

A PARAMETERIZED WASSERSTEIN HAMILTONIAN FLOW APPROACH FOR SOLVING THE SCHRÖDINGER EQUATION

HAO WU*, SHU LIU†, XIAOJING YE‡, AND HAOMIN ZHOU§

Abstract. In this paper, we propose a new method to compute the solution of time-dependent Schrödinger equation (TDSE). Using push-forward maps and Wasserstein Hamiltonian flow, we reformulate the TDSE as a Hamiltonian system in terms of push-forward maps. The new formulation can be viewed as a generative model in the Wasserstein space, which is a manifold of probability density functions. Then we parameterize the push-forward maps by reduce-order models such as neural networks. This induces a new metric in the parameter space by pulling back the Wasserstein metric on density manifold, which further results in a system of ordinary differential equations (ODEs) for the parameters of the reduce-order model. Leveraging the computational techniques from deep learning, such as Neural ODE, we design an algorithm to solve the TDSE in the parameterized push-forward map space, which provides an alternative approach with the potential to scale up to high-dimensional problems. Several numerical examples are presented to demonstrate the performance of this algorithm.

Key words. Wasserstein Hamiltonian flow; generative model; Schrödinger equation;

1. Introduction. Schrödinger equation plays a fundamental role in the study of quantum physics. In this paper, we are concerned with its numerical simulation. To better explain our objectives and ideas, we take the following nonlinear time-dependent Schrödinger equation (TDSE) as an example:

$$(1.1) \quad i \frac{\partial}{\partial t} \psi(t, x) = -\frac{1}{2} \Delta \psi(t, x) + \frac{\delta}{\delta \rho} \mathcal{F}_R(|\psi(t, x)|^2, x) \cdot \psi(t, x),$$

where ψ is a complex-valued function defined on $[0, T] \times \mathbb{R}^d$, $\rho = |\psi|^2$, which can be viewed as a probability density function associated with ψ , \mathcal{F}_R is a functional of ρ , and $\frac{\delta}{\delta \rho} \mathcal{F}_R$ is the L^2 first variation of \mathcal{F}_R .

With different choices of \mathcal{F}_R , the TDSE (1.1) models various physical problems, for example, quantum harmonic oscillator [11], many particle interaction systems [12], and Bose-Einstein condensation [1, 29], among many others. There exists an extensive literature with many remarkable advancements on the theory, computation, and application of (1.1). However, its direct numerical simulation remains a difficult task, especially when the dimension d is high, e.g., $d \geq 4$. To mitigate the computational challenges, we introduce a novel formulation that takes advantage of the most recent developments in generative models [17] from machine learning and Wasserstein Hamiltonian flow (WHF) [10] related to optimal transport theory [39]. More specifically, there are two main objectives in this paper:

1. Reformulate the TDSE (1.1), via the Madelung transform, as a generative model using push-forward maps in conjunction with a WHF.
2. Propose a numerical method that applies parameterized reduced-order models, such as deep neural networks (DNNs), to solve the generative model. The result is a system of ordinary differential equations (ODEs) in the parameter space, which can be solved by symplectic numerical schemes.

Unlike the existing formulations of TDSEs, the derived generative model provides a dynamical description of the push-forward maps. The induced numerical method is complementary to existing approaches and can be used as an alternative algorithm that is cost-efficient for high-dimensional simulations.

In Section 2, we give a brief discussion of related work in the literature. We provide concise introductions to WHF and generative models, two crucial tools used in this investigation, in Section 3. The reformulation of TDSE into a generative model is presented in Section 4. The numerical method based on the formation of neural networks is introduced in Sections 5 and 6. We illustrate the performance of the method by several examples in Section 7, followed by a short discussion to conclude the paper.

2. Related work. Numerical simulations of TDSE have been conducted extensively with numerous algorithms based on classical methods such as finite difference [15, 16, 36, 38], spectral method [4, 5], and level set methods [22]. A survey of different numerical schemes with comprehensive comparisons on their properties can be found in [2]. Classical methods can provide efficient and accurate solutions when the dimension is small, i.e. $d \leq 3$. However, they suffer a serious issue known as the curse of dimensionality, referring to the exponential growth in the computational cost with respect to the dimension d , especially when d is large. Compared to classical methods, particle dynamics-based simulations such as the quantum trajectory method

*Charlotte, NC, USA (hwu406@gmail.com).

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

‡Department of Mathematics and Statistics, Georgia State University, Atlanta, GA, USA(xye@gsu.edu).

§School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA(hmzhou@gatech.edu).

(QTM) [38] and smooth particle hydrodynamics (SPH) [25] have been proposed. They are Monte Carlo approaches designed using the formulation of Bohmian mechanics [6, 13, 30], which is a major source of motivations for our investigation. For example, the SPH generates samples from the initial distribution and simulates the sample dynamics under the quantum potential. These sample-based approaches scale well with dimensions, but the approximation to the density function as well as its gradient evaluation remain a major challenge in the computation.

In recent years, machine learning-based methods, such as the physics-informed neural network (PINN) [32], deep Ritz method (DRM) [14], weak adversarial network (WAN) [44], and many others [19, 35] have shown promising results of using neural networks to solve partial differential equations (PDEs) in high dimensions. Some of them have been adopted to solve Schrödinger equation. For example, PINN has been used in computing the solution of TDSE [31, 42]. A specific class of neural networks has been proposed to represent the many-electron wave function following the Pauli exclusion principle [20, 21, 28], which shows powerful and promising results for estimating ground state energies in many-electron systems. More research has been reported on the application of neural networks to simulate many body problems [8] and molecular dynamics [24]. Those studies are among the other sources that motivated this work.

Recently, a method called Deep Stochastic Mechanics (DSM) [27], which modifies the formulations in [26] and [18], was developed to generate samples following the time-evolving squared magnitude (density) of the solution to TDSE. While Madelung transform induces the coupled evolution PDEs of the phase and logarithm of density, the gradients of them satisfy a new coupled PDEs, which are solved by the PINN approach in DSM. These gradients can add up to form the drift of a stochastic differential equation (SDE), whose sample trajectories follow the density of TDSE. In contrast, we directly use the reformulation based on the Madelung transform and approximate the push-forward map by solving an ODE system which is free of network training.

Our formulation and algorithm are directly inspired by recent advancements in parameterized WHF [43], in which a numerical method based on neural network parameterization has been proposed for WHF in conjunction with generative models. It is known that, by using Madelung transform, the TDSE can be rewritten as a continuity equation coupled with a Hamilton-Jacobi equation, which forms a WHF. Hence, we can apply the parameterized WHF to solve the TDSE, providing substantial new improvements to handle the Fisher information that appeared in the Hamilton-Jacobi equation. To this end, we borrow a tool from generative models called Neural ODE [9] that can compute Fisher information efficiently.

3. Mathematical Tools. In this section, we give concise introductions to the WHF and Neural ODE, which are two main tools used to establish the reformulation of TDSE as generative models.

3.1. Wasserstein Hamiltonian Flow. Let M be a smooth manifold without boundary. For simplicity, we can assume $M = \mathbb{R}^d$ in our discussion. We consider the set of density functions defined on M with bounded second moment:

$$(3.1) \quad \mathcal{P}(M) = \left\{ \rho \in C^\infty(M) : \rho \geq 0, \int_M \rho dx = 1, \int_M |x|^2 \rho dx < \infty \right\}.$$

Given a Hamiltonian $\mathcal{H}(\rho, \Phi) = \frac{1}{2} \int_M |\nabla \Phi|^2 \rho(x) dx + \mathcal{F}(\rho)$ where $\Phi \in C^\infty$ and \mathcal{F} is an energy functional on $\mathcal{P}(M)$, the WHF with Hamiltonian \mathcal{H} is described by the following system,

$$(3.2a) \quad \partial_t \rho = \frac{\delta}{\delta \Phi} \mathcal{H}(\rho, \Phi),$$

$$(3.2b) \quad \partial_t \Phi = -\frac{\delta}{\delta \rho} \mathcal{H}(\rho, \Phi),$$

with initial values

$$(3.3) \quad \rho(0, x) = \rho_0(x) \quad \text{and} \quad \Phi(0, x) = \Phi_0(x).$$

Many classical PDEs can be formulated as WHF with different energy functionals \mathcal{F} [10]. The Schrödinger equation is among them. More details are given next.

3.2. Schrödinger equation and Madelung transform. Consider the Madelung transform

$$(3.4) \quad \psi(t, x) = \sqrt{\rho(t, x)} e^{i\Phi(t, x)},$$

which is a nonlinear symplectic transform mapping a complex-valued wave function to the pair of real-valued functions (ρ, Φ) , see [33] for more details. Using the Madelung transform, the Schrödinger equation (1.1) can

be reformulated as the following Madelung system by plugging (3.4) into (1.1) and matching the real and imaginary parts on its two sides:

$$(3.5a) \quad \partial_t \rho + \nabla \cdot (\rho \nabla \Phi) = 0,$$

$$(3.5b) \quad \partial_t \Phi + \frac{1}{2} |\nabla \Phi|^2 = -\frac{\delta}{\delta \rho} \mathcal{F}_R(\rho) + \frac{1}{2} \frac{\Delta \sqrt{\rho}}{\sqrt{\rho}},$$

with initial values $\rho(0, x) = |\psi(0, x)|^2$ and $\Phi(0, x)$ being the phase of $\psi(0, x)$ at every $x \in M$. Due to the Madelung transform, we assume $\rho > 0$ almost everywhere in M for any time t hereafter.

Remark 3.1. We always assume that Φ in (3.5) is single-valued, thereby eliminating the potential inequivalence between the Schrödinger equation and the Madelung system [41]. For more detailed discussions, we refer the reader to [41, 27] and the references therein.

From the perspective of Lagrangian mechanics on the density manifold $\mathcal{P}(M)$, the Schrödinger equation arises as the critical point of an action functional [23]. Furthermore, an optimal transport approach was proposed in [40] to interpret the Madelung system (3.5) as the Hamiltonian flow (3.2) associated with the following Hamiltonian:

$$(3.6) \quad \mathcal{H}(\rho, \Phi) = \int_{\mathbb{R}^d} \frac{1}{2} |\nabla \Phi(x)|^2 \rho(x) dx + \mathcal{F}_R(\rho) + \frac{1}{8} \mathcal{F}_Q(\rho),$$

where the last term $\mathcal{F}_Q = \int_{\mathbb{R}^d} |\nabla \log \rho(x)|^2 \rho(x) dx$ is known as the Fisher information in the context of quantum mechanics [34], and its L^2 first variation is the quantum potential [7]:

$$(3.7) \quad \frac{\delta}{\delta \rho} \mathcal{F}_Q(\rho, x) = -2\Delta \log \rho - |\nabla \log \rho|^2 = -4 \frac{\Delta \sqrt{\rho}}{\sqrt{\rho}}.$$

Our reformulation of (1.1) as a generative model is based on its WHF formulation (3.5).

3.3. Neural ODE. Generative models in machine learning include a large class of algorithms that aim to map samples from an initial distribution with density λ to samples of a target distribution with density μ . It is often easy to obtain samples from the initial distribution such as Gaussian while difficult to sample from the target one, about which only partial information is accessible. Well-known examples of generative models include generative adversarial nets (GAN) [17], Wasserstein GAN (WGAN) [3], and diffusion generative model [37]. A key component of a generative model is a push-forward map $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ defining a correspondence between samples $z \rightarrow T(z)$ where z is sampled from λ , denoted by $x \sim \lambda$. The map T induces a push-forward distribution with density $T_\# \lambda$ through

$$\int_E T_\# \lambda(x) dx = \int_{T^{-1}(E)} \lambda(z) dz \quad \text{for any measurable set } E \subset \mathbb{R}^d,$$

where $T^{-1}(E)$ is the pre-image of E . Correspondingly, we also have the change of variable formula to explicitly evaluate the induced push-forward density:

$$(3.8) \quad T_\# \lambda(z) = \lambda \circ T^{-1}(z) \det \left(\frac{d}{dz} T^{-1}(z) \right) \quad \forall z \in \mathbb{R}^d.$$

In a generative model, T is realized by a neural network and it is desirable to achieve $T_\# \lambda = \mu$.

To reformulate the Schrödinger equation (1.1) in a generative model, we select a special type of neural network called Neural Ordinary Differential Equations (Neural ODE) as the push-forward map T . Neural ODE was first proposed in [9]. It is defined through the solution map of a parameterized ODE. More precisely, let $[0, t^*]$ be a fixed time interval and $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a neural network with parameters $\theta \in \Theta \subset \mathbb{R}^m$ (m is the number of parameters in f_θ), the push-forward map given by the neural ODE is defined as $T_\theta(z) = w(t^*)$, where $w(t)$ satisfies

$$(3.9) \quad \dot{w}(t) = f_\theta(w(t)), \quad w(0) = z,$$

It is shown in numerous studies that Neural ODE has many desirable properties like strong approximation power and efficient implementations. More importantly, we select Neural ODE because it is easily invertible due to the uniqueness of the solution for the ODE (3.9), and the logarithmic density function can be conveniently evaluated through the adjoint equation [9]. Both features are crucial for us to express (1.1) as a continuous-time push-forward map and compute the Fisher information term appeared in (3.7). A brief discussion on how to implement the Fisher information term through Neural ODE is provided in Appendix A.

4. Expressing the Schrödinger equation by push-forward map. In this section, we present the reformulation of the TDSE (1.1) in terms of time-dependent push-forward map T .

Let $\lambda > 0$ be the density of a reference distribution on \mathbb{R}^d . We define

$$(4.1) \quad \mathcal{O} = \{T \in L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda) : T \text{ is diffeomorphism from } \mathbb{R}^d \text{ to } \mathbb{R}^d.\}$$

as the space of diffeomorphisms. We denote $\mathcal{T}_T \mathcal{O} := C^\infty(\mathbb{R}^d; \mathbb{R}^d) \cap L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda)$ as its tangent space at T . For any $T \in \mathcal{O}$ and $\sigma_i \in \mathcal{T}_T \mathcal{O}$ ($i = 1, 2$), we define a metric \mathcal{G} on the tangent bundle \mathcal{TO} as

$$(4.2) \quad \mathcal{G}(T)(\sigma_1, \sigma_2) = \int \sigma_1(z)^\top \sigma_2(z) \lambda(z) dz.$$

Then, we introduce a Lagrangian \mathbb{L} defined on \mathcal{TO} as

$$(4.3) \quad \mathbb{L}(T, \sigma) = \frac{1}{2} \mathcal{G}(T)(\sigma, \sigma) - \mathcal{F}(T_\# \lambda).$$

Now, for a time-dependent curve $\{T_t\}$ ($0 \leq t \leq t_0$) on \mathcal{O} , we denote $\dot{T}_t = \frac{dT_t}{dt} \in \mathcal{T}_{T_t} \mathcal{O}$ as the time derivative of T_t at t , and denote $\mathcal{F}(\rho) = \mathcal{F}_R(\rho) + \frac{1}{8} \mathcal{F}_Q(\rho)$. In order to introduce the Hamiltonian flow on \mathcal{O} , we first consider the stationary point associated with the following action functional $\mathcal{I} : \mathcal{O}_{0,t_0} \rightarrow \mathbb{R}$

$$(4.4) \quad \mathcal{I}(\{T_t\}) = \int_0^{t_0} \mathbb{L}(T_t, \dot{T}_t) dt, \quad \{T_t\} \in \mathcal{O}_{0,t_0},$$

where we define \mathcal{O}_{0,t_0} as the set of smooth paths $\{T_t\}_{0 \leq t \leq t_0}$ on \mathcal{O} with $T_{0\#} \lambda = \rho_0$ and $T_{t_0\#} \lambda = \rho_{t_0}$. Here, ρ_0, ρ_{t_0} are the density functions obtained by the Madelung transform (3.5) from the wavefunction $\psi(t, x)$ at $t = 0$ and $t = t_0$ respectively. The stationary point associated with (4.4), if it exists, satisfies the Euler-Lagrange equation, as stated in the following theorem. For brevity, we omit the subscript t and use T and \dot{T} to denote T_t and \dot{T}_t , respectively, in the following discussion.

THEOREM 4.1 (Hamiltonian system in the space of diffeomorphisms). *The critical point of (4.4), if exists, satisfies*

$$(4.5) \quad \ddot{T}(z) = -\nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda(\cdot), \cdot) \circ T(z).$$

Equivalently, by introducing the momentum $\Lambda := \frac{\partial \mathbb{L}(T, \dot{T})}{\partial \dot{T}} = \lambda \dot{T}$, equation (4.5) can also be written as a first order Hamiltonian system

$$(4.6) \quad \begin{aligned} \frac{d}{dt} T &= \frac{1}{\lambda} \Lambda, \\ \frac{d}{dt} \Lambda &= -\nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda(\cdot), \cdot) \circ T(z) \lambda(z), \end{aligned}$$

with Hamiltonian $\mathbb{H}(T, \Lambda) = \mathcal{G}(T)(\dot{T}, \dot{T}) - \mathbb{L}(T, \dot{T}) = \frac{1}{2} \mathcal{G}(T)(\frac{1}{\lambda} \Lambda, \frac{1}{\lambda} \Lambda) + \mathcal{F}(T_\# \lambda)$. Furthermore, the push-forward density $T_\# \lambda$ solves the WHF (3.2) whose Hamiltonian is given by (3.6).

Proof. For convenience, we denote $\mathbb{F}(T) := \mathcal{F}(T_\# \lambda)$. Following the definition of $\mathbb{L}(T, \dot{T})$ in (4.3), we have

$$(4.7) \quad \mathbb{L}(T, \dot{T}) = \frac{1}{2} \mathcal{G}(T)(\dot{T}, \dot{T}) - \mathbb{F}(T),$$

The Euler-Lagrange equation satisfied by the critical point of (4.4) is

$$(4.8) \quad \frac{\delta}{\delta T} \mathbb{L}(T, \dot{T}) = \frac{d}{dt} \frac{\delta}{\delta \dot{T}} \mathbb{L}(T, \dot{T}).$$

Let us first examine its right-hand side. Following the definition given in (4.2), we note that the metric \mathcal{G} is the same for all T , implying that \mathcal{G} is independent of T . This leads to $\frac{\delta}{\delta T} \mathbb{L}(T, \dot{T}) = \dot{T}(z) \lambda(z)$. Hence we have

$$(4.9) \quad \frac{d}{dt} \frac{\delta}{\delta \dot{T}} \mathbb{L}(T, \dot{T}) = \ddot{T}(x) \lambda(z).$$

We next compute the left-hand side of (4.8). Since the metric is independent of T , we have $\frac{\delta}{\delta T}\mathbb{L} = -\frac{\delta}{\delta T}\mathbb{F}$. In the following, we calculate $\frac{\delta}{\delta T}\mathbb{F}(T)$.

Let $S \in \mathcal{O}$ be arbitrary and define $T_\varepsilon^S(x)$ to be the solution $X(\varepsilon)$ of the ODE:

$$(4.10) \quad \begin{cases} X'(t) = (S \circ T^{-1})(X(t)), & 0 \leq t \leq \varepsilon, \\ X(0) = x = T(z) \sim T_\# \lambda, \end{cases}$$

at the end time $t = \varepsilon$, where $z \sim \lambda$. In addition, T_ε^S is a variation of T equivalent to $T + \varepsilon S$ up to the first order because for any z there are $X(0) = x = T(z)$ and

$$\begin{aligned} X(\varepsilon) &= X(0) + X'(0)\varepsilon + o(\varepsilon) \\ &= x + (S \circ T^{-1})(x)\varepsilon + o(\varepsilon) \\ &= T(z) + \varepsilon S(z) + o(\varepsilon) \\ &= (T + \varepsilon S)(z) + o(\varepsilon). \end{aligned}$$

We call ρ_t the density of $X(t)$ for $0 \leq t \leq \varepsilon$. Then the continuity equation of ρ_t corresponding to (4.10) is

$$(4.11) \quad \begin{cases} \partial_t \rho_t(x) + \nabla_X \cdot [\rho_t(x)(S \circ T^{-1})(x)] = 0, & 0 \leq t \leq \varepsilon, \\ \rho_0 = T_\# \lambda. \end{cases}$$

Now we derive the L^2 first-variation $\frac{\delta}{\delta T}\mathbb{F}(T)$. We have

$$\begin{aligned} \left\langle \frac{\delta}{\delta T}\mathbb{F}(T), S \right\rangle &= \frac{d}{d\varepsilon} \mathbb{F}(T_\varepsilon^S) \Big|_{\varepsilon=0} = \frac{d}{d\varepsilon} \mathcal{F}(T_\varepsilon^S \# \lambda) \Big|_{\varepsilon=0} \\ &= \lim_{\varepsilon \rightarrow 0} \int \frac{\delta}{\delta \rho} \mathcal{F}(\rho_0)(x) \frac{\rho_\varepsilon(x) - \rho_0(x)}{\varepsilon} dx \\ &= \int \frac{\delta}{\delta \rho} \mathcal{F}(\rho_0)(x) \partial_t \rho_\varepsilon(x) \Big|_{\varepsilon=0} dx. \end{aligned}$$

By the continuity equation (4.11), we know

$$\partial_t \rho_\varepsilon(x) \Big|_{\varepsilon=0} = -\nabla \cdot [\rho_0(x)(S \circ T^{-1})(x)].$$

Therefore

$$\begin{aligned} \left\langle \frac{\delta}{\delta T}\mathbb{F}(T), S \right\rangle &= \int \frac{\delta}{\delta \rho} \mathcal{F}(\rho_0)(x) \left(-\nabla_X \cdot [\rho_0(x)(S \circ T^{-1})(x)] \right) dx \\ &= \int \nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(\rho_0)(x) (S \circ T^{-1})(x) \rho_0(x) dx \\ &= \int \nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda)(T(z)) S(z) \lambda(z) dz, \end{aligned}$$

where the last equation is due to the change of variable $x = T(z)$. Since S is arbitrary, we must have

$$\frac{\delta}{\delta T}\mathbb{F}(T) = \nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda)(T(z)) \lambda(z).$$

This leads to

$$\frac{\delta}{\delta T}\mathbb{L}(T, \dot{T}) = -\frac{\delta}{\delta T}\mathbb{F}(T) = -\nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda(\cdot), \cdot) \circ T(z) \lambda(z).$$

Hence, the Euler-Lagrange equation (4.8) becomes

$$(4.12) \quad \frac{d}{dt} \left(\dot{T}(z) \lambda(z) \right) = \nabla_X \frac{\delta}{\delta \rho} \mathcal{F}(T_\# \lambda(\cdot), \cdot) \circ T(z) \lambda(z)$$

Since $\lambda(z) > 0$ for all $z \in \mathbb{R}^d$, this proves (4.5).

Following Proposition 2 in [10], we know that the push-forward density $T_\# \lambda$ satisfies the WHF (3.2), which gives (4.6). This completes the proof. \square

For convenience, we call the equation (4.5) or system (4.6) the generative model reformulation of TDSE (1.1). We would like to mention that the derivation of (4.5) is also inspired by the *Bohmian mechanics* [7], which aims at describing the particle dynamics governed by the Schrödinger equation. More precisely, each particle \mathbf{X} moves according to a dynamical system

$$(4.13) \quad \ddot{\mathbf{X}} = -\nabla_{\mathbf{X}} \frac{\delta}{\delta \rho} \mathcal{F}(\rho, \mathbf{X}),$$

assisted by the guiding function ρ , which is obtained by the Schrödinger equation and Madelung transform. Clearly, equations (4.13) and (4.5) share similarities in their formulation. However, they have significant distinctions. By definitions, we see that the system (4.5) is defined in the space of diffeomorphisms \mathcal{O} while the equation (4.13) is defined in \mathbb{R}^n . More importantly, the operator equation (4.5) is self-contained. Its trajectory is fully determined if the initial conditions of T and Λ are given. This implies that it can produce particle motions as well as density maps at the same time once T_t is known. In sharp contrast, (4.13) is not self-contained. It requires extra knowledge of the guiding function ρ to fully determine the motions of the particles. It is worth noting that investigating the well-posedness of the operator equation (4.5) and the Vlasov-type ODE (4.13) lies beyond the scope of this study and is left as a potential direction for future research.

To further digest the system (4.6), we consider its implication in the *energy eigen-state*. In this case, the generative model (4.6) reduces to a trivial (constant) solution in the space of diffeomorphisms \mathcal{O} . More precisely, at the energy eigen-state $\psi_0(t, x) = e^{-iE_0 t} \sqrt{\rho_0(x)}$ with eigen-energy E_0 , $\phi(x) = \sqrt{\rho_0(x)}$ solves

$$(4.14) \quad E_0 \phi(x) = -\frac{1}{2} \Delta \phi(x) + \frac{\delta}{\delta \rho} \mathcal{F}_R(|\phi(x)|^2, x) \cdot \phi(x).$$

By (3.7) and assuming $\phi > 0$, we have

$$\begin{aligned} \frac{\delta}{\delta \rho} \mathcal{F}_Q(\rho_0, x) &= -2\Delta \log(\phi^2) - |\nabla \log(\phi^2)|^2 \\ &= -4 \frac{\phi \Delta \phi - |\nabla \phi|^2}{\phi^2} - 4 \left| \frac{\nabla \phi}{\phi} \right|^2 \\ &= -4 \frac{\Delta \phi}{\phi}, \end{aligned}$$

Together with (4.14), we have

$$\begin{aligned} \frac{\delta}{\delta \rho} \mathcal{F}(\rho_0, x) &= \frac{\delta}{\delta \rho} \mathcal{F}_R(\rho_0, x) + \frac{1}{8} \frac{\delta}{\delta \rho} \mathcal{F}_Q(\rho_0, x) \\ &= \frac{E_0 \phi(x) + \frac{1}{2} \Delta \phi(x)}{\phi} + \frac{1}{8} \left(-4 \frac{\Delta \phi}{\phi} \right) \\ &= E_0 \end{aligned}$$

Hence (4.5) becomes

$$(4.15) \quad \ddot{T} = -\nabla_X E_0 = 0.$$

This corresponds to a trivial solution in the space of diffeomorphisms (constant speed).

5. Parameterized generative model of TDSE. The generative model for TDSE (4.6) is formulated in the space of diffeomorphisms \mathcal{O} and its tangent bundle, which are infinite dimensional spaces. To simulate its solution, we must approximate it in a finite dimensional space. To this end, we follow the ideas proposed in the parameterized WHF (PWHF) [43].

Instead of arbitrary $T \in \mathcal{O}$, we restrict the consideration in a subspace of \mathcal{O} in which each T can be parameterized by θ , denoted as T_θ . There may be different choices for T_θ , such as finite element or Fourier approximations. In this study, we choose neural networks, especially the Neural ODEs described in Section 3. In other words, \mathcal{O} is replaced by $\mathcal{O}_\theta = \{T_\theta : \theta \in \Theta \subset \mathbb{R}^m\}$ in the problem setting in Section 4. The tangent space of \mathcal{O}_θ is $\mathcal{T}_\theta \mathcal{O} = \text{span}\{\partial_{\theta_k} T_\theta : k = 1, \dots, m\}$; the metric (4.2) becomes

$$(5.1) \quad G(\theta) = \int \partial_\theta T_\theta(z)^\top \partial_\theta T_\theta(z) \lambda(z) dz;$$

and the Lagrangian in the parameter space Θ is reformulated to

$$(5.2) \quad L(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^\top G(\theta) \dot{\theta} - F(\theta),$$

where $F(\theta) := \mathcal{F}(\rho_\theta)$. Hence the variational formulation (4.4) is reduced to

$$(5.3) \quad \mathcal{I}^\Theta(\theta) = \inf_{\theta} \left\{ \int_0^{t_0} L(\theta(t), \dot{\theta}(t)) dt : \rho_{\theta(0)} = \rho_0, \rho_{\theta(t_0)} = \rho_{t_0} \right\}.$$

Mimicking the derivation of PWHF as detailed in [43], we obtain the parameterized generative model for TDSE (4.6).

PROPOSITION 5.1. *The critical point of $\mathcal{I}^\Theta(\theta)$ defined in (5.3) satisfies a Hamiltonian system*

$$(5.4a) \quad \dot{\theta} = G^\dagger p,$$

$$(5.4b) \quad \dot{p} = \frac{1}{2} [(G^\dagger p)^\top (\partial_{\theta_k} G) G^\dagger p]_{k=1}^m - \nabla_\theta F(\theta),$$

where G^\dagger is the Penrose-Moore pseudo inverse of G . The corresponding Hamiltonian is

$$(5.5) \quad H(\theta, p) = \frac{1}{2} p^\top \hat{G}^\dagger(\theta) p + F(\theta).$$

Solving (5.4) can provide an approximate solution $T_{\theta(t)}$ to the original flow (4.5) defined in the space of diffeomorphisms \mathcal{O} . This further gives the approximate density $\rho_\theta = T_{\theta^\#} \lambda$ in $\mathcal{P}(M)$. Similarly, we can approximate $\nabla \Phi(t, x)$ by $\partial_\theta T_{\theta(t)} \circ T_{\theta(t)}^{-1}(\cdot) p(t)$. Together, they form the foundation for our algorithm to compute the parameterized generative model for TDSE.

6. Numerical algorithm. To simulate the parameterized generative model of TDSE (5.4), we suggest the numerical method presented in Algorithm 6.1, which is based on a semi-implicit symplectic Euler scheme proposed in [43]. Here we omit its details.

Algorithm 6.1 Parameterized TDSE solver

Initialize neural network T_θ with parameters θ^0 at $t = 0$.

Initialize $p^0 = \nabla_\theta \mathbb{E}_{z \sim \lambda} [\Phi(0, T_\theta(z))]$ with $\Phi(0, x) = -i \log \frac{\psi(0, x)}{|\psi(0, x)|}$.

Set terminal time t_1 , number of steps K and fixed point iteration rate γ , define the step size as $h = t_1/K$.

for $l = 0, \dots, K - 1$ **do**

Sample $\{z_1, \dots, z_N\}$ from λ , and compute $X_i = T_{\theta^l}(z_i)$ as well as $\rho_{\theta^l}(x_i) = T_{\theta^l \#} \lambda(x_i)$

Apply MINRES to solve $\xi^{l,0}$ from equation $G(\theta^l) \xi = p^l$, set $\alpha^{l,0} = \theta^l$

for $j = 1, \dots, n_{in}$ **do**

Update $\alpha^{l,j} = \theta^l + h \xi^{l,j}$

Update $\xi^{l,j+1} = \xi^{l,j} - \gamma (G(\alpha^{l,j}) \xi^{l,j} - p^l)$

end for

Set $\theta^{l+1} = \alpha^{l, n_{in}}$, $\eta^{l+1} = \xi^{l, n_{in}}$

Compute $X_i^{l+1} = T_{\theta^{l+1}}(z_i)$ as samples from $\rho_{\theta^{l+1}}$, evaluate $\nabla_\theta F(\theta^{l+1})$ through samples $\{X_i^{l+1}\}$

Set $p^{l+1} = p^l + \frac{h}{2} [(\eta^{l+1})^\top \partial_{\theta_k} G \eta^{l+1}]_{k=1}^m - h \nabla_\theta F(\theta^{l+1})$

end for

Output: Solution at discrete time spots: $(\theta(lh) := \theta^l, p(lh) := p^l)$ for $l = 0, \dots, K$.

Algorithm 6.1 has several important features. First, the algorithm does not require spatial discretization or basis, and is sampling-based and can be executed as long as samples $\{z_1, \dots, z_N\}$ from the reference distribution are available.

This is a particularly important and the main reason that Algorithm 6.1 can scale up to high-dimensional cases. In practice, samples from the reference distribution can be taken from the same or different sets at each time step. Second, the parameters θ are computed by iterative methods for linear least squares problems. This is different from typical network training where the parameters are solved from challenging large-scale non-convex optimization problems. In particular, Algorithm 6.1 can be implemented without explicitly creating

the matrix G , because it only requires matrix vector multiplication. The minimal residual (MINRES) method is chosen for convenience. It can be replaced by other numerical solvers as long as the least-squares solution for the linear system can be obtained efficiently. Third, compared to the algorithm proposed in [43], we develop a new approach to evaluate the Fisher information appeared in $F(\theta)$ by leveraging the properties of the Neural ODE T_θ . A detailed PyTorch-based algorithm is given in Appendix A.

There are many options to initialize θ^0 in Algorithm 6.1 depending on the problem settings. We can select T_{θ^0} the identity map if ρ_0 is the same as the reference density, or samples at $t = 0$ are given. In the latter case, the algorithm can be carried out without the reference distribution. In general, we can initialize θ^0 by minimizing the difference between ρ_{θ^0} and ρ_0 , for example,

$$\theta^0 = \underset{\theta}{\operatorname{argmin}} \{ \mathcal{D}_{\text{KL}}(\rho_0 \| \rho_\theta) \},$$

where \mathcal{D}_{KL} stands for the Kullback-Leibler divergence between two probability densities. Other distance or divergence can be used as well. Once ρ^0 is available, p^0 can be initialized by $\nabla_\theta \mathbb{E}_{z \sim \lambda} [\Phi(0, T_\theta(z))]$, where the initial phase $\Phi(0, x)$ is obtained through the Madelung transform.

7. Numerical results. In this section, we present three examples to demonstrate the performance of the proposed formulation and algorithm. The first is the quantum harmonic oscillator; the second is the Gross-Pitaevskii equation (GPE); and the third is a three-particle system. We focus on their numerical simulations. Our computation is carried out on a desktop computer with an NVIDIA RTX-4080s GPU (16 GB memory) and CUDA enabled.

7.1. Coherent state solution to the quantum harmonic oscillator. Although the appearance of the parameterized system (5.4) is different from the TDSE given in (1.1), it can be verified that they are theoretically equivalent in some simplified situations. For example, let us consider the linear Schrödinger equation with quadratic potential

$$(7.1) \quad i\partial_t \psi = -\frac{1}{2}\Delta \psi + \frac{x^2}{2}\psi, \quad x = (x_1, x_2) \in \mathbb{R}^2,$$

$$(7.2) \quad \psi(0, x) = \frac{1}{\pi^{1/4}} \exp\left(-\frac{1}{2}\left[(x_1 - \sqrt{2})^2 + x_2^2\right] + i\sqrt{2}x_2\right).$$

This equation (7.1) is known as the quantum harmonic oscillator whose coherent state solution can be verified analytically as

$$(7.3) \quad \psi(t, x) = \frac{1}{\pi^{1/4}} \exp\left(-\frac{1}{2}|x - \sqrt{2}\alpha(t)|^2 + i\sqrt{2}\beta(t)^\top x + ic(t)\right),$$

where $\alpha(t) = (\cos t, \sin t)^\top$, $\beta(t) = (-\sin t, \cos t)^\top$, and $c(t)$ is a time dependent function satisfying $c(0) = 0$.

If we select an affine transformation as the push-forward map, i.e.,

$$(7.4) \quad T_\theta(z) = A(t)z + b(t), \quad \text{with parameters } \theta(t) = (A(t), b(t)), \quad A(t) \in \mathbb{R}^{d \times d}, \quad b(t) \in \mathbb{R}^d,$$

and set the reference density λ as

$$(7.5) \quad \lambda(z) = \frac{1}{\sqrt{\pi}} \exp(-|z|^2),$$

the system for the parameters (5.4) becomes

$$(7.6a) \quad \ddot{A}(t) = -\nabla_A \left(\frac{1}{4} \operatorname{Tr} [A(t)^\top A(t) + A(t)^{-\top} A(t)^{-1}] \right), \quad \ddot{b}(t) = -\nabla_b \left(\frac{1}{2} b(t)^\top b(t) \right),$$

$$(7.6b) \quad A(0) = I_{d \times d}, \quad \dot{A}(0) = 0_{d \times d}, \quad b(0) = (\sqrt{2}, 0)^\top, \quad \dot{b}(0) = (0, \sqrt{2})^\top.$$

Its solution can be verified as

$$(7.7) \quad A(t) = I_{d \times d}, \quad b(t) = \sqrt{2}\alpha(t).$$

Plugging (7.7) into (3.8), we get the parameterized density function

$$(7.8) \quad \rho_\theta(x) = \frac{1}{\sqrt{\pi}} \exp(-|x - \sqrt{2}\alpha(t)|^2),$$

which equals to the density function given by the true solution (7.3).

We use Algorithm 6.1 to solve the parameter ODEs (7.6) for $d = 2$. A total of $N = 20,000$ samples are used in the calculation. The matrix $A(t)$ remains as an identity and the vector $b(t) - b(0)$ is depicted in Figure 1. Clearly, the numerical solution coincides with the exact solution as our analysis suggested in this example.

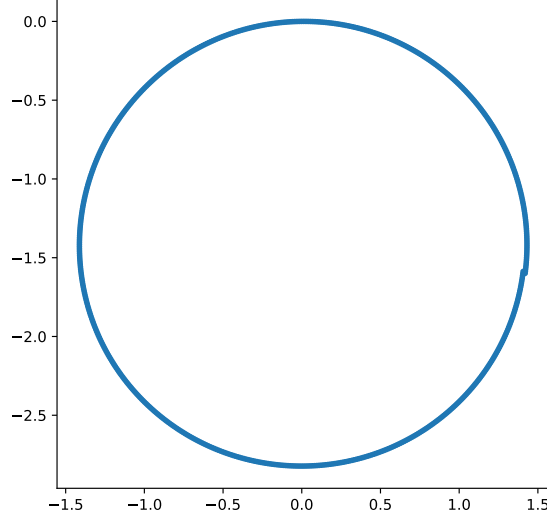


Fig. 1: The blue curve shows the trajectory of $b(t) - b(0)$ for the harmonic oscillator with quadratic potential example (7.6).

7.2. Gross-Pitaevskii equation. The Gross-Pitaevskii equation (GPE) is

$$(7.9) \quad i \frac{\partial}{\partial t} \psi(t, x) = -\frac{1}{2} \Delta \psi(t, x) + V(x) \psi(t, x) + \gamma |\psi(t, x)|^2 \psi(t, x),$$

where $\gamma \in \mathbb{R}$ is a constant. The GPE is a special case of the TDSE (1.1) if taking $\mathcal{F}_R(\rho) = \int V(x) \rho(x) dx + \frac{\gamma}{2} \int \rho(x)^2 dx$. When $V(x) \equiv 0$, $d = 1$, $\gamma = -2$, the GPE (7.9) admits a set of solutions with explicit expression given by

$$(7.10) \quad \psi(t, x) = \sqrt{\mu/2} \exp \left(i \left[\frac{1}{2} vx + \frac{1}{2} \left(\mu - \frac{1}{4} |v|^2 \right) t \right] \right) \operatorname{sech} \left(\sqrt{\mu} \left(x - \frac{1}{2} vt \right) \right),$$

where $v \in \mathbb{R}, \mu \in \mathbb{R}^+$ are constants and $\operatorname{sech}(y) = \frac{2e^y}{e^{2y} + 1}$.

Now we apply Algorithm 6.1 to the GPE (7.9) whose exact solution is given by (7.10). We use Neural-ODE architecture to parameterize the push-forward map. The right-hand side of the Neural-ODE consists of two hidden layers, each containing 50 neurons and the hyperbolic tangent as the activation function. We generate 10,000 samples for the computation of G metric and potential energy \mathcal{F} . The time step size is taken as $h = 0.005$ unless otherwise stated. We compute the solutions for $\theta(t)$ and use the push-forward $T_{\theta(t)}$ to generate samples for $\rho_{\theta(t)}$. The initial condition is taken from (7.10) with $t = 0$. In Figure 2, we compare the analytical solution for $\rho(t)$ (blue curve) with the histogram generated by $T_{\theta(t)}$ (orange bars) at different time snapshots. The computed solution matches the analytic solution well.

In the next two experiments, we apply Algorithm 6.1 to solve the GPE in $d = 3$ and $d = 6$, respectively. We also set $V(x) \equiv 0, \gamma = -2$ in both experiments. The initial values are taken as the same as $d = 1$ with dimension adjustments. Figure 4 shows the histograms for the first components of the samples generated by the push-forward map T_{θ} in the $d = 3$ case. Similarly, the histograms are plotted in Figure 5 for $d = 6$. We also plot the center of distribution as a function of time in Figure 3 for the $d = 3$ experiment.

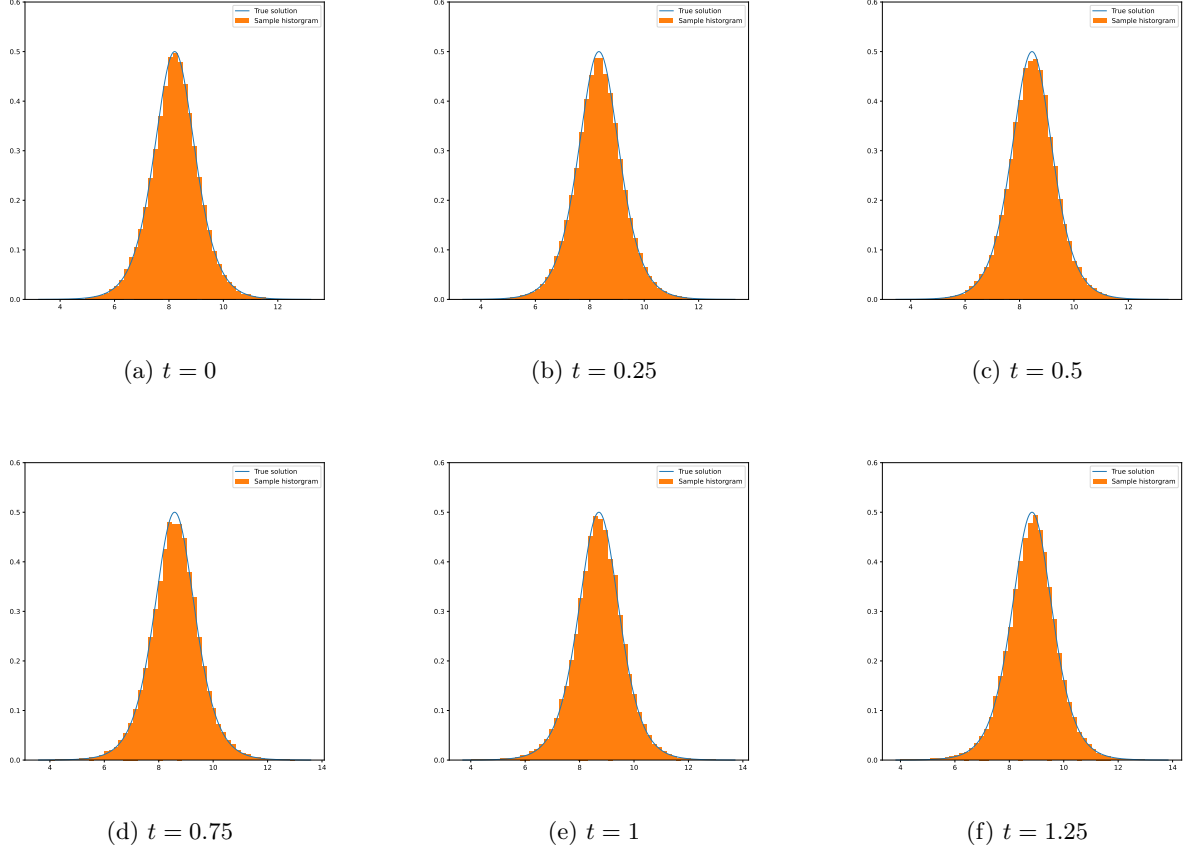


Fig. 2: Sample histogram of computed ρ_θ at different time t for 1D ($d = 1$) GPE.

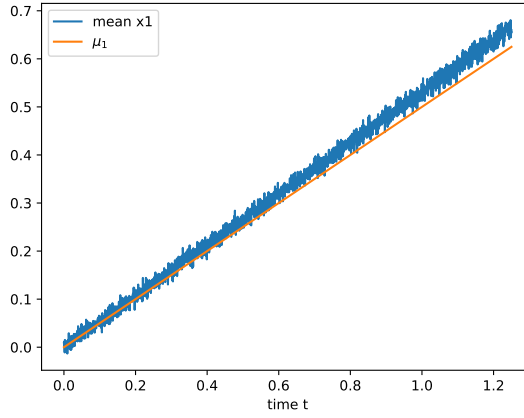


Fig. 3: Center of samples for the GPE with $d = 3$. The orange curve is the analytical values, the blue curve are obtained by samples generated by the push-forward $T_{\theta(t)}$.

7.3. TDSE for system with three interactive-particles. Finally, we present a numerical example of solving a system with three interactive-particles. We consider the three particles in a 3D space, with the

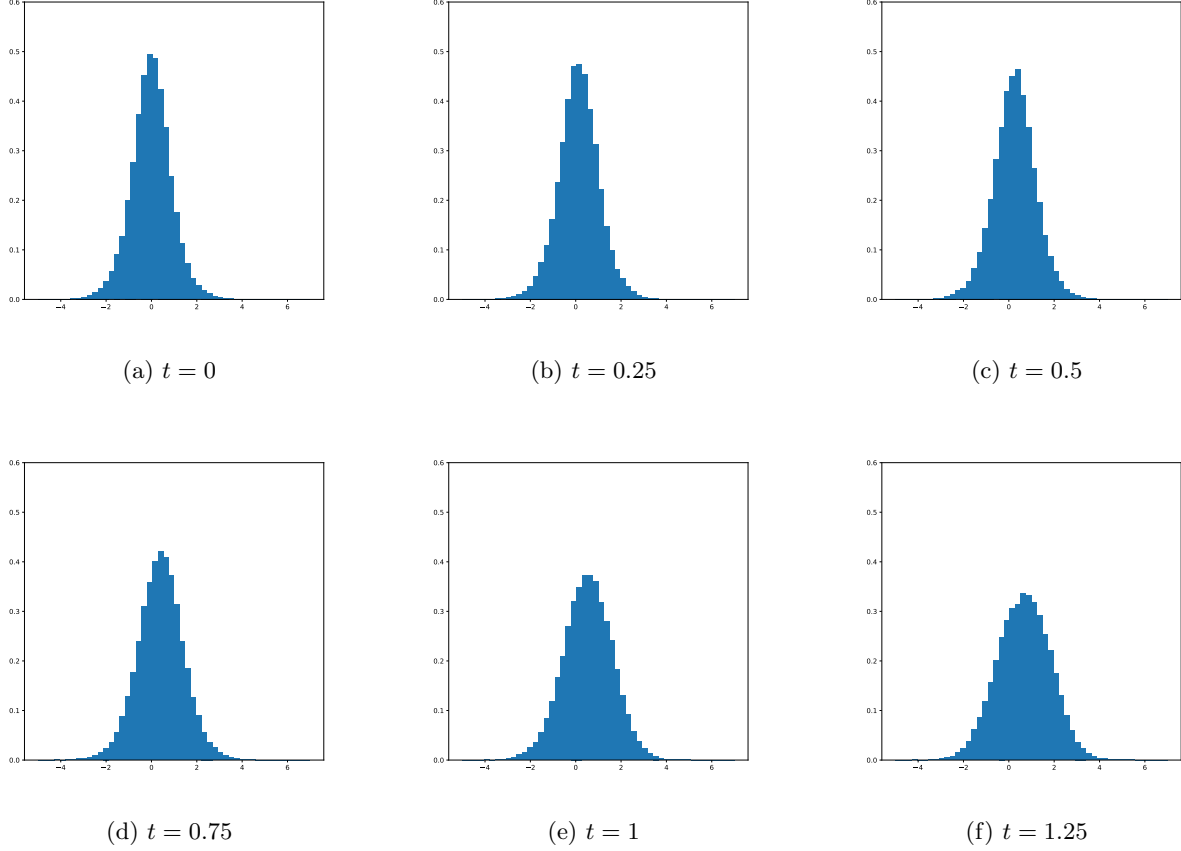


Fig. 4: Sample histogram of computed ρ_θ at different time t for 3D GPE.

origin exerting an attractive force on each of them. We set the potential \mathcal{F}_R to be

$$(7.11) \quad \mathcal{F}_R(\rho) = \iiint \left[-\sum_k \frac{3}{\epsilon + |z_k|} + \sum_{k < j} \frac{1}{\epsilon + |z_k - z_j|} \right] \rho(z_1, z_2, z_3) dz_1 dz_2 dz_3,$$

where $z_k := (x_{3k-2}, x_{3k-1}, x_{3k})^\top \in \mathbb{R}^3$ for $k = 1, 2, 3$, ϵ is a small positive number which is set to 0.005 in our computation, and $|\cdot|$ is the l^2 norm in Euclidean space. The L^2 first variation of \mathcal{F}_R is:

$$(7.12) \quad \frac{\delta}{\delta \rho} \mathcal{F}_R(\rho) = -\sum_k \frac{3}{\epsilon + |z_k|} + \sum_{k < j} \frac{1}{\epsilon + |z_k - z_j|}.$$

The corresponding TDSE in the wavefunction form is:

$$(7.13) \quad i \frac{\partial}{\partial t} \psi(t, x) = -\frac{1}{2} \Delta \psi(t, x) + \left[-\sum_k \frac{3}{\epsilon + |z_k|} + \sum_{k < j} \frac{1}{\epsilon + |z_k - z_j|} \right] \cdot \psi(t, x).$$

We want to generate sample trajectories following the density function $\rho_t(x_1, \dots, x_9)$ corresponding to the solution of (7.13).

In our simulation, the initial density function is taken as a Gaussian in \mathbb{R}^9 :

$$(7.14) \quad \rho_0(z_1, z_2, z_3) = \frac{1}{(\sqrt{2\pi})^9} \exp \left(-\frac{1}{2} \left[|z_1 - c_1|^2 + |z_2 - c_2|^2 + |z_3 - c_3|^2 \right] \right)$$

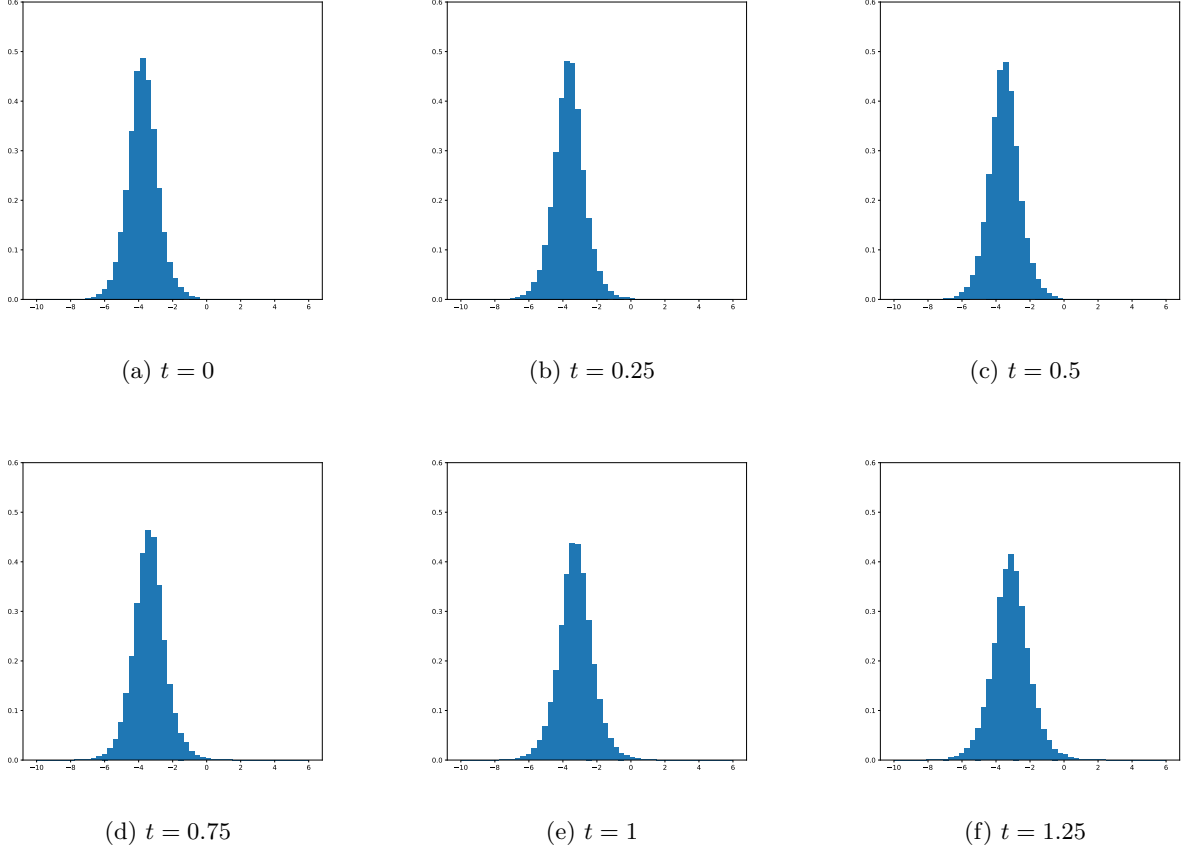


Fig. 5: Sample histogram of computed ρ_θ at different time t for 6D GPE.

where $c_1 = (1, 0, 0)$, $c_2 = (0, 1, 0)$, $c_3 = (0, 0, 1)$. We take the initial Φ_0 as:

$$(7.15) \quad \Phi_0(x) = \frac{x_2^2 - x_3^2}{(x_2^2 + x_3^2)^{0.4}} + \frac{x_6^2 - x_4^2}{(x_4^2 + x_6^2)^{0.4}} + \frac{x_7^2 - x_8^2}{(x_7^2 + x_8^2)^{0.4}}.$$

Again we use the same Neural-ODE architecture for GPE in Section 7.2, consisting of two hidden layers, each containing 50 neurons, with the hyperbolic tangent as the activation function. We generate 12,000 samples for the computation of the operator G while 3,000,000 samples are used to estimate the potential energy \mathcal{F} . These choices are intuitively set according to the computation complexities of estimating G and F , which are quadratic and linear with respect to the number of samples, respectively. The time step size is set as $h = 0.001$. We plot the histogram of x_1 versus time in Figure 6.

In Figure 7, we plot the trajectories of an example triple $(z_1(t), z_2(t), z_3(t))$ where $z_k(t) \in \mathbb{R}^3$ for $k = 1, 2, 3$ and $(z_1(0), z_2(0), z_3(0))$ is randomly sampled from the initial distribution. We also plot in Figure 8 the time evolution of the kinetic energy (orange), interactive potential $\mathcal{F}_R(\rho)$ (blue), and the Hamiltonian (green) in the simulation. The algorithm preserves the Hamiltonian up to time $t = 3$ in this 9D example.

8. Conclusion. We propose a reformulation of TDSE (1.1) in terms of push-forward map inspired by strategies developed for the generative model. This is accomplished by viewing TDSE as a Wasserstein Hamiltonian flow in the probability density manifold via the Madelung transform. Unlike the traditional wavefunction formulation of TDSE or Bohmian mechanics, the new formulation describes the dynamics in the space of diffeomorphisms. A main benefit is its convenience in handling the density evolution and the particle dynamics in a single closed system. Using the properties of Neural-ODE, we derived the corresponding equations in the parameter space of the neural network and designed an algorithm to simulate the solutions numerically. The algorithm provides an alternative to the existing methods. Since it works on samples

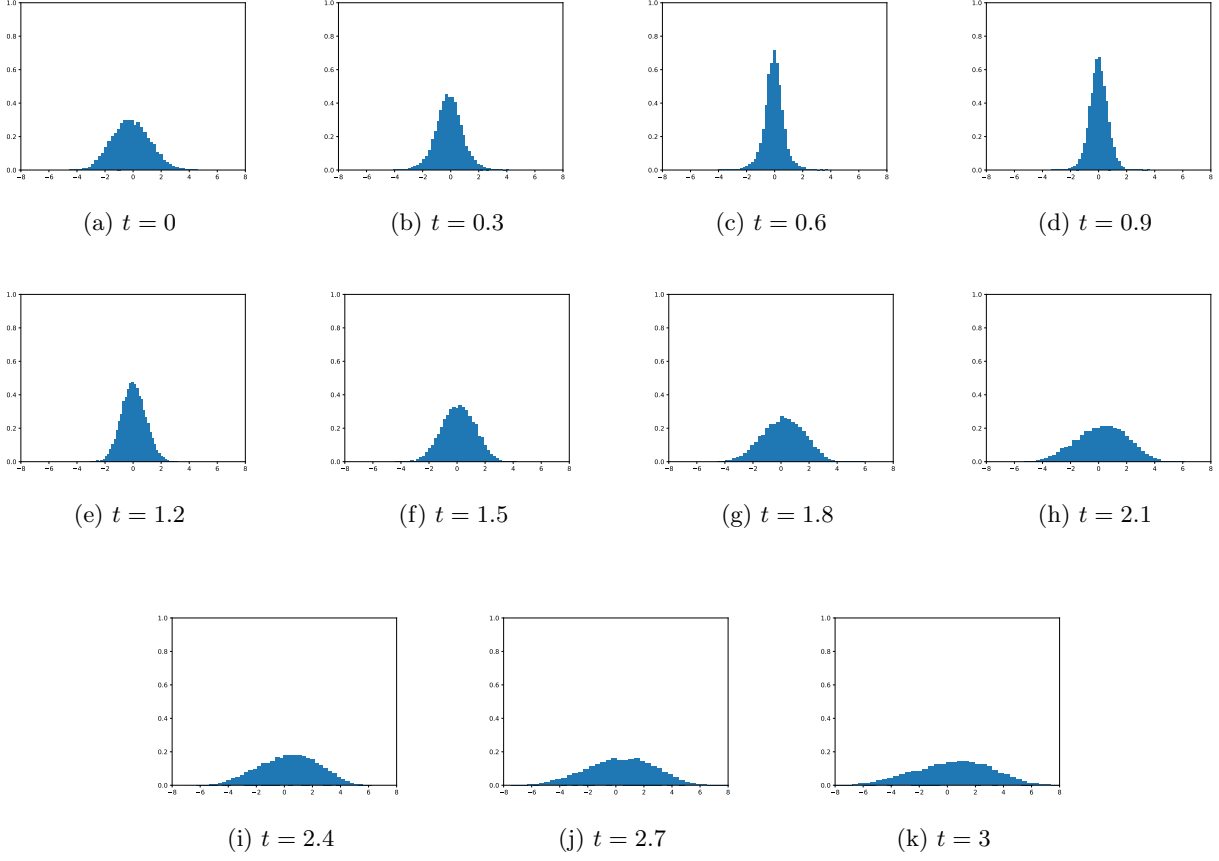


Fig. 6: Sample histogram of computed ρ_θ at different time t for 9D TDSE.

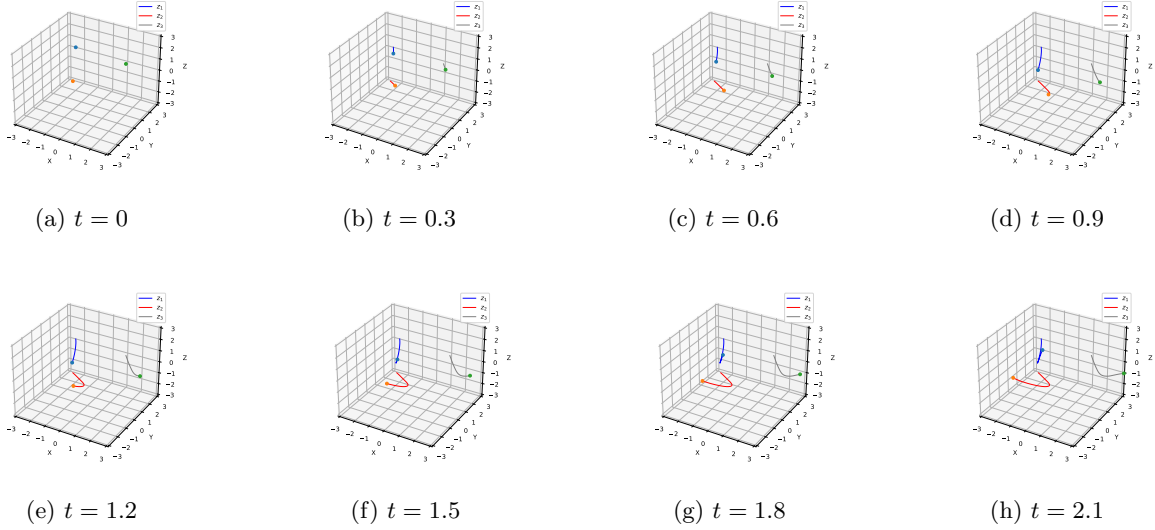


Fig. 7: Trajectories of an example triple $(z_1(t), z_2(t), z_3(t)) \in \mathbb{R}^9$ with $z_k(t) = (x_{3k-2}(t), x_{3k-1}(t), x_{3k}(t)) \in \mathbb{R}^3$ for $k = 1, 2, 3$ at different time slots t for the interactive system experiment which is a 9D TDSE.

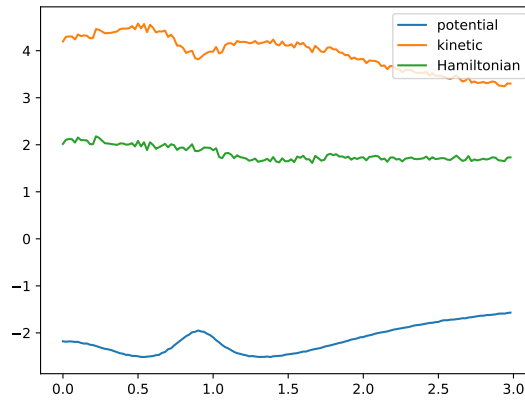


Fig. 8: Evolution of the kinetic energy, interactive potential, and Hamiltonian in the numerical simulation for the 9D interactive system example.

and neural network parameter space, it is computationally friendly to problems in high dimensions. We demonstrated its performance in various problems including a 9D particle system simulation on a desktop. It is also worth mentioning that the simulations are done by the traditional ODE solvers, and there is no data needed to train the neural network.

Meanwhile, the reformulation of the TDSE in the generative model invites a number of interesting questions. For example, how do we theoretically understand and analyze the dynamics in terms of the push forward map? Can we design new neural network structures to capture the dynamics more efficiently? What is the dependence of computational accuracy on the network structure (including the depth, width, and activation functions) and the sample size used in the calculation of G and \mathcal{F} ? Can the error be bounded theoretically in Wasserstein metric? Those are among a long list of questions that are worth exploring in future studies.

9. Acknowledgments. This research is partially supported by NSF grants DMS-1925263, DMS-2152960, DMS-2307465, DMS-2307466 and DMS-2409868. All authors made equal contributions. The authors would like to thank Aleksei Ustimenko for his valuable suggestions to this manuscript.

REFERENCES

- [1] M. H. ANDERSON, J. R. ENSHER, M. R. MATTHEWS, C. E. WIEMAN, AND E. A. CORNELL, *Observation of bose-einstein condensation in a dilute atomic vapor*, Science, 269 (1995), pp. 198–201, <https://doi.org/10.1126/science.269.5221.198>, <https://arxiv.org/abs/https://www.science.org/doi/pdf/10.1126/science.269.5221.198>.
- [2] X. ANTOINE, W. BAO, AND C. BESSE, *Computational methods for the dynamics of the nonlinear schrödinger/gross-pitaevskii equations*, Computer Physics Communications, 184 (2013), pp. 2621–2633.
- [3] M. ARJOVSKY, S. CHINTALA, AND L. BOTTOU, *Wasserstein generative adversarial networks*, in International conference on machine learning, PMLR, 2017, pp. 214–223.
- [4] W. BAO AND Y. CAI, *Mathematical theory and numerical methods for bose-einstein condensation*, Kinetic and Related Models, 6 (2013), pp. 1–135, <https://doi.org/10.3934/krm.2013.6.1>, /article/id/4105f74e-2db0-4c66-a27e-3c5f7e92f67a.
- [5] W. BAO, S. JIN, AND P. A. MARKOWICH, *Numerical study of time-splitting spectral discretizations of nonlinear schrödinger equations in the semiclassical regimes*, SIAM Journal on Scientific Computing, 25 (2003), pp. 27–64, <https://doi.org/10.1137/S1064827501393253>, <https://doi.org/10.1137/S1064827501393253>, <https://arxiv.org/abs/https://doi.org/10.1137/S1064827501393253>.
- [6] K. BERNDL, D. DÜRR, S. GOLDSTEIN, G. PERUZZI, AND N. ZANGHÌ, *On the global existence of bohmian mechanics*, Communications in Mathematical Physics, 173 (1995), pp. 647–673.
- [7] D. BOHM, *A suggested interpretation of the quantum theory in terms of "hidden" variables. i, ii*, Physical review, 85 (1952), pp. 180–193, <https://doi.org/10.1103/PhysRev.85.180>, <https://link.aps.org/doi/10.1103/PhysRev.85.180>.
- [8] G. CARLEO AND M. TROYER, *Solving the quantum many-body problem with artificial neural networks*, Science, 355 (2017), pp. 602–606, <https://doi.org/10.1126/science.aag2302>, <https://www.science.org/doi/abs/10.1126/science.aag2302>, <https://arxiv.org/abs/https://www.science.org/doi/pdf/10.1126/science.aag2302>.
- [9] R. T. CHEN, Y. RUBANOVA, J. BETTENCOURT, AND D. K. DUVENAUD, *Neural ordinary differential equations*, Advances in neural information processing systems, 31 (2018).
- [10] S.-N. CHOW, W. LI, AND H. ZHOU, *Wasserstein hamiltonian flows*, Journal of Differential Equations, 268 (2020), pp. 1205–

1219.

- [11] H. DEKKER, *Classical and quantum mechanics of the damped harmonic oscillator*, Physics Reports, 80 (1981), pp. 1–110.
- [12] P. A. M. DIRAC, *Quantum mechanics of many-electron systems*, Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences, 123 (1929), pp. 714–733.
- [13] D. DÜRR, S. GOLDSTEIN, R. TUMULKA, AND N. ZANGHÌ, *Bohmian mechanics*, in Compendium of quantum physics, Springer, 2009, pp. 47–55.
- [14] W. E AND B. YU, *The deep ritz method: a deep learning-based numerical algorithm for solving variational problems*, Communications in Mathematics and Statistics, 6 (2018), pp. 1–12.
- [15] T. FEVENS AND H. JIANG, *Absorbing boundary conditions for the schrödinger equation*, SIAM Journal on Scientific Computing, 21 (1999), pp. 255–282, <https://doi.org/10.1137/S1064827594277053>, <https://doi.org/10.1137/S1064827594277053>, <https://arxiv.org/abs/https://doi.org/10.1137/S1064827594277053>.
- [16] A. GOLDBERG, H. M. SCHEY, AND J. L. SCHWARTZ, *Computer-Generated Motion Pictures of One-Dimensional Quantum-Mechanical Transmission and Reflection Phenomena*, American Journal of Physics, 35 (1967), pp. 177–186, <https://doi.org/10.1119/1.1973991>, <https://doi.org/10.1119/1.1973991>, https://arxiv.org/abs/https://pubs.aip.org/aapt/ajp/article-pdf/35/3/177/10112535/177_1-online.pdf.
- [17] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. COURVILLE, AND Y. BENGIO, *Generative adversarial nets*, Advances in neural information processing systems, 27 (2014).
- [18] F. GUERRA AND L. M. MORATO, *Quantization of dynamical systems and stochastic control theory*, Physical review D, 27 (1983), p. 1774.
- [19] J. HAN, A. JENTZEN, ET AL., *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Communications in mathematics and statistics, 5 (2017), pp. 349–380.
- [20] J. HAN, L. ZHANG, AND W. E, *Solving many-electron schrödinger equation using deep neural networks*, Journal of Computational Physics, 399 (2019), p. 108929, <https://doi.org/https://doi.org/10.1016/j.jcp.2019.108929>, <https://www.sciencedirect.com/science/article/pii/S0021999119306345>.
- [21] J. HERMANN, Z. SCHÄTZLE, AND F. NOÉ, *Deep-neural-network solution of the electronic schrödinger equation*, Nature Chemistry, 12 (2020), pp. 891–897.
- [22] 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 (2005), pp. 222–241.
- [23] J. D. LAFFERTY, *The density manifold and configuration space quantization*, Transactions of the American Mathematical Society, 305 (1988), pp. 699–741.
- [24] D. LU, H. WANG, M. CHEN, L. LIN, R. CAR, E. WEINAN, W. JIA, AND L. ZHANG, *86 pflops deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy*, Computer Physics Communications, 259 (2021), p. 107624.
- [25] P. MOCZ AND S. SUCCI, *Numerical solution of the nonlinear schrödinger equation using smoothed-particle hydrodynamics*, Phys. Rev. E, 91 (2015), p. 053304, <https://doi.org/10.1103/PhysRevE.91.053304>, <https://link.aps.org/doi/10.1103/PhysRevE.91.053304>.
- [26] E. NELSON, *Derivation of the schrödinger equation from newtonian mechanics*, Physical review, 150 (1966), p. 1079.
- [27] E. ORLOVA, A. USTIMENKO, R. JIANG, P. Y. LU, AND R. WILLETT, *Deep stochastic mechanics*, arXiv preprint arXiv:2305.19685, (2023).
- [28] D. PFAU, J. S. SPENCER, A. G. MATTHEWS, AND W. M. C. FOULKES, *Ab initio solution of the many-electron schrödinger equation with deep neural networks*, Physical review research, 2 (2020), p. 033429.
- [29] L. P. L. P. PITAEVSKII, *Bose-Einstein condensation / Lev Pitaevskii, Sandro Stringari.*, International series of monographs on physics ; 116, Clarendon Press, Oxford ;, 2003.
- [30] O. V. PREZHDO AND C. BROOKSBY, *Quantum backreaction through the bohmian particle*, Phys. Rev. Lett., 86 (2001), pp. 3215–3219, <https://doi.org/10.1103/PhysRevLett.86.3215>, <https://link.aps.org/doi/10.1103/PhysRevLett.86.3215>.
- [31] J. C. PU, J. LI, AND Y. CHEN, *Solving localized wave solutions of the derivative nonlinear schrödinger equation using an improved pinn method*, Nonlinear Dynamics, 105 (2021), pp. 1723 – 1739.
- [32] M. RAISSI, P. PERDIKARIS, AND G. E. KARNIADAKIS, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational physics, 378 (2019), pp. 686–707.
- [33] M. REDDIGER AND B. POIRIER, *Towards a mathematical theory of the madelung equations: Takabayasi’s quantization condition, quantum quasi-irrotationality, weak formulations, and the wallstrom phenomenon*, Journal of Physics A: Mathematical and Theoretical, 56 (2023), p. 193001.
- [34] M. REGINATTO, *Derivation of the equations of nonrelativistic quantum mechanics using the principle of minimum fisher information*, Physical Review A, 58 (1998), p. 1775.
- [35] L. RUTHOTTO, S. J. OSHER, W. LI, L. NURBEKYAN, AND S. W. FUNG, *A machine learning framework for solving high-dimensional mean field game and mean field control problems*, Proceedings of the National Academy of Sciences, 117 (2020), pp. 9183–9193.
- [36] T. SHIBATA, *Absorbing boundary conditions for the finite-difference time-domain calculation of the one-dimensional schrödinger equation*, Phys. Rev. B, 43 (1991), pp. 6760–6763, <https://doi.org/10.1103/PhysRevB.43.6760>, <https://link.aps.org/doi/10.1103/PhysRevB.43.6760>.
- [37] Y. SONG, J. SOHL-DICKSTEIN, D. P. KINGMA, A. KUMAR, S. ERMION, AND B. POOLE, *Score-based generative modeling through stochastic differential equations*, in International Conference on Learning Representations, 2021.
- [38] A. SORIANO, E. A. NAVARRO, J. A. PORTI, AND V. SUCH, *Analysis of the finite difference time domain technique to solve the Schrödinger equation for quantum devices*, Journal of Applied Physics, 95 (2004), pp. 8011–8018, <https://doi.org/10.1063/1.1753661>, <https://doi.org/10.1063/1.1753661>, https://arxiv.org/abs/https://pubs.aip.org/aip/jap/article-pdf/95/12/8011/10634419/8011_1-online.pdf.
- [39] C. VILLANI ET AL., *Optimal transport: old and new*, vol. 338, Springer, 2009.
- [40] M.-K. VON RENESSE, *An optimal transport view of schrödinger’s equation*, Canadian Mathematical Bulletin, 55 (2012), p. 858–869, <https://doi.org/10.4153/CMB-2011-121-9>.

- [41] T. C. WALLSTROM, *Inequivalence between the schrödinger equation and the madelung hydrodynamic equations*, Physical Review A, 49 (1994), p. 1613.
- [42] B. WU, O. HENNIGH, J. KAUTZ, S. CHOUDHRY, AND W. BYEON, *Physics informed rnn-dct networks for time-dependent partial differential equations*, in International Conference on Conceptual Structures, 2022.
- [43] H. WU, S. LIU, X. YE, AND H. ZHOU, *Parameterized wasserstein hamiltonian flow*, SIAM Journal on Numerical Analysis, 63 (2025), pp. 360–395.
- [44] Y. ZANG, G. BAO, X. YE, AND H. ZHOU, *Weak adversarial networks for high-dimensional partial differential equations*, Journal of Computational Physics, 411 (2020), p. 109409.

Appendix A. Evaluation of Fisher information.

We denote $x = T_\theta(z)$ and consider evaluating the parameterized Fisher information:

$$(A.1) \quad F_Q(\theta) = \mathcal{F}_Q(\rho_\theta) = \frac{1}{8} \int_{\mathbb{R}^d} |\nabla_x \log \rho_\theta(x)|^2 \rho_\theta(x) dx = \frac{1}{8} \int_{\mathbb{R}^d} |\nabla_x \log \rho_\theta(T_\theta(z))|^2 \lambda(z) dz.$$

The gradient of log density term needs careful treatment, since $\log \rho_\theta \circ T_\theta(\cdot)$ is a function of z while the gradient is taken with respect to x . The following algorithm is designed to compute (A.1) efficiently with PyTorch:

Algorithm A.1 Compute Fisher information

Define a reference density λ , and a reversible neural network $T_\theta(z)$.
 Generate samples $\{z_1, \dots, z_N\}$ from λ , and compute $x_i = T_\theta(z_i)$.
 Detach $\{x_i\}_{i=1}^N$ from computational graph and set $\{x_i\}_{i=1}^N$ as leaf variable by enabling requires_grad property.
 Construct a function $h(x) = \log \rho_\theta \circ T_\theta \circ T_\theta^{-1}(x)$. $h(x)$ is a function of $x = T_\theta(z)$ with traced gradient computation.
 Take the gradient of $\sum_{i=1}^N h(x_i)$ with respect to $\{x_i\}_{i=1}^N$, which gives $\{\nabla_{x_i} h(x_i) = \nabla_{x_i} \log \rho_\theta(x_i)\}_{i=1}^N$.
 Compute the empirical Fisher information $\hat{F}_Q(\theta) = \frac{1}{8N} \sum_{i=1}^N |\nabla_{x_i} h(x_i)|^2$.
 Compute $\nabla_\theta \hat{F}_Q(\theta)$ using backpropagation.
Output: empirical Fisher information $\hat{F}_Q(\theta)$ and its gradient $\nabla_\theta \hat{F}_Q(\theta)$.

In our experiments, Neural-ODE is used as the push-forward map, since it provides an efficient evaluation of T_θ^{-1} and $\log \rho_\theta \circ T_\theta(\cdot)$. T_θ^{-1} can be constructed simply by reversing the time evolution of the ODE, and $\log \rho_\theta \circ T_\theta(\cdot)$ can be computed using the instant change of variables formula in [9].