tilde{x}^{(k)}_{t_i}\}_{1\le k\le N}$. This leads to

$$\begin{aligned}&\frac{1}{N}\sum_{k=1}^N\left|\nabla\left(\frac{\partial}{\partial t}\psi_\theta(\tilde{x}^{(k)}_{t_i}, t_i) + H(\tilde{x}^{(k)}_{t_i}, \nabla\psi_\theta(\tilde{x}^{(k)}_{t_i}))\right)\right| \\ &\le\frac{1}{2}\lambda(\theta,i)h + \underbrace{(L^C_{\theta,i+1}+1)\frac{1}{N}\sum_{k=1}^N C_{\tilde{\Phi}_h}(\tilde{x}^{(k)}_{t_i}, \tilde{p}^{(k)}_{t_i})}_{\text{denote as } \eta(\theta,i)}h^{r-1} + \frac{1}{N}\sum_{k=1}^N\frac{|e^{(k)}_{i+1}-e^{(k)}_i|}{h} + \underbrace{(M_{\theta,i}L_1+L_2)}_{\text{denote as } \nu(\theta,i)}|e^{(k)}_i|.\end{aligned} \tag{3.25}$$

This provides an upper bound on the empirical average of the L^1-residual of ψ_θ using the computed samples $\{\tilde{x}^{(k)}_{t_i}\}_{1\le k\le N}$ at time node t_i.

To further estimate the expectation of the L^1-residual at all the time nodes $\{t_1, \dots, t_T\}$, let us denote $\tilde{\rho}_{t_i} = (\tilde{\Phi}_h \circ \cdots \circ \tilde{\Phi}_h)_\sharp\rho_0$ as the probability density function of the numerical solution $\tilde{x}_{t_i}$ computed by the chosen scheme starting from $x_0 \sim \rho_0$. For simplicity, let us denote the residual term of the Hamilton-Jacobi equation as

$$\mathcal{R}[\psi_\theta](x,t) = \nabla\left(\frac{\partial}{\partial t}\psi_\theta(x,t) + H(x, \nabla\psi_\theta(x,t))\right).$$

For a fixed time t_i and samples $\{\tilde{x}^{(k)}_{t_i}\}_{1\le k\le N} \sim \tilde{\rho}_{t_i}$, by Hoeffding's inequality (see e.g. [37]), for any $0<\delta<1$, with probability $1-\delta$, we can bound the gap between the expectation and the empirical average of the L^1 residual as

$$\left|\int_{\mathbb{R}^d}|\mathcal{R}[\psi_\theta](x,t_i)|\tilde{\rho}_{t_i}\,dx - \frac{1}{N}\sum_{k=1}^N|\mathcal{R}[\psi_\theta](\tilde{x}^{(k)}_{t_i}, t_i)|\right| \le \underbrace{\sup_{x\in\operatorname{supp}(\tilde{\rho}_{t_i})}|\mathcal{R}[\psi_\theta](x,t_i)|}_{\text{denote as } R(\theta,i)}\sqrt{\frac{\ln\frac{2}{\delta}}{2N}}. \tag{3.26}$$

Since we assume that $\operatorname{supp}(\rho_0)$ is a bounded set, and the solution map $\tilde{\Phi}_h$ of the numerical scheme is continuous, then $\operatorname{supp}(\tilde{\rho}_{t_i})$ is also bounded. Thus $R(\theta,i)$ is guaranteed to be finite.

By combining (3.25) and (3.26), for any time node t_i, with probability $1-\delta$, we can estimate the average L^1 residual of Hamilton-Jacobi equation at time t_i as

$$\begin{aligned}&\int_{\mathbb{R}^d}\left|\nabla\left(\frac{\partial}{\partial t}\psi_\theta(x,t_i) + H(x, \nabla\psi_\theta(x,t_i))\right)\right|\tilde{\rho}_{t_i}\,dx \\ &\le \frac{1}{2}\lambda(\theta,i)h + \eta(\theta,i)h^{r-1} + \left(\frac{1}{N}\sum_{k=1}^N\frac{|e^{(k)}_{i+1}-e^{(k)}_i|}{h} + \nu(\theta,i)|e^{(k)}_i|\right) + R(\theta,i)\sqrt{\frac{\ln\frac{2}{\delta}}{2N}}.\end{aligned} \tag{3.27}$$

If we denote the subset Ω_{t_i} of the sample space on which (3.27) holds. It follows that $\mathbb{P}(\Omega_{t_i}^c) \leq \delta$. Then we have

$$\mathbb{P}\left(\bigcap_{i=1}^{M} \Omega_{t_i}\right) = 1 - \mathbb{P}\left(\bigcup_{i=1}^{M} \Omega_{t_i}^c\right) \geq 1 - \sum_{i=1}^{M} \mathbb{P}\left(\Omega_{t_i}^c\right) \geq 1 - M\delta.$$

By letting $M\delta = \epsilon$, we have shown that for the fixed neural network ψ_θ, initial distribution with density ρ_0 and initial samples $\{x_{t_0}^{(k)}\}_{k=1}^N \sim \rho_0$, with probability $1-\epsilon$,

$$\int_{\mathbb{R}^d} \left| \nabla \left(\frac{\partial}{\partial t}\psi_\theta(x, t_i) + H(x, \nabla\psi_\theta(x, t_i)) \right) \right| \tilde{\rho}_{t_i} \, dx$$

$$\leq \frac{1}{2}\lambda(\theta, i)h + \eta(\theta, i)h^{r-1} + \delta_i^{N,h} + \nu(\theta, i)\varepsilon_i^N + R(\theta, i)\sqrt{\frac{\ln M + \ln \frac{2}{\epsilon}}{2N}} \tag{3.28}$$

holds at any time node t_i, $i = 1, 2, \ldots, M$. ∎

We want to highlight that the *posterior* estimation on the L^1-residual of $\nabla\psi_\theta$ consists of three parts: the numerical error depending on the geometric integrator $\frac{1}{2}\lambda(\theta, i)h + \eta(\theta, i)h^{r-1}$ in (3.8), the training error $\delta_i^{N,h} + \nu(\theta, i)\varepsilon_i^N$ caused by the neural network approximation, and the sampling error $R(\theta, i)((\ln M + \ln \frac{2}{\epsilon})/(2N))^{1/2}$ due to the Monte–Carlo method. For the results about explicit bound of ε_i^N, one may use the McDiarmid's inequality [37] and Rademacher complexity $\mathrm{Rad}(F)$ of the function set $F = \{\mathcal{R}[\psi_\theta] \circ \tilde{\Phi}_h^i\}_{i=0,1,\ldots,M}$, as well as Masaart Lemma [37] on estimating the upper bound of $\mathrm{Rad}(F)$. Since ε_i^N mainly relies on the approximation power of ψ_θ, which is another topic beyond the scope of this work, we omit its detailed discussion here.

We note that the error estimate (3.8) is established for *density-weighted* residual of $\nabla\psi_\theta$. Here the probability density $\tilde{\rho}_{t_i}$ of numerical solution $\tilde{x}_{t_i}$ is solved via the geometric integrator $\tilde{\Phi}_h$. We anticipate smaller residual values of $\nabla\psi_\theta$ at the region on which $\tilde{\rho}_{t_i}$ possesses a higher probability. On the contrary, no estimate is provided outside of the support of $\tilde{\rho}_{t_i}$. Such an observation is verified in the later section 4.1.

We would like to remark that, if assuming the existence of the classical solution, one can show that the temporal convergence order of numerical integrator in proposed algorithm can be improved to $r-2$ ($r > 2$) via similar arguments as in the proof of Theorem 3.1. Besides, the error analysis in Theorem 3.1 works for any $T > 0$ even when T goes beyond the threshold time T_* of classical solution. However, when t_i is approaching (or even surpassing) T_*, the superposition of momentum vectors in the configuration space often leads to a larger training loss $\mathcal{E}_i$, which increases the error upper bound in (3.8). Such increment in the loss values $\mathcal{E}_i$ is reflected in several numerical examples demonstrated in section 4.2. This is justifiable because the classical solution itself even cannot be extended beyond T_*, and we are not able to control the residual value of $\nabla\psi_\theta$ when time t_i approaches (or surpasses) T_*. On the other hand, in our proposed algorithm, the numerical solution ψ_θ extends naturally beyond T_*, which can be treated as the approximation to the $\mu_t(\cdot|x)-$weighted "solution" $\widehat{\psi}$ to the HJ equation (1.1) discussed in remark 2.1. Several numerical examples of such $\mu_t(\cdot|x)-$weighted "solution" are also demonstrated in section 4.2.

4. Numerical tests. In our implementation, we set $\psi_\theta(\cdot,\cdot) : \mathbb{R}^{d+1} \to \mathbb{R}$ as neural network with ResNet [22] structure in our implementation. To be more precise, we consider the following neural network $\mathcal{NN}_\theta^{L,\widetilde{d}}(\cdot,\cdot) : \mathbb{R}^{d+1} \to \mathbb{R}$ with depth L and width (hidden dimension) $\widetilde{d}$ as

$$\mathcal{NN}_\theta^{L,\widetilde{d}}(x,t) = f_L \circ f_{L-1} \circ \dots f_2 \circ f_1(x,t),$$

with each $f_k(y) = \sigma(y + \kappa(A_k y + b_k))$. We choose the activation function $\sigma(\cdot)$ as the hyperbolic tangent function $\tanh(\cdot)$. And $\kappa \in \mathbb{R}^+$ is the stepsize of each layer, we choose $\kappa = 0.5$ in our experiments. Furthermore, $A_1 \in \mathcal{M}_{\widetilde{d}\times(d+1)}(\mathbb{R}), b_1 \in \mathbb{R}^{\widetilde{d}}$, $A_k \in \mathcal{M}_{\widetilde{d}\times\widetilde{d}}(\mathbb{R}), b_k \in \mathbb{R}^{\widetilde{d}}$ for all $2 \leq k \leq L-1$, and $A_L \in \mathcal{M}_{1\times\widetilde{d}}(\mathbb{R}), b_L \in \mathbb{R}^1$ compose the parameter $\theta \in \mathbb{R}^{(L-2)\widetilde{d}^2+\widetilde{d}(d+2)+(L-1)\widetilde{d}+1}$ of this neural network.

We apply the Adam method [26] to train ψ_θ in Algorithm 3.1. We pick the random batch size $N_0 = 1200$ and the threshold $err_0 = 10^{-4}$ for all the numerical experiments discussed in this section. All the numerical examples are tested on Google Colab with GPU acceleration. The training time for ψ_θ on each time interval is around 3-10 minutes for problems with dimensions varying from 2 to 30.

4.1. Residual and error bounds. Theorem 3.1 states that the expectation of the residual can be bounded, where the expectation is taken with respect to the distribution $\tilde{\rho}_{t_i}$ of samples used for training ψ_θ. Thus we anticipate a smaller residual value on the support of $\tilde{\rho}_{t_i}$; On the other hand, the residual outside of the support of $\tilde{\rho}_{t_i}$ can not be controlled due to lack of learning samples. This is observed in the following examples.

Consider the Hamilton-Jacobi equation on $\mathbb{R}^2 \times [0,T]$ with $T = 3$, $H(x,p) = \frac{|p|^2}{2} + \frac{|x|^2}{2}$ and initial data $u(x) = \frac{|x|^2}{2}$. We choose $\rho_0 = \mathcal{N}((3,3), I)$, i.e., the normal distribution shifted by $(3,3)$. We set $\psi_\theta = \mathcal{NN}_\theta^{L,\widetilde{d}}$ with $L = 7, \widetilde{d} = 40$. We choose the number of time subintervals $M = 40$, and the number of samples $N = 7500$. We set the learning rate $lr = 0.5 \cdot 10^{-4}$ and perform Adam's method for $N_{\text{Iter}} = 8000$ iterations. We plot the heat map of the residual term

$$\text{Res}(x,t) = \left| \nabla \left(\frac{\partial}{\partial t} \psi_\theta(x,t) + H(x, \nabla\psi_\theta(x,t)) \right) \right| \tag{4.1}$$

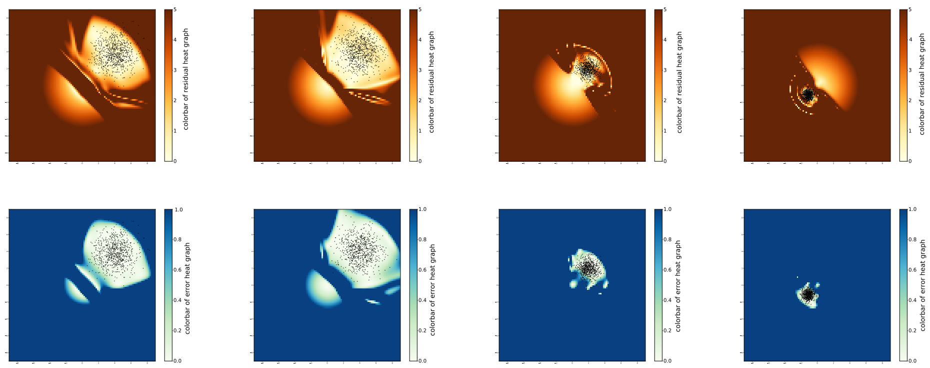
together with the samples $\{x_{t_i}^{(k)}\}_{k=1}^N$ at different time nodes t_i in the first row of Figure 1. The support of the samples mostly overlaps with the region on which the residual value $\text{Res}(x,t)$ is small. A similar observation is also found about the error between $\nabla\psi_\theta(x,t)$ and the real solution $\nabla u(x,t)$, where $u(x,t) = \frac{1}{2}\cot(t+\frac{\pi}{4})|x|^2$, i.e.

$$\text{Err}(x,t) = |\nabla\psi_\theta(x,t) - \nabla u(x,t)|. \tag{4.2}$$

The results are demonstrated in the second row of Figure 1.

Another interesting question is how the sample size N affects the accuracy of the numerical solution $\nabla\psi_\theta$. To test it, we train ψ_θ by using different sample size N while keeping other hyperparameters unchanged. We examine the relationship between the $L^2(\rho_t)$ error $\|\nabla\psi_\theta(\cdot,t) - \nabla u(\cdot,t)\|^2_{L^2(\rho_t)}$ and the sample size N on time interval $[0, 0.25]$, where we discretize the time interval into $M = 100$ subintervals.

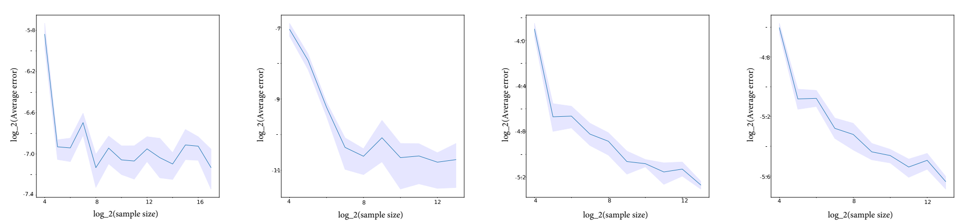
We repeat Algorithm 3.1 for different sample sizes $N = 16 \cdot 2^k$ with $k = 0, 1, \dots 9$. We approximate the $L^2(\rho_t)$ discrepancy between numerical solution $\nabla\psi_\theta$ and real solution ∇u by using the Monte–Carlo method with a large sample size 45000. We conduct the numerical experiments on the same Hamilton-Jacobi equation with dimensions being 2 and 10 respectively. The results are plotted in Figure 2, showing



(a) $t = 3/8$ (b) $t = 9/8$ (c) $t = 15/8$ (d) $t = 21/8$

Figure 1: (Up row) Heat graphs of the residual $\mathrm{Res}(x, t)$ of the numerical solution ψ_θ and the sample points (black) at different time stages t. (Down row) Heat graphs of the error $\mathrm{Err}(x, t)$ of the numerical solution ψ_θ and the sample points (black) at different time stages t.

that the accuracy of the proposed method improves as the number of sample sizes N increases.



(a) $d = 2$, $t = 0.1$ (b) $d = 2$, $t = 0.2$ (c) $d = 10$, $t = 0.1$ (d) $d = 10$, $t = 0.2$

Figure 2: Average error versus sample size plots ($\log_2 - \log_2$) for 2D and 10D HJ equation (plots with confidence interval $(25\% - 75\%)$ based on 40 sets of data)

4.2. Solving HJ equations . In this part, we first test our algorithm on the separable Hamiltonian $H(x, p) = K(p) + V(x)$ with the quadratic kinetic energy $K(p) = \frac{1}{2}|p|^2$. For these examples, we apply our method to solve equation (1.1) with the one-step Störmer–Verlet scheme [20] for the corresponding Hamiltonian system (3.1) We then compute an HJ equation with non-separable Hamiltonian $H(x, p)$ in example 4.2.4, in which the explicit symplectic scheme proposed in [40] is used to compute the Hamiltonian system (3.1) in our algorithm. Finally, in example 4.2.5, we apply our algorithm to the linear quadratic control (LCQ) problem of inverted pendulums with terminal density constraint.

We summarize the hyperparameters used in our algorithm for each numerical example in the following table. Recall that L is the depth and $\widetilde{d}$ is the width of the neural network ψ_θ; M denotes the total number of time steps; M_T denotes the number of subintervals used to divide the entire time interval $[0, T]$, which will be explained in details in example 4.2.1; N is the number of samples used in our computation; lr is the learning rate for the Adam method; and N_{Iter} denotes the total iteration number.

Example (dimension)	L	$\widetilde{d}$	M	M_T	N	lr	N_{Iter}
4.2.1 $(d = 30)$	6	50	200	25	8000	10^{-4}	30000
4.2.2 $(d = 20)$	6	50	30	1	12000	0.5×10^{-4}	6000
4.2.3 $(d = 30)$	6	80	100	1	5000	0.5×10^{-4}	6000
4.2.4 $(d = 20)$	7	40	100	4	5000	10^{-4}	12000
4.2.5 $(d = 4)$	6	40	100	1	5000	$0.5 \cdot 10^{-4}$	20000

Table 1: Hyperparameters of our algorithm for examples 4.2.1 - 4.2.4.

4.2.1. Example with Quadratic Potential. We set the potential and the initial condition as $V(x) = \frac{1}{2}|x|^2$ and $g(x) = \frac{1}{2}|x|^2$. We choose $\rho_0 = \mathcal{N}(\underbrace{(3, ..., 3)}_{30}, I)$ and solve this equation on $[0, 5]$.

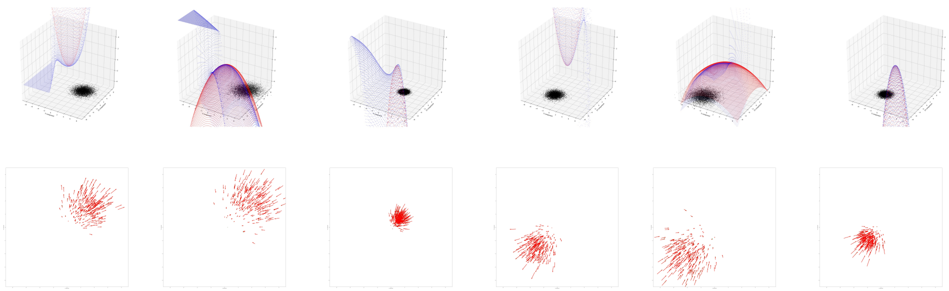
It can be verified directly that $u(x, t) = \frac{1}{2}\cot(t + \frac{\pi}{4})|x|^2$ is the classical solution to the equation on $[0, \frac{3\pi}{4})$. When t approaches $T_* = \frac{3\pi}{4}$, this classical solution blows up. Our method is able to compute both the classical solution as well as the extended solution beyond T^*.

The solution to this HJ equation possesses a rather strong oscillatory profile along time t. Due to the rigidity of the neural network, given $T = 5$, it is generally difficult for a single neural network to capture the overall shape of $\{u(x, t)\}_{t \geq 0}$ [29].

As a remedy, in order to make our computation more efficient, we apply the multi-interval training strategy in this example. We separate $[0, T]$ into multiple shorter subintervals and train different neural networks on each subinterval. Our experiments indicate that such treatment of training the networks independently on each subinterval and concatenating together improves the flexibility of the numerical solution $\psi_\theta(x, t)$ and thus enhances the performance. To be more specific, we divide $[0, T]$ into $M_T = 25$ equal intervals, i.e., $[0, T] = \bigcup_{k=1}^{M_T} I_k$ with each $I_k = [\frac{k-1}{M_T}T, \frac{k}{M_T}T)$ for $1 \leq k \leq M_T - 1$ and $I_{M_T} = [\frac{M_T - 1}{M_T}T, T]$. We train ψ_{θ_k} on each I_k and set $\psi_\theta(x, t) = \sum_{k=1}^{M_T} \chi_{I_k}(t)\psi_{\theta_k}(x, t)$ as our numerical solution. Here χ_{I_k} is the indicator function of time interval I_k.

We demonstrate the numerical solutions in Figure 3. Since the solution is a high dimensional function, we plot its graph on the 5-th and 15-th coordinates. For convenience, we call it 5th − 15th plane. It is observed that both the solution and vector field have good agreements with their exact counterparts at the regions where samples are drawn.

Recall the $\{\epsilon_i^N\}$ defined in (3.7), we calculate the total loss $\sum_{i=(j-1)l}^{jl-1} \varepsilon_i^N$ among the time nodes located in the subinterval I_j, where $l = \frac{M}{M_T}$, and plot $\sum_{i=(j-1)l}^{jl-1} \varepsilon_i^N$ $(1 \leq j \leq M_T)$ versus time in Figure 4. It is clear that the error increases significantly around $T_* = \frac{3\pi}{4} \approx 2.36$. According to our experience, it is intrinsically difficult to compute the solution near singular point T_*.



(a) $t = 0.0$ (b) $t = 1.0$ (c) $t = 2.0$ (d) $t = 3.0$ (e) $t = 4.0$ (f) $t = 5.0$

Figure 3: 1st row: Graphs of the numerical solution ψ_θ (blue) and the exact solution (red) at different time stages on the 5th − 15th plane; 2nd row: Plots of vector fields $\nabla\psi_\theta(\cdot, t)$ (green) with momentums of samples (red) at different time stages on the 5th − 15th plane.

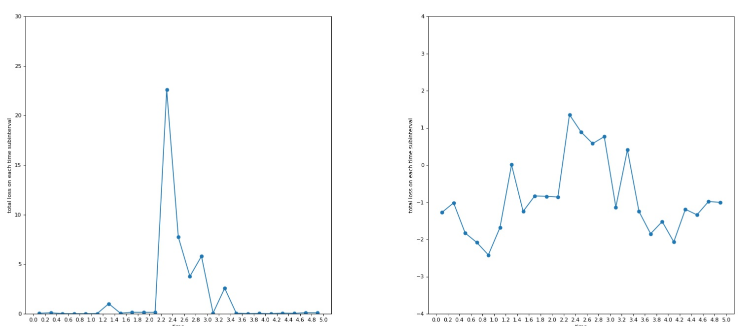


Figure 4: Plot of $\sum_{i=(j-1)l}^{jl-1} \varepsilon_i^N$ $(1 \le j \le M_T)$ versus time (Left) and its semi-$\log_{10}$ plot (Right).

4.2.2. Example with Sinusoidal Initial Condition. In this example, we consider the Hamiltonian with a degenerate quadratic kinetic energy and without potential energy. We set the kinetic energy $K(p) = \frac{1}{2}p^\top \boldsymbol{\Sigma} p + \tau \boldsymbol{\eta}^\top p$ with $\boldsymbol{\Sigma} = \frac{1}{d}\mathbf{1}\mathbf{1}^\top$, $\boldsymbol{\eta} = \frac{1}{\sqrt{d}}\mathbf{1}$, $\tau = 3$. Here we define $\mathbf{1} = (1, 1, ..., 1)^\top$ as a d-dimensional vector. We pick the initial condition $u(x, 0) = g(x)$ with $g(x) = \cos(\sqrt{3}\boldsymbol{\eta}^\top x)$. We choose ρ_0 as the uniform distribution on the square region $[-4.5, 4.5]^d$ and solve this equation on $[0, \frac{2}{3}]$.

It can be verified that the classical solution $u(x, t)$ of (1.1) takes the form $u(x, t) = f(\boldsymbol{\eta}^\top x, t)$, where $f(\cdot, t) : \mathbb{R} \to \mathbb{R}$ satisfies

$$f'(\xi + t(\tau - \sqrt{3}\sin(\sqrt{3}\xi)), t) = -\sqrt{3}\sin(\sqrt{3}\xi),$$

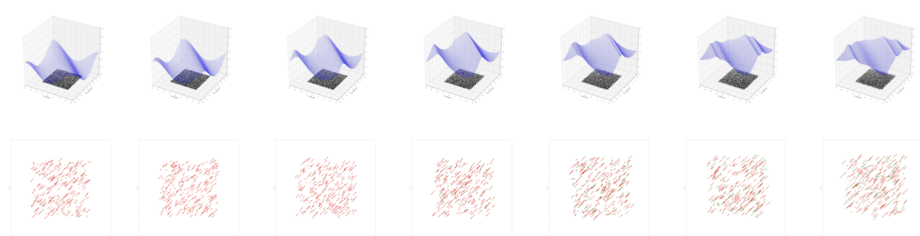
for any $\xi \in \mathbb{R}$. We denote $\varphi_t(\xi) = \xi + t(\tau - \sqrt{3}\sin(\sqrt{3}\xi))$. Since $\varphi_t'(\xi) = 1 - 3t\cos(\sqrt{3}\xi)$,

φ_t is injective when time $t < \frac{1}{3}$. Thus,

$$f'(x,t) = -\sqrt{3}\sin(\sqrt{3}\varphi_t^{-1}(x)),$$

for all $t \in [0, 1/3)$, on which we can also verify that the classical solution to Hamilton-Jacobi equation (1.1) exists.

We demonstrate the numerical solutions in Figure 5. In order to compare our



(a) $t = 0$ (b) $t = \frac{1}{9}$ (c) $t = \frac{2}{9}$ (d) $t = \frac{1}{3}$ (e) $t = \frac{4}{9}$ (f) $t = \frac{5}{9}$ (g) $t = \frac{2}{3}$

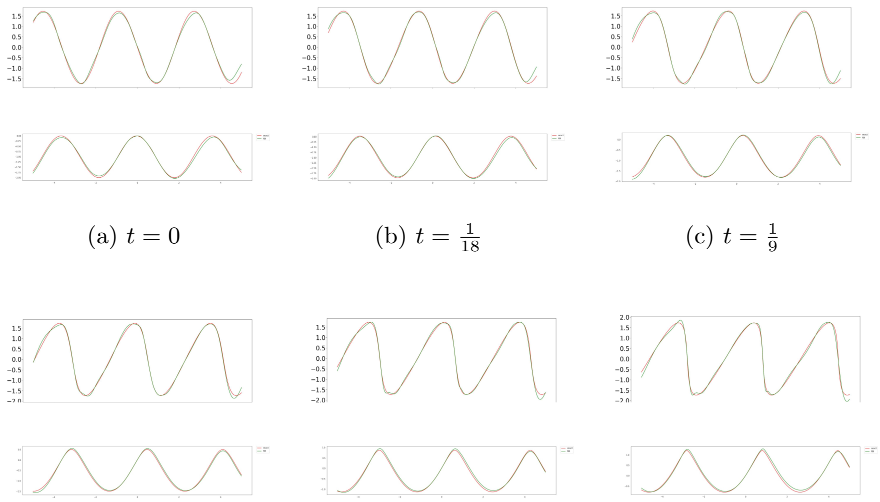
Figure 5: 1st row: Graphs of our numerical solution ψ_θ (blue) at different time stages on the 5th − 15th plane; 2nd row: Plots of vector fields $\nabla\psi_\theta(\cdot, t)$ (green) with momentums of samples (red) at different time stages on the 5th − 15th plane.

numerical solution with the exact solution clearly, we fix on the diagonal line passing through 0 in $\mathbb{R}^{20}$ and plot our numerical solution (green) against the exact solution (red) before time $T_* = \frac{1}{3}$ in Figure 6. They show good agreement.

We further plot the loss $\frac{1}{N}\sum_{k=1}^{N}|e_{t_i}^{(k)}|^2$ (recall $e_{t_i}^{(k)}$ defined in (3.6)) versus the time nodes t_i in Figure 10(left subfigure). One can observe that the loss remains small before $T_* = \frac{1}{3}$ and increases significantly afterward. This is due to the singularity developed at T_*.

4.2.3. Example with Sinusoidal Potential and Gaussian Mixture as the Initial Distribution . We now consider the Hamiltonian with a sinusoidal potential energy $H(x,p) = \frac{1}{2}|p|^2 + \cos(2x_{i_1} + 0.4) + \cos(2x_{i_2} + 0.4)$, the initial condition $u(x,0) = g(x) = \sin(x_{i_1} + 0.15) + \sin(x_{i_2} + 0.15)$, and the initial distribution $\rho_0 = \frac{1}{2}(\mathcal{N}(\mu_1, I) + \mathcal{N}(\mu_2, I))$, where $\mu_1 = -\frac{\pi}{2}(\boldsymbol{e}_{i_1} + \boldsymbol{e}_{i_2})$ and $\mu_2 = \frac{\pi}{2}(\boldsymbol{e}_{i_1} + \boldsymbol{e}_{i_2})$. Here $\boldsymbol{e}_i$ denotes the vector with i-th entry being 1 and remaining entries all 0; and i_1, i_2 are two different integers between 1 and d. In this example, we set $d = 30$, $i_1 = 10, i_2 = 20$. We solve the equation on $[0, 1]$. A similar equation in one dimension was first considered in [23] and [24] in which the multivalued physical observables for the semiclassical limit of the Schrödinger equation was computed.

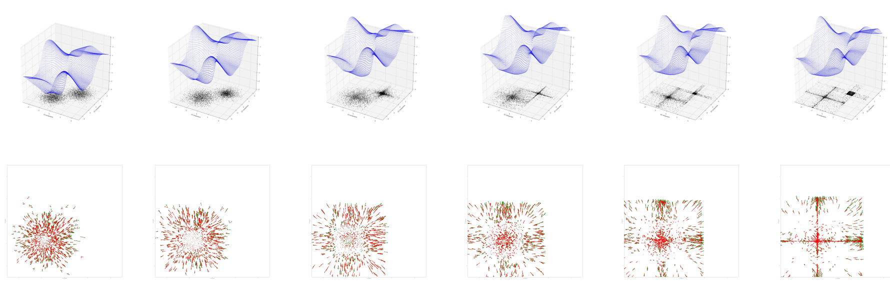
We demonstrate the numerical solutions in Figure 7. Similarly, we plot the loss $\frac{1}{N}\sum_{k=1}^{N}|e_{t_i}^{(k)}|^2$ versus time nodes t_i in Figure 10(middle subfigure), which shows a significant increase in loss after $t = 0.4$. We don't know the exact solution for this example. The numerical result suggests that the kinks of the solution may develop at $T_* \approx 0.4$.



(a) $t = 0$ (b) $t = \frac{1}{18}$ (c) $t = \frac{1}{9}$

(d) $t = \frac{1}{6}$ (e) $t = \frac{2}{9}$ (f) $t = \frac{5}{18}$

Figure 6: 1st and 3rd row: Comparison between directional derivative of numerical solution $\boldsymbol{\eta}^\top \nabla \psi_\theta(x, t)$ (green) and exact solution $\boldsymbol{\eta}^\top \nabla u(x, t)$ (red); 2nd and 4th row: Compare the function value of numerical solution $\psi_\theta(x, t)$ (green) with exact solution $\psi(x, t)$ (red). Both are restricted on the diagonal line in $\mathbb{R}^{20}$.



(a) $t = 0$ (b) $t = 0.2$ (c) $t = 0.4$ (d) $t = 0.6$ (e) $t = 0.8$ (f) $t = 1.0$

Figure 7: (Up row) Graphs of our numerical solution ψ_θ (blue) at different time stages on the 10th $-$ 20th plane; (Down row) Plots of vector fields $\nabla \psi_\theta(\cdot, t)$ (green) with momentums of samples (red) at different time stages on the 10th $-$ 20th plane.

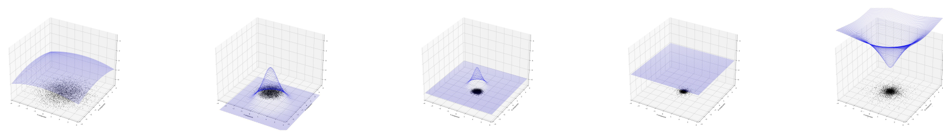
4.2.4. Example of non-separable Hamiltonian. In this example, we consider the following non-separable Hamiltonian

$$H(x, p) = \frac{1}{2}(|x|^2 + 1)(|p|^2 + 1). \tag{4.3}$$

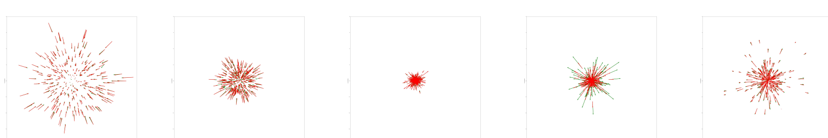
We take the initial value $u(x, 0) = g(x) = 0$ and solve this equation on $[0, 1]$. We set the initial distribution $\rho_a = \mathcal{N}(0, 2I)$ and the dimension $d = 10$. We adopt the explicit symplectic scheme (with $\omega = 10$) proposed in [40] to integrate the Hamiltonian system (3.1) associated with the Hamiltonian (4.3). The phase portraits are plotted in Figure 8.

Figure 8: Phase portraits of the Hamiltonian system associated with non-separable Hamiltonian (4.3). Here $0 \le t \le 1$. The dimension of x is 10, the dimension of the system is 20. We visualize the portraits by projecting the trajectories onto the first component of x and p. We use different colors to separate time intervals: green-$[0, 0.2)$; blue-$[0.2, 0.4)$; orange-$[0.4, 0.6)$; red-$[0.6, 8)$; pink-$[0.8, 1.0)$.

We demonstrate the graphs of the numerical solution $\psi_\theta(\cdot, t)$ at different time stages in Figure 9. The comparison between the learned vector field $\nabla\psi_\theta(\cdot, t)$ and the exact momentums is also provided in Figure 9. The gradient field and the momentum match well before $t = 0.4$ and after $t = 0.9$. This is also verified in the $\frac{1}{N}\sum_{k=1}^{N} |e_{t_i}^{(k)}|^2$-versus-$t_i$ plot presented in Figure 10 (right subfigure).

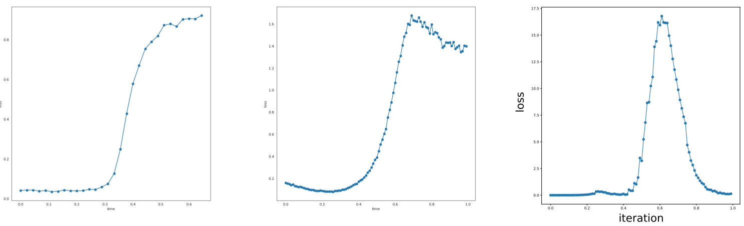


(a) $t = 0.1$ (b) $t = 0.3$ (c) $t = 0.5$ (d) $t = 0.7$ (e) $t = 0.9$



(f) $t = 0.1$ (g) $t = 0.3$ (h) $t = 0.5$ (i) $t = 0.7$ (j) $t = 0.9$

Figure 9: (Up row) Graphs of the numerical solution ψ_θ at different time stages on the 4th − 8th plane. (Down row) Plots of $\nabla\psi_\theta(\cdot, t)$ (green) with the momentum of samples (red) at different time stages.



(a) Example 4.2.2 (b) Example 4.2.3 (c) Example 4.2.4

Figure 10: Plots of the loss $\frac{1}{N}\sum_{k=1}^{N}|e_{t_i}^{(k)}|^2$ versus time t_i for examples 4.2.2, 4.2.3, 4.2.4.

4.2.5. Application to Linear Quadratic Control (LQC) problem with given terminal distribution . The Linear Quadratic Control (LQC) problem in $\mathbb{R}^d$ [27][39] is usually posed as

$$\min_{\{x_\tau\}_0^T, \{v_\tau\}_0^T} \int_0^T \frac{1}{2} v_\tau^\top R v_\tau + \frac{1}{2} x_\tau^\top Q x_\tau \, d\tau + \frac{1}{2} x_T^\top P_1 x_T, \tag{4.4}$$
$$\text{subject to } \dot{x}_\tau = A x_\tau + B v_\tau, x_\tau|_{\tau=0} = x_0.$$

Here we assume that R, Q, P_1 are symmetric matrices, R is positive definite, Q, P_1 are semi-positive definite. The critical point of this LQC problem solves the following ODE system based on the Pontryagin's minimum principle,

$$\begin{aligned} \dot{x}_\tau &= A x_\tau + B v_\tau, \quad v_\tau = R^{-1} B^\top \lambda_\tau, \quad x_\tau|_{\tau=0} = x_0, \\ \dot{\lambda}_\tau &= -A^\top \lambda_\tau + Q x_\tau, \quad \lambda_T = -P_1 x_T. \end{aligned} \tag{4.5}$$

Furthermore, we consider the value function

$$u(x,t) = \min_{\{x_\tau\}_t^T, \, \{v_\tau\}_t^T} \int_t^T \frac{1}{2} v_\tau^\top R v_\tau + \frac{1}{2} x_\tau^\top Q x_\tau \, d\tau + \frac{1}{2} x_T^\top P_1 x_T$$
$$\text{subject to } \dot{x}_\tau = A x_\tau + B v_\tau, \; x_\tau|_{\tau=t} = x,$$

then one verifies that $u(\cdot,t)$ solves the following Hamilton-Jacobi equation with terminal condition

$$\frac{\partial u(x,t)}{\partial t} + \underbrace{\min_v \left\{ \nabla u(x,t)^\top B v + \frac{1}{2} v^\top R v + \nabla u(x,t)^\top A x + \frac{1}{2} x^\top Q x \right\}}_{J(x, \nabla u(x))} = 0, \tag{4.6}$$
$$u(x,T) = \frac{1}{2} x^\top P_1 x.$$

The term $J(x, \nabla u(x,t))$ takes an explicit form

$$J(x, \nabla u(x,t)) = -\frac{1}{2} (B^\top \nabla u(x,t))^\top R^{-1} (B^\top \nabla u(x,t)) + \nabla u(x,t)^\top A x + \frac{1}{2} x^\top Q x.$$

The optimal control v_τ is given by

$$v_\tau = -R^{-1}B^\top \nabla u(x_\tau, \tau). \tag{4.7}$$

Now let we consider the LQC problem of a swarm of agents in which each of them minimizes its own control cost by resolving (4.4), while we want the terminal distribution formed by this swarm equals the given probability distribution ρ_T.

Our method readily handles this control problem with terminal density constraints. To be more specific, we consider the "time-reversal" of the Hamilton-Jacobi equation (4.6), i.e., we denote $\widetilde{u}(x,t) = u(x, T-t)$. This yields $\partial_t \widetilde{u} = -\partial_t u$. Thus $\widetilde{u}$ solves the HJ equation with initial condition

$$\frac{\partial \widetilde{u}(x,t)}{\partial t} + \underbrace{\frac{1}{2}(B^\top \nabla \widetilde{u}(x,t))^\top R^{-1}(B^\top \nabla \widetilde{u}(x,t)) - \nabla \widetilde{u}(x,t)^\top Ax - \frac{1}{2}x^\top Qx}_{H(x,\nabla u(x)) = -J(x,\nabla u(x))} = 0, \tag{4.8}$$

$$\widetilde{u}(x,0) = \frac{1}{2}x^\top P_1 x.$$

Here we denote the Hamiltonian $H(x,p)$ as

$$H(x,p) = \frac{1}{2}(B^\top p)^\top R^{-1}(B^\top p) - p^\top Ax - \frac{1}{2}x^\top Qx.$$

We then apply our method to (4.8) coupled with the initial probability distribution $\widetilde{\rho}_0 = \rho_T$.

Notice that the associated Hamiltonian system is

$$\dot{q}_t = \partial_p H(q_t, p_t), \quad \dot{p}_t = -\partial_x H(q_t, p_t). \quad \text{with } q_0 \sim \widetilde{\rho}_0,\ p_0 = P_1 q_0.$$

This yields the linear ODE system

$$\begin{bmatrix} \dot{q_t} \\ p_t \end{bmatrix} = \begin{bmatrix} -A & BR^{-1}B^\top \\ Q & A^\top \end{bmatrix} \begin{bmatrix} q_t \\ p_t \end{bmatrix}, \quad \begin{matrix} q_0 \sim \widetilde{\rho}_0, \\ p_0 = P_1 q_0. \end{matrix} \tag{4.9}$$

We denote $\widetilde{\rho}_T$ as the density of $\mathrm{Law}(q_T)$.

It is worth mentioning that this Hamiltonian system is equivalent to the ODE (4.5) obtained from the Pontryagin's minimum Principle up to the transformation $q_t = x_{T-t}, p_t = -\lambda_{T-t}$.

Now, recall (4.7) and $\widetilde{u}$ as the time-reversal of u, the optimal control is given by $v_\tau = -R^{-1}B^\top \nabla \widetilde{u}(x_\tau, T-\tau)$ for $0 \le \tau \le T$. In computation, we evaluate for the neural network-surrogate solution $\nabla \psi_\theta \approx \nabla \widetilde{u}$ of the HJ equation (4.8). To verify the accuracy of $\nabla \psi_\theta$, we compare the trajectory $\{\widehat{x}_\tau\}$ under our learned control

$$\dot{\widehat{x}}_\tau = -A\widehat{x}_\tau + BR^{-1}B^\top \nabla \psi_\theta(\widehat{x}_\tau, \tau), \quad \widehat{x}_0 \sim \rho_0 = \widetilde{\rho}_T,$$

with the dynamic computed from the Pontryagin's minimum principle (4.5).

Inverted Pendulum Specifically, we apply our method described above to the inverted pendulum model [19][38]. In this example, we denote the position of the cart as x_t, and the angle between the stick and the vertical direction as θ_t at time t (we take the counter-clockwise as the positive direction for θ_t). Suppose we exert a force

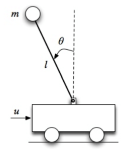


Figure 11: Illustration of inverted pendulum [3].

u_t on the cart at time t, the mechanics of the cart and the stick are governed by the following differential equation (The equation has been linearized at $\theta \approx 0, \dot{\theta} \approx 0$.)

$$\begin{aligned} u_t =& (M+m)\ddot{x}_t - ml\ddot{\theta}_t \\ l\ddot{\theta}_t =& g\theta_t + \ddot{x}_t. \end{aligned}$$

This yields

$$\begin{aligned} \ddot{x}_t =& \frac{m}{M} g\theta_t + \frac{u_t}{M} \\ \ddot{\theta}_t =& \frac{M+m}{Ml} g\theta_t + \frac{u_t}{Ml}. \end{aligned}$$

By introducing $y_t = \dot{x}_t, \phi_t = \dot{\theta}_t$, we consider the following dynamics of

$$\mathbf{X}_t = (x_t, y_t, \theta_t, \phi_t)$$

with the external force u_t as the control,

$$\overset{\cdot}{\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ \phi_t \end{bmatrix}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{M+m}{Ml} g & 0 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ \phi_t \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{Ml} \end{bmatrix} [u_t] \overset{\text{denote as}}{=} A\mathbf{X}_t + Bu_t.$$

We wish to exert the control $\{u_t\}$ to this dynamics so that both the cart and the stick stay stably, and at the same time, minimize the effort u_t paid to the control. Thus, we consider the following cost functional

$$\int_0^T \frac{1}{2}\mathbf{X}_t^\top Q \mathbf{X}_t + \frac{1}{2} R u_t^2 + \frac{1}{2}\mathbf{X}_T^\top P_1 \mathbf{X}_T.$$

Here we pick $Q = P_1 = \operatorname{diag}(1,0,1,0)$, $R = 1$. This is a optimal control problem in 4-dimensional phase space of x, θ. We assume the terminal distribution ρ_T as $\mathcal{N}(0, \sigma_x^2 I_2) \otimes \mathcal{U}([-\theta_0, \theta_0]) \otimes \mathcal{N}(0, \sigma_{\dot{\theta}}^2)$. That is, if $(x, \dot{x}, \theta, \dot{\theta}) \sim \rho_T$, then $(x, \dot{x}) \sim \mathcal{N}(0, \sigma_x^2 I_2)$, $\theta \sim \mathcal{U}([-\theta_0, \theta_0])$, $\dot{\theta} \sim \mathcal{N}(0, \sigma_{\dot{\theta}}^2)$. Here $\mathcal{U}([a,b])$ denotes the uniform distribution on the interval $[a,b]$. In this example, we set $\sigma_x = \sigma_{\dot{\theta}} = 0.2, \theta_0 = \frac{\pi}{20}$. We pick terminal $T = 2$. To carry out our computation, we evolve the Hamiltonian system (4.9) with initial samples drawn from $\widetilde{\rho}_0 = \rho_T$. We then apply our algorithm to compute for $\{\psi_\theta(\cdot, t)\}_{0 \le t \le T}$ as the solution to the HJ equaiton (4.8).

Moreover, upon evolving (4.9), we denote $\widetilde{\rho}_T$ as the distribution of terminal particles. We set the initial distribution of the swarm ρ_0 as $\widetilde{\rho}_T$. For any samples of ρ_0, we calculate the trajectory under our learned control $\{\psi_\theta(\cdot, t)\}_{0 \le t \le T}$ and compare it

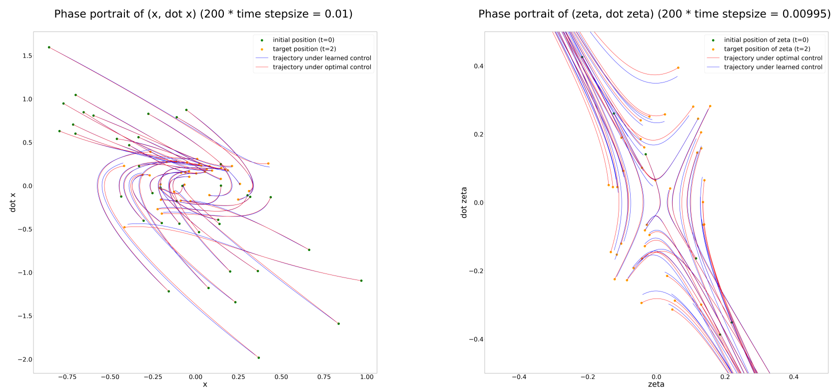


Figure 12: Plot of different trajectories under the learned control $\nabla\psi_\theta(\cdot, t)$ (blue) and the corresponding trajectories under the optimal control (red). Left: plot on $(x, \dot{x})$ plane; Right: plot on $(\theta, \dot{\theta})$.

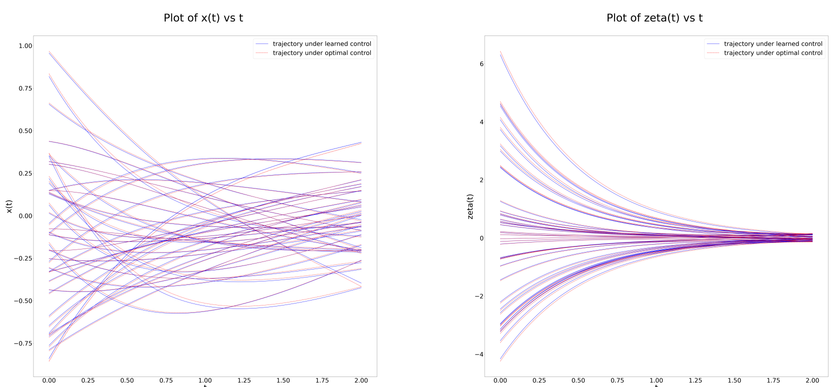


Figure 13: Plot of different trajectories under the learned control $\nabla\psi_\theta(\cdot, t)$ (blue) and the corresponding trajectories under the optimal control (red). Left: plot of x_t vs t; Right: plot of θ_t vs t.

with the trajectory under the optimal control (i.e., the trajectories solved from the Pontryagin's minimum principle (4.5)). The results are demonstrated in the Figure 12 and 13.

The L^2 loss decay curve shown in Figure 14 converges exponentially to 0, suggesting that our algorithm works properly on this example.

Two more numerical examples on Hamiltonian Jacobi equations with double well potential and Duffing oscillator can be found in the supplementary material.

5. Conclusion. In this paper, we propose a supervised learning algorithm to compute the first-order HJ equation by the density-coupling strategy. Such treatment

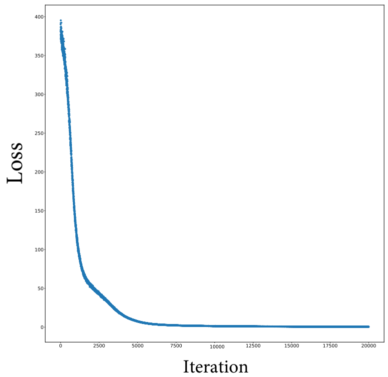


Figure 14: Plot of L^2 loss vs iteration in our training.

is inspired by the Wasserstein Hamiltonian flow, which bridges the HJ equation and its associated Hamiltonian ODE system. We then reformulate our method as a regression algorithm using the Bregman divergence. Furthermore, we provide error estimation on the L^1 residual term for the proposed method. The efficiency of our algorithm is verified by a series of numerical examples.

Multiple research directions may serve as the proceeding of this work. To name some of them,

- Our method can compute the solution to the HJ equation beyond the caustics, which is different from the commonly considered viscosity solution [11]. Is it possible to modify our algorithm at points at which caustics develop to compute the viscosity solution of the HJ equation?
- As mentioned in remark 2.1, our treatment leads to a new way to extend the classical solution of HJ equation beyond the caustics. What are the mathematical properties of such a solution? What is the relationship between this solution and the viscosity solution to the HJ equation?
- As discussed in section 3.2, we are not able to control the residual outside of the support of the swarm of particles. How can we propose the initial distribution ρ_0 such that the support of ρ_t covers the desired region on which we wish to obtain the accurate solution to the HJ equation?

We leave these topics to be investigated in the future.

REFERENCES

[1] Y. Achdou, G. Barles, H. Ishii, and G. L. Litvinov. *Hamilton-Jacobi equations: approximations, numerical analysis and applications*, volume 2074 of *Lecture Notes in Mathematics*. Springer, Heidelberg; Fondazione C.I.M.E., Florence, 2013. Lecture Notes from the CIME Summer School held in Cetraro, August 29–September 3, 2011, Edited by Paola Loreti and Nicoletta Anna Tchou, Fondazione CIME/CIME Foundation Subseries.

[2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

[3] Ramashis Banerjee and Arnab Pal. Stabilization of inverted pendulum on cart based on lqg optimal control. In *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pages 1–4. IEEE, 2018.

[4] R. Bellman. *Adaptive control processes: A guided tour*. Princeton University Press, Princeton,

NJ, 1961.

[5] J. D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[6] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker. An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. *J. Sci. Comput.*, 55(3):575–605, 2013.

[7] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[8] S.-N. Chow, W. Li, and H. Zhou. Wasserstein Hamiltonian flows. *Journal of Differential Equations*, 268(3):1205–1219, 2020.

[9] Y. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for state-dependent hamilton-jacobi equations, 2018.

[10] Y. T. Chow, W. Li, S. Osher, and W. Yin. Algorithm for Hamilton-Jacobi equations in density space via a generalized Hopf formula. *J. Sci. Comput.*, 80(2):1195–1239, 2019.

[11] M. Crandall and P. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

[12] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43(167):1–19, 1984.

[13] J. Cui, L. Dieci, and H. Zhou. A continuation multiple shooting method for Wasserstein geodesic equation. *SIAM J. Sci. Comput.*, 44(5):A2918–A2943, 2022.

[14] J. Cui, L. Dieci, and H. Zhou. Time discretizations of Wasserstein-Hamiltonian flows. *Math. Comp.*, 91(335):1019–1075, 2022.

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

[16] 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.*, 3:Paper No. 19, 26, 2016.

[17] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.*, 6(1):1–12, 2018.

[18] A. Fahim, N. Touzi, and X. Warin. A probabilistic numerical method for fully nonlinear parabolic pdes. *Annals of Applied Probability*, 21(4):1322–1364, 2011.

[19] Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River, 2002.

[20] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006. Structure-preserving algorithms for ordinary differential equations.

[21] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] S. Jin, H. Liu, S. Osher, and Y.-H. R. Tsai. Computing multivalued physical observables for the semiclassical limit of the schrödinger equation. *Journal of Computational Physics*, 205(1):222–241, 2005.

[24] S. Jin, P. Markowich, and C. Sparber. Mathematical and computational methods for semiclassical schrödinger equations. *Acta Numerica*, 20:121–209, 2011.

[25] S. Jin and S. Osher. A level set method for the computation of multivalued solutions to quasi-linear hyperbolic PDEs and Hamilton-Jacobi equations. *Commun. Math. Sci.*, 1(3):575–591, 2003.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Norman Lehtomaki, NJAM Sandell, and Michael Athans. Robustness results in linear-quadratic gaussian based multivariable control designs. *IEEE Transactions on Automatic Control*, 26(1):75–93, 1981.

[28] S. Liu, M. Jacobs, W. Li, L. Nurbekyan, and S. J. Osher. Computational methods for first-order nonlocal mean field games with applications. *SIAM J. Numer. Anal.*, 59(5):2639–2668, 2021.

[29] Z. Liu, H. Tilman, and U. Masahito. Neural networks fail to learn periodic functions and how to fix it. *Advances in Neural Information Processing Systems*, 33:1583–1594, 2020.

[30] T. Meng, W. Hao, S. Liu, S. Osher, and W. Li. Primal-dual hybrid gradient algorithms for

computing time-implicit Hamilton-Jacobi equations. *arXiv preprint arXiv:2310.01605*, 2023.

[31] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. A causality-free neural network method for high-dimensional Hamilton-Jacobi-Bellman equations. In *2020 American Control Conference (ACC)*, pages 787–793. IEEE, 2020.

[32] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing*, 43(2):A1221–A1247, 2021.

[33] N. Nüsken and L. Richter. Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2:1–48, 2021.

[34] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.

[35] W. Qiu, Q. Song, and G. Yin. Solving elliptic Hamilton-Jacobi-Bellman equations in a value space. *IEEE Control Systems Letters*, 5(1):55–60, 2021.

[36] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[37] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

[38] K. Sultan and A. Mirza. Inverted pendulum, analysis, design and implementation. *Visionaries Document*, 2003.

[39] Russ T. *Underactuated Robotics.* 2023.

[40] M. Tao. Explicit symplectic approximation of nonseparable hamiltonians: Algorithm and long time performance. *Phys. Rev. E*, 94:043303, Oct 2016.

[41] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694, 2003.

[42] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[43] Y. Zang, G. Bao, X. Ye, and H. Zhou. Weak adversarial networks for high-dimensional partial differential equations. *J. Comput. Phys.*, 411:109409, 14, 2020.