Computing the Gross–Pitaevskii Ground State via Wasserstein Gradient Flow in Diffeomorphism Space

Xiangxiong Zhang[a,*], Haomin Zhou[b]

[a]*Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907, USA*
[b]*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Abstract

We compute the ground state u of the Gross–Pitaevskii equation (GPE) via Wasserstein gradient descent in diffeomorphism space. We represent the density $\rho = u^2$ as the push-forward of a fixed reference measure through a parameterized transport map T_θ, realized by a boundary-preserving Neural ODE. The Wasserstein gradient flow on probability densities then lifts to natural gradient descent in the finite-dimensional parameter space, with metric tensor given by the pullback of the Wasserstein metric. The method is entirely mesh-free and preserves the unit-mass constraint without normalization. We present numerical experiments in dimensions $d = 1, 2, 3$ and demonstrate that the parameterized Wasserstein gradient flow (PWGF) output can be used to initialize the H^1 Sobolev gradient flow, reducing the initial energy gap by a factor of 7 in 2D and 4.5 in 3D compared to trivial initial conditions.

Keywords: Gross–Pitaevskii equation, Wasserstein gradient flow, natural gradient descent, Neural ODE, ground state computation
2000 MSC: 35Q55, 49Q22, 65K10, 81Q05

1. Introduction

A standard mathematical model of the equilibrium states in Bose–Einstein condensation (BEC) is to consider the ground state of the Gross–Pitaevskii

*Corresponding author
Email addresses: zhan1966@purdue.edu (Xiangxiong Zhang), hmzhou@math.gatech.edu (Haomin Zhou)

equation (GPE) on a bounded domain $\Omega \subset \mathbb{R}^d$ with homogeneous Dirichlet conditions, which is the minimizer of the energy

$$E(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 + V|u|^2 + \frac{\beta}{2}|u|^4\, dx \tag{1}$$

over the unit sphere $\mathbb{S} = \{u \in H_0^1(\Omega) : \|u\|_{L^2} = 1\}$, with an external trapping potential $V \geq 0$. Equivalently, the ground state satisfies the nonlinear eigenvalue problem

$$-\Delta u + Vu + \beta|u|^2 u = \lambda u, \qquad u\big|_{\partial\Omega} = 0, \qquad \|u\|_{L^2} = 1, \tag{2}$$

where the eigenvalue λ is the Lagrange multiplier enforcing the constraint. Existence, uniqueness, and strict positivity of the ground state for the defocusing regime $\beta \geq 0$ are classical [16, 17, 23].

The recent review by Henning and Jarlebring [14] organizes the extensive literature on (2) into two complementary paradigms. The *algebraic paradigm* [14, Section 4] discretizes first, producing a nonlinear eigenvalue problem $A(\mathbf{v})\mathbf{v} = \lambda\mathbf{v}$ in $\mathbb{R}^n$, then applies nonlinear eigenvalue algorithms such as Self Consistent Field. The *variational paradigm* [14, Section 5] works with the continuous energy, equipping $\mathbb{S}$ with a Sobolev inner product, such as L^2, H^1, or the solution-adapted a_u, to derive gradient flows with progressively stronger convergence guarantees [14, Theorems 5.15, 5.25]. The gradient flow methods can also be regarded as Riemannian optimization methods on unit sphere [10]. Convergence rate is often related to the spectral gap of the differential operator [12]. The two paradigms are connected, e.g., the J-type Newton method coincides with Rayleigh quotient iteration [14, Section 5.6.2]. All methods in both paradigms rely on a spatial discretization of Ω, with representation cost $O(n^d)$ in the number of mesh points and per-step cost dominated by an elliptic solve (for Sobolev flows) or solving linear systems (for algebraic methods).

In this paper we compute GPE ground state by Wasserstein gradient descent in diffeomorphism space, extending the parameterized Wasserstein Hamiltonian flow (PWHF) framework of Liu et al. [19, 18] from the time-dependent Schrödinger equation to the nonlinear eigenvalue problem (2). Since the ground state satisfies $u > 0$, the Madelung transform $\rho = u^2$ rewrites the GP energy as a functional on probability densities $\rho \in \mathcal{P}(\Omega)$:

$$E(\rho) = \underbrace{\tfrac{1}{8}\int_\Omega |\nabla \log\rho|^2 \rho\, dx}_{F_Q(\rho)} + \underbrace{\tfrac{1}{2}\int_\Omega V\rho\, dx}_{F_V(\rho)} + \underbrace{\tfrac{\beta}{4}\int_\Omega \rho^2\, dx}_{F_R(\rho)}, \tag{3}$$

where F_Q is the Fisher information (§2.1). The constraint $\|u\|_{L^2} = 1$ becomes $\int \rho \, dx = 1$, which is automatic for probability densities. The gradient descent with respect to the 2-Wasserstein metric gives

$$\partial_\tau \rho = \nabla \cdot \big(\rho \, \nabla \tfrac{\delta E}{\delta \rho}\big), \tag{4}$$

a continuity equation that preserves mass by construction, unlike the classical methods which usually require explicit renormalization.

We represent the density as the push-forward $\rho_\theta = (T_\theta)_\sharp \mu$ of a fixed reference density μ through a parameterized diffeomorphism $T_\theta \colon \Omega \to \Omega$, realized by a boundary-preserving Neural ODE (§3). One draws N sample points $z_i \sim \mu$ once; at each iteration the particle positions $x_i = T_\theta(z_i)$ and density values $\rho_\theta(x_i) = \mu(z_i)/|\det \nabla T_\theta(z_i)|$ are obtained by evaluating the map (§2.4). The method requires neither a spatial mesh nor a basis expansion, and preserves the unit-mass constraint without normalization.

The Wasserstein metric on $\mathcal{P}(\Omega)$ pulls back to the parameter space $\Theta \subset \mathbb{R}^M$ via the metric tensor (§2.5)

$$\widehat{G}(\theta) = \tfrac{1}{N} \textstyle\sum_{i=1}^N (\partial_\theta T_\theta(z_i))^\top (\partial_\theta T_\theta(z_i)) \in \mathbb{R}^{M \times M},$$

and the Wasserstein gradient descent becomes natural gradient descent [2]:

$$\theta^{k+1} = \theta^k - \alpha \, \widehat{G}(\theta^k)^+ \nabla_\theta E(\theta^k), \tag{5}$$

where $(\cdot)^+$ denotes a regularized pseudoinverse. The parameters θ evolve by a geometric flow on the Wasserstein manifold, discretized in parameter space. The Fisher information F_Q is computed via an augmented ODE that tracks the log-determinant and its spatial derivative analytically, avoiding second-order automatic differentiation (§2.4). Eigenvalues are recovered from $\lambda = 2E + 2F_R$.

This work is inspired by the time-dependent PWHF framework of [19, 18] in which (θ, p) evolves as a Hamiltonian system, and p encodes the quantum phase Φ. For the ground state, there is no Hamiltonian, thus we name (5) as *PWGF* (parameterized Wasserstein gradient flow).

In this paper, we provide a computational realization of PWGF for GPE ground states in $d = 1, 2, 3$. In PWGF, there is an approximation error from using neural network parameters θ to parametrize the diffeomorphism T_θ. The PWGF (5) also contains an ill conditioned matrix $\widehat{G}$, which often causes difficulty for high precision computation. We benchmark against exact

solutions or the finite difference references of [23], and propose a hybrid strategy in which PWGF output initializes the H^1 Sobolev gradient flow, reducing the initial energy gap by a factor of 4.5–7× compared to trivial initializations.

For simplicity, we only consider the defocusing case $\beta \geq 0$ on bounded domains with Dirichlet conditions. Convergence analysis of the parameterized Wasserstein gradient flow remains open, and the numerical results presented here are empirical. Section 2 derives the mathematical framework, Section 3 describes the algorithm, and Section 4 presents numerical experiments and the warm-start comparison, and concluding remarks are given in Section 5.

2. Mathematical Framework

This section derives the parameterized Wasserstein gradient flow in general dimension d. We reformulate the GP energy in density form (§2.1), compute first variations (§2.2), derive the Wasserstein gradient flow (§2.3), lift it to diffeomorphism space (§2.4), and reduce to finite dimensions by Neural ODE parameterization (§2.5).

2.1. Madelung transform: from eigenfunction to density

The ground state u is strictly positive in Ω for the following reasons. The energy satisfies $E(u) \geq E(|u|)$, so the minimizer can be taken nonnegative. A nonnegative minimizer satisfies (2) with the self-adjoint operator $A_{u^*} = -\Delta + V + \beta(u^*)^2$, whose Green's function is strictly positive by the strong maximum principle. The Krein–Rutman theorem then implies $u^* > 0$ on Ω; see [16, 17, 23] and, at the discrete level, [23, Theorem 3.4].

Because $u > 0$, we may write

$$u = \sqrt{\rho}, \qquad \rho = u^2 > 0. \tag{6}$$

The kinetic energy $\frac{1}{2}\int |\nabla u|^2\, dx$ transforms into the Fisher information of ρ.

Kinetic energy as Fisher information. By writing $u = \sqrt{\rho}$ and applying the chain rule, we get

$$\frac{1}{2}\int_\Omega |\nabla u|^2\, dx = \frac{1}{2}\int_\Omega |\nabla \sqrt{\rho}|^2\, dx = \frac{1}{8}\int_\Omega |\nabla \log \rho|^2\, \rho\, dx, \tag{7}$$

where $F_Q(\rho) := \frac{1}{8}\int_\Omega |\nabla \log \rho|^2\, \rho\, dx$ is the Fisher information.

By substituting (7) and using $|u|^2 = \rho$, $|u|^4 = \rho^2$, we get (3), where F_Q is the Fisher information, F_V the potential energy, and F_R the interaction energy. The density formulation (3) also appears in Bao and Ruan [4], who observe that $|\nabla\rho|^2/\rho$ in F_Q becomes singular as $\rho \to 0^+$, a difficulty for any grid-based method that evaluates ρ at fixed spatial nodes near $\partial\Omega$.

Remark 1 (Regularization in grid-based density methods)**.** *Bao and Ruan [4] address the singularity in F_Q by ε-regularization: replace ρ by $\rho + \varepsilon$ in the denominator, giving the regularized energy*

$$E^\varepsilon(\rho) = \int_\Omega \frac{|\nabla\rho|^2}{8(\rho+\varepsilon)} + \frac{1}{2}V\rho + \frac{\beta}{4}\rho^2\,dx.$$

They prove that E^ε is convex for $\beta \geq 0$, that the minimizer $\rho_g^\varepsilon \to \rho_g$ as $\varepsilon \to 0^+$. The convergence rate $\|\rho_g - \rho_g^\varepsilon\|_{L^2} = O(\sqrt{\varepsilon})$ was expected from their analysis, though the asymptotic error $O(\varepsilon)$ can be observed numerically in some setup.

Constraint transformation. The L^2-normalization $\|u\|_{L^2} = 1$ becomes $\int_\Omega \rho\,dx = 1$: the density is automatically a probability measure. The Dirichlet boundary condition $u|_{\partial\Omega} = 0$ becomes $\rho|_{\partial\Omega} = 0$.

2.2. First variations and eigenvalue recovery

We compute the first variation of each term in (3) with respect to ρ, subject to $\int \rho\,dx = 1$.

First variation of F_Q. The first variation is

$$\frac{\delta F_Q}{\delta\rho} = -\frac{1}{2}\frac{\Delta\sqrt{\rho}}{\sqrt{\rho}}. \tag{8}$$

To verify, write $F_Q(\rho) = \frac{1}{2}\int |\nabla\sqrt{\rho}|^2\,dx$ and take the Gâteaux derivative: for a perturbation σ with $\int \sigma\,dx = 0$,

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} F_Q(\rho+\varepsilon\sigma) = \frac{1}{2}\int_\Omega \nabla\sqrt{\rho}\cdot\nabla\left(\frac{\sigma}{2\sqrt{\rho}}\right)\,dx = -\frac{1}{2}\int_\Omega \frac{\Delta\sqrt{\rho}}{\sqrt{\rho}}\cdot\frac{\sigma}{2}\,dx + (\text{b.t.})$$

after integrating by parts (boundary terms vanish since $\rho|_{\partial\Omega} = 0$). Reading off the coefficient of $\sigma/2$ gives (8). The quantity $-\frac{1}{2}\Delta\sqrt{\rho}/\sqrt{\rho}$ is the *quantum potential* (Bohm potential), connecting the Madelung and de Broglie–Bohm pictures.

First variations of F_V and F_R. The remaining variations are immediate:

$$\frac{\delta F_V}{\delta \rho} = \frac{1}{2}V, \qquad \frac{\delta F_R}{\delta \rho} = \frac{\beta}{2}\rho. \tag{9}$$

Euler–Lagrange equation. At a minimizer of $E(\rho)$ subject to $\int \rho\, dx = 1$, the first-order optimality condition with Lagrange multiplier λ reads

$$\frac{\delta E}{\delta \rho} = -\frac{1}{2}\frac{\Delta\sqrt{\rho}}{\sqrt{\rho}} + \frac{1}{2}V + \frac{\beta}{2}\rho = \frac{\lambda}{2}. \tag{10}$$

Multiplying both sides by $2\sqrt{\rho}$ and writing $u = \sqrt{\rho}$ recovers the GPE (2).

Eigenvalue formula. Multiplying the GPE (2) by u and integrating by parts,

$$\lambda = \int_\Omega \left(|\nabla u|^2 + Vu^2 + \beta u^4\right) dx = 2F_Q + 2F_V + 4F_R. \tag{11}$$

Since $E = F_Q + F_V + F_R$, we obtain

$$\lambda = 2E + 2F_R. \tag{12}$$

This is the eigenvalue estimator used in all our numerical experiments.

2.3. The Wasserstein gradient flow

Having reformulated the ground state as the minimizer of $E(\rho)$ over probability densities, we now choose a *metric* on the space of densities to define the gradient descent. Different metrics for u give different gradient flows, e.g., the L^2 metric leads to the normalized gradient flow of Bao and Du [5] and a Sobolev metric leads to the Sobolev gradient flow [13]. Here we use the 2-Wasserstein metric from optimal transport for ρ.

The 2-Wasserstein distance. For two probability densities $\rho_0, \rho_1 \in \mathcal{P}(\Omega)$, the 2-Wasserstein distance is

$$W_2(\rho_0, \rho_1)^2 = \inf_\pi \int_{\Omega\times\Omega} |x - y|^2 \, d\pi(x, y), \tag{13}$$

where the infimum is over all couplings π with marginals ρ_0 and ρ_1 [22, 3]. The Wasserstein distance metrizes $\mathcal{P}(\Omega)$ with a geometry tied to the *cost of transporting mass*, rather than to pointwise comparison of densities.

Wasserstein metric tensor (Otto calculus). Following Otto [20], the tangent space at $\rho \in \mathcal{P}(\Omega)$ consists of perturbations σ satisfying $\int \sigma \, dx = 0$. Each such σ is associated with a velocity potential Φ via the continuity equation $\sigma = -\nabla \cdot (\rho \nabla \Phi)$. The Wasserstein Riemannian metric is

$$g_\rho^W(\sigma_1, \sigma_2) = \int_\Omega \nabla \Phi_1 \cdot \nabla \Phi_2 \; \rho \, dx, \qquad \text{where } -\nabla \cdot (\rho \nabla \Phi_i) = \sigma_i. \tag{14}$$

This is an *inner product on tangent directions* that weights perturbations by how much mass must be moved.

Wasserstein gradient. The Wasserstein gradient $\operatorname{grad}_W E(\rho)$ is the unique tangent vector satisfying $g_\rho^W(\operatorname{grad}_W E, \, \sigma) = dE(\rho)[\sigma]$ for all admissible σ. Using (14) and integrating by parts, one finds

$$\operatorname{grad}_W E(\rho) = -\nabla \cdot \left(\rho \, \nabla \frac{\delta E}{\delta \rho} \right). \tag{15}$$

The gradient flow. Steepest descent $\partial_\tau \rho = -\operatorname{grad}_W E(\rho)$ gives

$$\partial_\tau \rho = \nabla \cdot \left(\rho \, \nabla \frac{\delta E}{\delta \rho} \right). \tag{16}$$

This PDE preserves mass ($\int \rho \, dx = 1$) automatically because the right-hand side is a divergence. No normalization step is needed.

The stationary point of (16) satisfies $\nabla(\delta E/\delta \rho) = 0$, i.e., $\delta E/\delta \rho = \text{const} = \lambda/2$, recovering the Euler–Lagrange equation (10) and hence the GPE.

Relation to Sobolev gradient flows. In the H^1-gradient flow [23], the gradient step is $\partial_\tau u = -(-\Delta + a)^{-1}(\delta E/\delta u)$, requiring an elliptic solve per step and explicit normalization. The Wasserstein flow (16) operates on ρ rather than u, preserves mass by construction, and its per-step cost depends on the parameterization (see §2.5 below) rather than on a mesh-based elliptic solver.

No momentum for the ground state. For the time-dependent Schrödinger equation, the Madelung system involves both ρ and a phase Φ, leading to a Wasserstein *Hamiltonian* flow [19, 18]. For the ground state, $\Phi \equiv 0$ (no phase), and no momentum variable p is needed. This is one particular difference compared to the time-dependent PWHF framework.

2.4. Lifting to diffeomorphism space

Following [19, 18], we lift the gradient flow (16) from $\mathcal{P}(\Omega)$ to the space of diffeomorphisms $\mathcal{O} = \{T : \Omega \to \Omega : T \text{ is a diffeomorphism}\}$ by representing each density as the push-forward of a fixed reference density through a transport map T. The Wasserstein metric pulls back to the $L^2(\mu)$-metric on $\mathcal{O}$ (see (18)), and the gradient flow becomes a pointwise ODE (19) whose trajectory is fully determined by its initial condition. All integrals over ρ become sample averages over fixed reference particles $z_i \sim \mu$ [18].

Push-forward map. Let μ be a fixed *reference density* on Ω (chosen to satisfy $\mu|_{\partial\Omega} = 0$, with the same boundary vanishing as ρ). An orientation-preserving diffeomorphism $T : \Omega \to \Omega$ pushes μ forward to

$$\rho = T_\sharp \mu, \qquad \rho(T(z)) = \frac{\mu(z)}{|\det \nabla T(z)|} \quad \text{for } z \in \Omega. \tag{17}$$

By the change of variables formula, $\int_\Omega \rho\, dx = \int_\Omega \mu\, dz = 1$, so the push-forward preserves the probability constraint automatically. Expectations under ρ become expectations under μ:

$$\int_\Omega \varphi(x)\, \rho(x)\, dx = \int_\Omega \varphi(T(z))\, \mu(z)\, dz \approx \frac{1}{N} \sum_{i=1}^{N} \varphi(T(z_i)),$$

where $z_1, \ldots, z_N \sim \mu$ are fixed quadrature particles.

Metric on diffeomorphism space. The Wasserstein metric (14) on $\mathcal{P}(\Omega)$ induces a metric on the space $\mathcal{O} = \{T : \Omega \to \Omega \text{ diffeo.}\}$. For two tangent vectors $\sigma_1, \sigma_2 \in T_T\mathcal{O}$ (vector fields on Ω), the induced metric is [19]

$$\mathfrak{g}_T(\sigma_1, \sigma_2) = \int_\Omega \sigma_1(z)^\top \sigma_2(z)\ \mu(z)\, dz. \tag{18}$$

This is simply the $L^2(\mu)$-inner product on vector fields. This pullback is exact [19].

Gradient flow on $\mathcal{O}$. The Wasserstein gradient descent (16) pulls back to

$$\partial_\tau T(z) = -\nabla_x \frac{\delta E}{\delta \rho}\big(T_\sharp \mu, \cdot\big) \circ T(z). \tag{19}$$

This is an ODE on the space of diffeomorphisms, driven by the gradient of $\delta E/\delta\rho$ evaluated along the current map.

2.5. Finite-dimensional reduction via Neural ODE

We restrict T to a finite-dimensional family $\mathcal{O}_\Theta = \{T_\theta : \theta \in \Theta \subset \mathbb{R}^M\}$, where T_θ is a Neural ODE (details in Section 3.1).

Pullback metric tensor. A tangent vector $\dot{\theta} \in \mathbb{R}^M$ in parameter space maps to a tangent vector $\sigma = (\partial_\theta T_\theta)\dot{\theta}$ in diffeomorphism space. Substituting into the metric (18):

$$\begin{aligned}\mathfrak{g}(\dot{\theta}_1, \dot{\theta}_2) &= \int_\Omega \left[(\partial_\theta T_\theta)\dot{\theta}_1\right]^\top \left[(\partial_\theta T_\theta)\dot{\theta}_2\right] \, \mu \, dz \\ &= \dot{\theta}_1^\top \underbrace{\left[\int_\Omega (\partial_\theta T_\theta(z))^\top (\partial_\theta T_\theta(z)) \, \mu(z) \, dz\right]}_{G(\theta)} \dot{\theta}_2.\end{aligned}$$

Here $\partial_\theta T_\theta(z) \in \mathbb{R}^{d \times M}$ is the Jacobian of $T_\theta(z)$ with respect to θ, so $G(\theta) \in \mathbb{R}^{M \times M}$ is the *pullback metric tensor*. Approximating the μ-integral by Monte Carlo with the fixed particles $z_1, \ldots, z_N \sim \mu$:

$$\widehat{G}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \big(\partial_\theta T_\theta(z_i)\big)^\top \big(\partial_\theta T_\theta(z_i)\big) \; \in \; \mathbb{R}^{M \times M}. \tag{20}$$

Natural gradient descent. Restricting the gradient flow (19) to $\mathcal{O}_\Theta$ and using the pullback metric, the Wasserstein gradient descent in parameter space becomes

$$\theta^{k+1} = \theta^k - \alpha \, \widehat{G}(\theta^k)^+ \, \nabla_\theta E(\theta^k), \tag{21}$$

where $(\cdot)^+$ denotes a regularized pseudoinverse and $\alpha > 0$ is the step size. This is *natural gradient descent* [2]: the matrix $\widehat{G}(\theta)^+$ rotates and rescales the Euclidean gradient $\nabla_\theta E$ to account for the non-Euclidean geometry of the parameter-to-density map $\theta \mapsto \rho_\theta$.

Remark 2 (No regularization needed in PWGF)**.** *Unlike grid-based density methods [4], the push-forward parameterization avoids the singularity in F_Q without any ε-regularization. In PWGF, the density ρ_θ is never evaluated at arbitrary grid points; it is evaluated only at the pushed particle locations $x_i = T_\theta(z_i)$, where*

$$\rho_\theta(x_i) = \frac{\mu(z_i)}{|\det \nabla T_\theta(z_i)|}.$$

This ratio is strictly positive because: (i) *the particles z_i are sampled from the interior of Ω, where the reference density satisfies $\mu(z_i) > 0$; and* (ii) *T_θ is an orientation-preserving diffeomorphism, so $\det \nabla T_\theta(z_i) > 0$. The score $\partial_x \log \rho_\theta$ is computed via the augmented ODE (see Section 3.1), not by numerical differentiation of ρ, eliminating the $|\nabla\rho|^2/\rho$ singularity at the discrete level. The only requirements are that μ vanish at $\partial\Omega$ and that sample points avoid the boundary; both hold when z_i are drawn from a Beta distribution on the open interval.*

2.6. Summary of PWGF

We summarize the PWGF approach and compare it with two families of gradient-flow methods for the GPE ground state.

Conceptual steps of PWGF. For the GPE ground state eigenvalue problem (2):

(i) *Madelung transform.* Use $u > 0$ to write $\rho = u^2$, converting the energy $E(u)$ into a functional $E(\rho) = F_Q(\rho) + F_V(\rho) + F_R(\rho)$ on probability densities (§2.1).

(ii) *Wasserstein gradient.* Choose the 2-Wasserstein metric on $\mathcal{P}(\Omega)$ and derive the gradient flow $\partial_\tau \rho = \nabla \cdot (\rho \nabla(\delta E/\delta\rho))$ (§2.3).

(iii) *Lift to diffeomorphism space.* Represent $\rho = T_\sharp \mu$ and pull back the gradient flow to an ODE on the space of maps: $\partial_\tau T = -\nabla_x(\delta E/\delta\rho) \circ T$ (§2.4).

(iv) *Parameterize and reduce.* Restrict T to a Neural ODE family T_θ, yielding natural gradient descent $\theta^{k+1} = \theta^k - \alpha\, \widehat{G}^{+} \nabla_\theta E$ in $\mathbb{R}^M$ (§2.5).

Steps (iii)–(iv) are the distinctive features of PWGF: the lifting to diffeomorphism space replaces the density-space PDE with a sampling-based ODE, and the Neural ODE parameterization reduces the problem to finite-dimensional natural gradient descent without spatial discretization.

Sobolev gradient flows.. The Sobolev gradient flow framework [13, 23] works directly with the eigenfunction u rather than the density ρ. Given a Hilbert space X with inner product $(\cdot,\cdot)_X$, the X-Sobolev gradient $\nabla_X E(u)$ is defined as the Riesz representative of $E'(u)$: $(\nabla_X E(u),\, w)_X = \langle E'(u),\, w\rangle$ for all $w \in X$. The gradient flow on the unit sphere $\mathbb{S}$ is then $\partial_\tau u = -P_{u,X}\nabla_X E(u)$, where $P_{u,X}$ projects onto the tangent space of $\mathbb{S}$ at u. Different choices of X give different methods:

- L^2 *gradient with* $(w,z)_X = (w,z)$ $(X = L^2)$: $\nabla_{L^2} E(u) = -\Delta u + Vu + \beta u^3$, leading to the GFDN scheme(semi-implicit gradient flow) [5];
- H^1 *gradient* $(X = H_0^1$ with $(w,z)_X = (\nabla w, \nabla z) + (w,z))$: computing $(-\Delta + I)^{-1}$ per step, giving the H^1-flow [23];
- a_0 *gradient* $(X = H_0^1$ with $(w,z)_X = (\nabla w, \nabla z) + (Vw,z))$: computing $(-\Delta + V)^{-1}$ per step, admits adaptive step sizes [14];
- a_u *gradient* $(X = H_0^1$ with $(w,z)_X = (\nabla w, \nabla z) + (Vw,z) + \beta(u^2 w, z))$: $\nabla_{a_u} E(u) = u$ trivially, reducing the flow to damped inverse iteration [14].

All of these are mesh-based: they require a spatial discretization (finite element or finite difference) and solve one or two elliptic PDEs per iteration. They enjoy rigorous convergence guarantees such as global energy decrease and local exponential convergence with rate governed by the spectral gap of the differential operator involved. See [14] and references therein.

By contrast, PWGF is mesh-free: it uses N fixed particles and a Neural ODE with M parameters, and computes the natural gradient direction by solving an $M \times M$ linear system rather than an elliptic PDE. The Wasserstein metric automatically preserves the mass constraint $\int \rho\, dx = 1$, so no normalization step is needed, whereas all Sobolev flows require explicit projection back to $\mathbb{S}$.

Direct Wasserstein discretizations. The Wasserstein gradient flow (16) on $\mathcal{P}(\Omega)$ is the mathematical starting point for PWGF, but discretizing it directly (e.g., via the JKO scheme [15]) requires a spatial mesh and suffers from the curse of dimensionality. The PWGF approach differs from such direct discretizations in two ways:

(a) *Lifting.* Rather than discretizing the density-space PDE (16), PWGF lifts it to the space of diffeomorphisms $\mathcal{O}$, where the flow becomes a pointwise ODE (19). This ODE is self-contained: the trajectory of T is fully determined by its initial condition, without solving the continuity equation separately.

(b) *Parameterization.* The Neural ODE T_θ reduces the infinite-dimensional ODE on $\mathcal{O}$ to a finite-dimensional system in $\mathbb{R}^M$. The pullback of the Wasserstein metric gives the metric tensor $\widehat{G}$, and the flow becomes natural gradient descent (21).

The price is the difficulty of rigorous convergence guarantees. In PWGF, the energy is not guaranteed to decrease monotonically since the finite dimensional parameterization introduces approximation error and the linear solver is only an approximation. A rigorous convergence proof of PWGF is widely open.

Related work. The parameterized Wasserstein gradient flow framework was introduced in [24] for classical Wasserstein gradient flows (Fokker–Planck, porous medium, and aggregation equations), and its Hamiltonian counterpart was developed in [19] for the time-dependent Schrödinger equation. Neither paper addresses the GPE *ground state* (nonlinear eigenvalue) problem, which is the focus of the present work.

To the best of our knowledge, no prior work applies Wasserstein gradient flow to the GPE eigenvalue problem. The L^2 normalized gradient flow for the GPE ground state has been analyzed extensively (see [5] and the recent work [27]); Sobolev gradient flows (H^1, a_0, a_u metrics) are treated in [13, 12, 23, 14]. All of these operate in function space and require spatial discretization. In a different direction, Bao et al. [26] minimize the GP energy via a normalized deep neural network with standard training; their method is not based on Wasserstein geometry or push-forward maps. The push-forward parameterization $\rho_\theta = (T_\theta)_\sharp \mu$ is closely related to normalizing flows [28, 29]; the key difference is that PWGF evolves parameters by a geometric ODE rather than by maximum-likelihood training. Neklyudov et al. [25] use a Wasserstein gradient flow on the Born distribution space to accelerate quantum variational Monte Carlo for fermionic systems, but work directly in the space of distributions (without lifting to diffeomorphism space) and target the many-body Schrödinger equation rather than the mean-field GPE. Peterseim et al. [30] use neural networks to accelerate conventional iterative

solvers for nonlinear Schrödinger eigenvalue problems, leveraging knowledge from prior simulations to predict improved solution trajectories.

3. Algorithm

Implementing the natural gradient descent (21) requires three ingredients: a parameterization T_θ that evaluates the density ρ_θ and score $\nabla_x \log \rho_\theta$ (§3.1); a discretization of $E(\theta)$ and its gradient (§3.2); and a solver for the $M \times M$ system $\widehat{G}\xi = \nabla_\theta E$ (§3.3). The complete algorithm is stated in §3.4; dimension-specific choices are in §3.5.

3.1. Boundary-preserving Neural ODE

On $(-L, L)^d$ with Dirichlet conditions, the boundary constraint $\rho_\theta|_{\partial\Omega} = 0$ requires $T_\theta(\pm L) = \pm L$. We enforce this by defining the Neural ODE velocity field (per coordinate) as

$$f_\theta(w) = \left(1 - w^2/L^2\right) g_\theta(w), \tag{22}$$

where $g_\theta : \mathbb{R} \to \mathbb{R}$ is a tanh network $1 \to H \to H \to 1$ with $M_1 = H^2 + 4H + 1$ parameters. The factor $(1 - w^2/L^2)$ vanishes at $w = \pm L$, so the ODE $\dot{w} = f_\theta(w)$ has fixed points at $\pm L$: starting from any $w(0) = z \in (-L, L)$, the trajectory remains in $(-L, L)$ and $T_\theta(\pm L) = \pm L$ for all θ.

Spatial derivatives. The energy computation requires the first and second derivatives of f_θ with respect to the spatial variable w:

$$f'_\theta(w) = -\frac{2w}{L^2}\, g_\theta(w) + \left(1 - w^2/L^2\right) g'_\theta(w), \tag{23}$$

$$f''_\theta(w) = -\frac{2}{L^2}\, g_\theta(w) - \frac{4w}{L^2}\, g'_\theta(w) + \left(1 - w^2/L^2\right) g''_\theta(w). \tag{24}$$

Here g'_θ and g''_θ are computed analytically via the chain rule through the tanh layers.

Augmented ODE. Following [18], we evolve an augmented state $(w,\, \ell,\, J,\, \ell')$ with dynamics

$$\begin{aligned}
\dot{w} &= f_\theta(w), \\
\dot{\ell} &= f'_\theta(w), \\
J &= \exp(\ell) \quad \text{(updated after each } \ell \text{ step)}, \\
\dot{\ell'} &= f''_\theta(w) \cdot J,
\end{aligned}$$

where $\ell = \log|T'_\theta(z)|$ is the log-Jacobian determinant, $J = |T'_\theta(z)| = e^\ell$ is the Jacobian determinant itself, and $\ell' = \partial_z \log|T'_\theta(z)|$. All four components are evolved per coordinate for each of the N particles, using forward Euler with N_{ODE} substeps.

Score function. Through the ODE, the augmented state avoids second-order differentiation, which causes severe numerical instabilities [18]. The score at $x_k = T_{\theta_k}(z_k)$ for coordinate k is

$$\partial_{x_k} \log \rho_\theta = \frac{\partial_{z_k} \log \mu_k(z_k) - \ell'_k}{T'_{\theta_k}(z_k)}, \tag{25}$$

where $\partial_{z_k} \log \mu_k$ is the score of the reference density (Section 3.5).

3.2. Energy and its gradient

Given N particles $z_i \sim \mu$ and the augmented ODE outputs at $x_i = T_\theta(z_i)$, the energy terms (3) become sample averages via the change of variables $\int h(x)\rho\,dx = \int h(T(z))\mu\,dz$:

$$F_Q(\theta) \approx \frac{1}{8N} \sum_{i=1}^{N} \sum_{k=1}^{d} s_{k,i}^2, \quad s_{k,i} = \frac{\partial_{z_k} \log \mu_k(z_{k,i}) - \ell'_{k,i}}{J_{k,i}}, \tag{26}$$

$$F_V(\theta) \approx \frac{1}{2N} \sum_{i=1}^{N} V(x_i), \tag{27}$$

$$F_R(\theta) \approx \frac{\beta}{4N} \sum_{i=1}^{N} \rho_\theta(x_i), \quad \rho_\theta(x_i) = e^{\log \mu(z_i) - \ell_i}, \tag{28}$$

where for the product-map ansatz (Section 3.5) the total log-Jacobian determinant factorizes additively as $\ell_i = \sum_{k=1}^{d} \ell_{k,i}$, and similarly $\log \mu(z_i) = \sum_k \log \mu_k(z_{k,i})$.

The eigenvalue is estimated from (12): $\lambda = 2E + 2F_R$.

Gradient computation. The gradient $\nabla_\theta E$ is obtained by automatic differentiation through the forward ODE: all N particles are integrated in a single batched pass, and the computational graph is differentiated in the standard reverse mode. The metric tensor $\widehat{G}$, on the other hand, requires the per-sample Jacobians $\partial_\theta T_\theta(z_i) \in \mathbb{R}^{d \times M}$, which are computed via a separate forward pass using functional differentiation (reverse-mode Jacobian computation vectorized over the N samples), cf. (20).

3.3. Linear solver for the natural gradient

The natural gradient direction ξ solves $\widehat{G}(\theta)\,\xi = \nabla_\theta E(\theta)$, where $\widehat{G}$ is symmetric positive semi-definite. However, $\widehat{G}$ is highly ill conditioned with an initial condition number around $O(10^{10})$ and $\widehat{G}$ is often rank deficient in our experiments.

We use conjugate gradients (CG) with Tikhonov regularization $(\widehat{G} + \epsilon I)\xi = \nabla_\theta E$, $\epsilon = 10^{-6}$, run for up to n_{CG} iterations (100 for 1D, 200 for 2D, 300 for 3D). The regularization $\epsilon = 10^{-6}$ is chosen to ensure the CG iteration converges in a controlled number of steps. After computing the CG solution ξ, a backtracking safety check evaluates $E(\theta - \alpha\xi)$ via a trial forward pass; if $E_{\mathrm{new}} > E + E_{\mathrm{tol}}$, the step is rejected in favor of a small normalized gradient descent step. Throughout the iteration, the parameters achieving the lowest energy (*best parameters*) are recorded for use in the final reconstruction.

All steps are followed by norm clipping: $\xi \leftarrow \min(1,\, C/\|\xi\|)\,\xi$. Table 1 lists the solver parameters for each problem.

3.4. The complete algorithm

The complete PWGF iteration is as follows.

1. **Initialize.** Draw $z_1, \ldots, z_N \sim \mu$ (fixed throughout) using *sign-symmetric sampling*: draw $N/2$ particles from $\mu|_{(0,L)}$ and mirror as $z_{N/2+j} = -z_j$. This enforces $T_\theta(-z) = -T_\theta(z)$ (bias gradients cancel by symmetry, keeping g_θ odd), so ρ_θ is symmetric about $x = 0$. Initialize θ^0 with small random weights (zero biases).

2. **For** $k = 0, 1, \ldots, K-1$**:**

   (a) Integrate the augmented Neural ODE (Section 3.1) for all N particles with forward Euler (N_{ODE} substeps) to obtain $\{(x_i,\, \ell_i,\, J_i,\, \ell_i')\}_{i=1}^N$.

   (b) Evaluate $E(\theta^k) = F_Q + F_V + F_R$ by particle quadrature (Section 3.2) and compute its gradient $\nabla_\theta E$ by automatic differentiation through the forward ODE.

   (c) Assemble $\widehat{G}(\theta^k)$ from the per-sample Jacobians $\partial_\theta T_\theta(z_i)$, cf. (20).

   (d) Solve $\widehat{G}(\theta^k)\,\xi = \nabla_\theta E(\theta^k)$ by CG with backtracking (Section 3.3).

   (e) Update $\theta^{k+1} = \theta^k - \alpha\, \min\bigl(1,\, C/\|\xi\|\bigr)\,\xi$.

3. **Reconstruct.** Using the best parameters θ^* (lowest energy seen), evaluate T_{θ^*} on a uniform grid, compute ρ_θ, set $u = \sqrt{\rho_\theta}$, normalize in L^2, and export.

Since $E(\theta^k)$ is not guaranteed to decrease monotonically (the CG solve is approximate and the step may overshoot), the best-parameter tracking in step 3 is used for the final reconstruction.

Remark 3. *There are two alternatives in the algorithm. First, one may use MINRES instead of CG. It avoids using Tikhonov regularization. The stopping criteria in MINRES plays the regularization role. Second, there is no need to form $\hat{G}$ explicitly. The matrix vector multiplication can be directly implemented through $\partial_\theta T_\theta$.*

3.5. Product-map ansatz and dimension-specific choices

The algorithm above is dimension-independent. For $d > 1$ with an additively separable potential $V(\mathbf{x}) = \sum_{k=1}^{d} V_k(x_k)$, we adopt a product-map ansatz:

$$T_\theta(\mathbf{z}) = \big(T_{\theta_1}(z_1),\ \ldots,\ T_{\theta_d}(z_d)\big),$$

where each T_{θ_k} is an independent copy of the 1D boundary-preserving Neural ODE with its own parameters $\theta_k \in \mathbb{R}^{M_1}$; the total parameter count is $M = d \cdot M_1$. The metric tensor $\widehat{G}$ inherits a block-diagonal structure from the product factorization, though our implementation assembles the full matrix. In general, the product-map ansatz incurs a representation error since the true ground-state density ρ^* may not be a product.

Reference densities. The reference density μ must vanish at $\partial\Omega$ (Remark 2), and the particle locations $z_i \sim \mu$ should concentrate where the ground-state density is large. We use three choices:

Problem	Domain	Reference μ
1D	$(-1, 1)$	Beta(2,2): $\frac{3}{4}(1 - z^2)$
2D	$(-16, 16)^2$	Gaussian mixture per axis
3D	$(-8, 8)^3$	Beta(5,5)^3: $C(1 - z^2/L^2)^4$

For the 2D problem, $\mu_1(z) \propto (1 - z^2/L^2) \sum_{k \in \mathcal{W}} e^{-(z-k)^2/(2\sigma^2)}$ with $\mathcal{W} = \{-12, -8, -4, 0, 4, 8, 12\}$, $\sigma = 1.5$ (Section 4.2).

4. Numerical Results

In 2D and 3D experiments below, we first run PWGF to obtain an approximate ground state, and then use it as a warm start for the H^1 Sobolev gradient flow [23]. Thus we refer to the PWGF output as the *PWGF warm start* throughout. Table 1 collects the hyperparameters used in all three experiments. All three use the same learning rate $\alpha = 0.005$; the 1D and 2D experiments use clip $C = 10$ while the 3D experiment uses $C = 50$. The 2D problem additionally uses 4-fold sign-symmetric sampling (Section 4.2) to enforce the symmetry of the potential and keep network biases at zero throughout optimization.

Table 1: Hyperparameters for all three experiments. $M_1 = H^2 + 4H + 1$ (parameter count for a $1 \to H \to H \to 1$ network); $M = d \cdot M_1$ for d-dimensional product maps.

Parameter	1D	2D	3D
N (particles)	3000	3000	6000
K (steps)	400	400	400
α (step size)	0.005	0.005	0.005
H (hidden width)	10	10	10
M (parameters)	141	282	423
N_{ODE} (Euler steps)	10	10	10
n_{CG} (CG iterations)	100	200	300
E_{tol} (backtracking)	0.05	0.05	5.0
Clip C	10	10	50

4.1. 1D Test

We solve the stationary GPE on $\Omega = (-1, 1)$ with Dirichlet boundary conditions:

$$-u'' + V(x)\,u + \beta|u|^2 u = \lambda u, \quad x \in (-1, 1), \quad u(\pm 1) = 0, \quad \|u\|_{L^2} = 1. \tag{29}$$

The potential and interaction coefficient are

$$V(x) = \beta \cos^2\Big(\frac{\pi(x+1)}{2}\Big), \qquad \beta = 10.$$

This problem has the exact ground-state solution

$$u^*(x) = \sin\Big(\frac{\pi(x+1)}{2}\Big), \qquad \lambda^* = \frac{\pi^2}{4} + \beta, \qquad E^* = \frac{\lambda^*}{2} - \frac{3\beta}{16}. \tag{30}$$

Numerically: $\lambda^* \approx 12.4674$, $E^* \approx 4.3587$.

Figure 1 shows the PWGF iterates with the settings in Table 1 (runtime $\approx$ 26 s on Macbook Pro with M1 CPU). The energy estimate plateaus at $E \approx 4.187$, below $E^* \approx 4.359$, because the particle-based Fisher estimator systematically underestimates F_Q due to tail under-sampling (see Section 4.1). The wavefunction error $\|u^k - u^*\|_{L^2}$ decreases from 0.20 at step 1 to 0.036 by step 100 and remains stable thereafter.

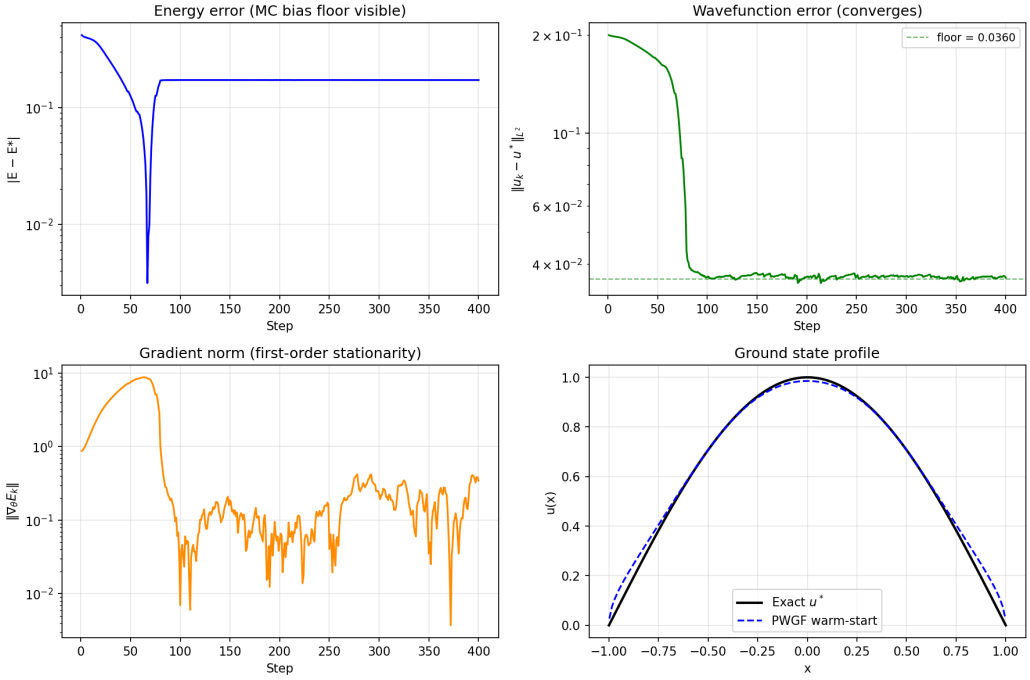

Figure 1: 1D PWGF ground-state computation. *Top-left:* energy error $|E^k - E^*|$ (log scale). *Top-right:* L^2 error $\|u^k - u^*\|$; decreases to the floor ≈ 0.036 (Table 2). *Bottom-left:* gradient norm $\|\nabla_\theta E^k\|$. *Bottom-right:* reconstructed u_θ vs. exact u^*.

Remark 4 (Energy below E^*)**.** *The particle estimate $E_{\mathrm{PWGF}} \approx 4.19$ falls below the exact ground-state energy $E^* \approx 4.36$. This does not violate the variational principle: the true energy $E(\rho_\theta)$ is above E^*, but its Monte Carlo estimate is biased downward. All three energy components F_Q, F_V, F_R are estimated by particle quadrature and are subject to bias; the ablation study in Section 4.1 identifies F_Q (Fisher information) as the dominant source.*

Sensitivity to hyperparameters

To identify the dominant source of the $\approx 3.6\%$ error floor, we vary each hyperparameter independently while holding the others fixed.

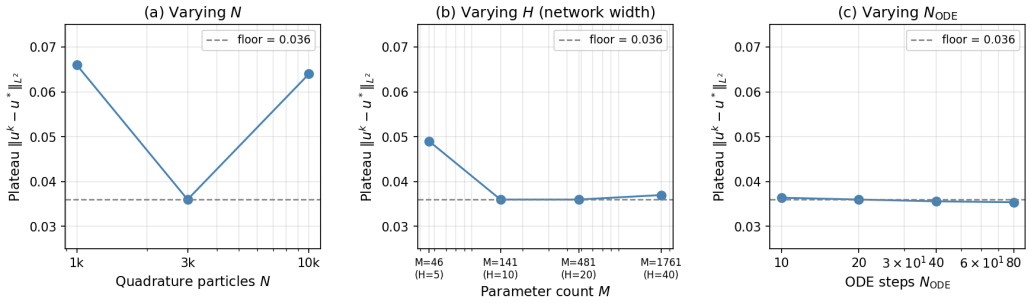

Figure 2: Plateau $\|u^k - u^*\|_{L^2}$ vs. each hyperparameter (baseline: $N = 3000$, $H = 10$, $N_{\text{ODE}} = 10$; dashed line at 0.036). *(a)* N: non-monotone; $N = 3000$ is optimal. *(b)* H: saturates at $H = 10$. *(c)* N_{ODE}: negligible improvement (0.0364 $\to$ 0.0354 for 8$\times$ more compute).

Axis varied	Range	Plateau range	Observations
N (particles)	$1000 \to 10000$	0.036–0.066	non-monotone
H (network width)	$5 \to 40$	0.036–0.049	improves H=5 $\to$ 10
N_{ODE} (ODE steps)	$10 \to 80$	0.0354–0.0364	little improvement

Table 2: Error-floor ablations (Figure 2). The bottleneck is MC bias in $\nabla_\theta F_Q$ from tail under-sampling.

All three ablations identify the same bottleneck: systematic MC bias in the Fisher estimator F_Q from tail under-sampling. Increasing N beyond 3000 does not help because the optimizer converges more tightly to the biased MC minimum; network capacity saturates at $H = 10$; and 8$\times$ more ODE steps yield only 2.7% improvement.

4.2. 2D Test

The second test problem is the GPE on $(-16, 16)^2$ with $\beta = 10$:

$$-\Delta u + V(x_1, x_2)\, u + \beta |u|^2 u = \lambda u, \quad (x_1, x_2) \in (-16, 16)^2, \quad \beta = 10, \quad (31)$$

with potential $V(x_1, x_2) = 2\sin^2(\pi x_1/4)\sin^2(\pi x_2/4)$. No analytical ground state is known with the reference values $E^* \approx 0.2171$ computed by Sobolev gradient flows from [23].

4.2.1. Reference density

The potential vanishes on a 7×7 grid at $(x_1, x_2) = (4j, 4k)$, $j, k \in \{-3, \dots, 3\}$, so the ground-state density has 49 bumps. We use a *Gaussian-mixture reference density* that already places mass at all seven potential wells per coordinate:

$$\mu_1(z) = C \cdot \Big(1 - \frac{z^2}{L^2}\Big) \sum_{k \in \mathcal{W}} \exp\Big(-\frac{(z-k)^2}{2\sigma^2}\Big), \quad \mathcal{W} = \{-12, -8, -4, 0, 4, 8, 12\}, \tag{32}$$

where $\sigma = 1.5$, the boundary factor enforces $\mu_1(\pm L) = 0$ and C is a normalization constant. The 2D reference is $\mu(\mathbf{z}) = \mu_1(z_1)\,\mu_1(z_2)$, and its score

$$\partial_z \log \mu_1(z) = \sum_{k \in \mathcal{W}} w_k(z) \cdot \frac{-(z-k)}{\sigma^2} - \frac{2z}{L^2 - z^2}, \tag{33}$$

where $w_k(z)$ are softmax-normalized Gaussian weights, enters the augmented ODE for the Fisher estimator.

The symmetry $V(-x_1, x_2) = V(x_1, -x_2) = V(x_1, x_2)$ implies the optimal maps are odd: $T_k(-z) = -T_k(z)$. We enforce this analytically via *4-fold sign-symmetric sampling*: draw $N/4$ base particles (z_1^+, z_2^+) from the positive quadrant and include all four sign mirrors $(\pm z_1^+, \pm z_2^+)$. Gradient contributions of each bias parameter cancel identically across opposite pairs, keeping the network biases at zero throughout the iteration.

4.2.2. Results

With the settings in Table 1, the energy decreases from ≈ 0.225 at step 20 to $E_{\text{best}} = 0.2162$ by step 400, with the CG step accepted at every iteration. The 7×7 multi-bump structure is reproduced (Figure 3). CPU runtime is 78 s on Macbook Pro with M1 CPU.

4.2.3. Warm-start comparison with the H^1 gradient flow

The PWGF output is bilinearly interpolated onto the $n = 200$ FD grid and used to initialize the H^1 gradient flow:

Initialization	Initial FD energy $E^{(0)}$	$\lvert E^1 - E_h^* \rvert$
Constant one	0.6495	8×10^{-3}
Random ($\lvert\mathcal{N}(0,1)\rvert$, seed 42)	29.49	1.5×10^{-2}
PWGF warm start	**0.2327**	$\mathbf{1.3 \times 10^{-3}}$

At H1 step 1, the PWGF warm start is $\sim 7\times$ closer to E_h^* than constant-one and $\sim 11\times$ closer than random (Figure 3).

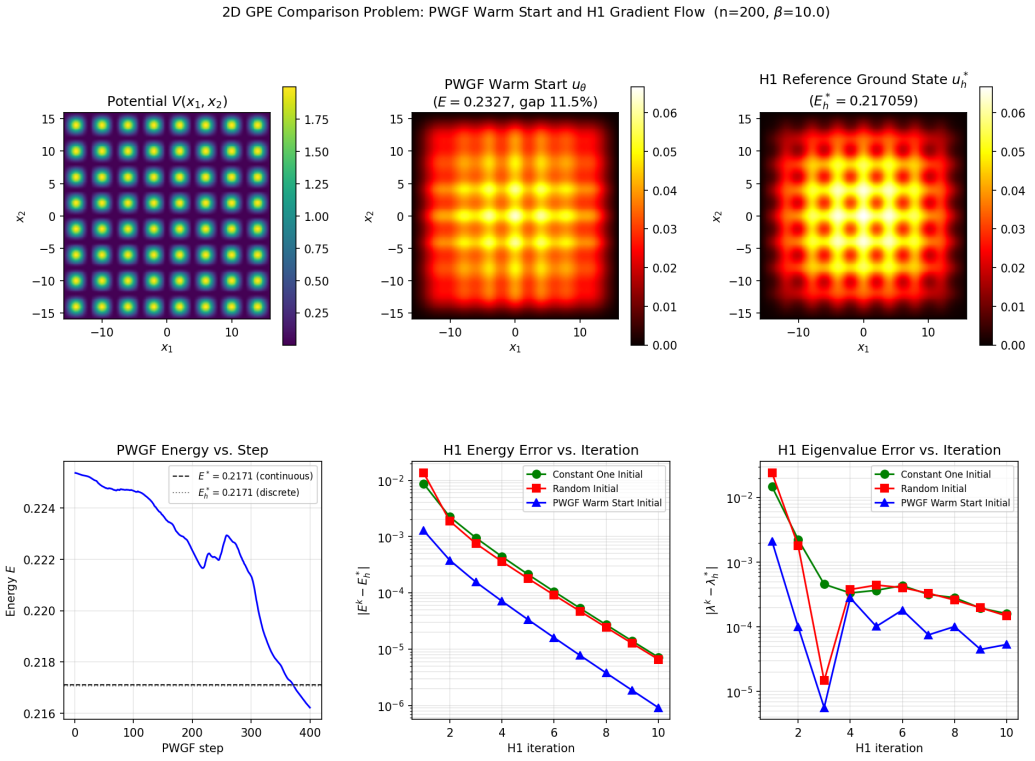

Figure 3: 2D problem ($\beta = 10$, $n = 200$). *Top:* potential V, PWGF warm start u_θ ($E_{\mathrm{FD}} = 0.2327$), and H1 reference u_h^* ($E_h^* = 0.21706$). *Bottom left:* PWGF energy history. *Bottom center/right:* energy and eigenvalue error vs. H1 iteration.

4.3. 3D Test

For the third test we take the GPE on $(-8, 8)^3$ with $\beta = 1600$:

$$-\Delta u + V(x_1, x_2, x_3)\, u + \beta |u|^2 u = \lambda u, \quad \mathbf{x} \in (-8, 8)^3, \quad \beta = 1600, \tag{34}$$

with harmonic trap plus optical lattice

$$V(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 + 100\Big(\sin^2 \tfrac{\pi x_1}{4} + \sin^2 \tfrac{\pi x_2}{4} + \sin^2 \tfrac{\pi x_3}{4}\Big). \tag{35}$$

Reference values from [23] are $E^* = 33.80228$ and $\lambda^* = 80.895$.

4.3.1. Reference density

We use a scaled Beta(5,5) reference density on each coordinate interval $(-L, L)$, concentrated within $|z_k| < 4$ where the potential is low:

$$\mu(\mathbf{z}) = C_1^3 \prod_{k=1}^{3} \Big(1 - \frac{z_k^2}{L^2}\Big)^4, \quad C_1 = \frac{315}{256L}, \tag{36}$$

with score $\partial_{z_k} \log \mu = -8z_k/(L^2 - z_k^2)$. The normalization $C_1 = 315/(256L)$ follows from $\int_{-1}^{1}(1 - t^2)^4\, dt = 256/315$, obtained by expanding $(1 - t^2)^4$ and integrating term by term. Quadrature uses $N = 6000$ particles drawn with 8-fold sign symmetry (750 base triplets $\times$ all $(\pm, \pm, \pm)$ sign patterns), preserving the sign-symmetry of $\nabla_\theta E$. All three coordinate networks are initialized identically with small random weights (scale 0.01) to avoid the saddle point at zero initialization.

4.3.2. Results

With the settings in Table 1, the energy decreases from $E \approx 84$ at step 20 to $E_{\text{best}} \approx 37.25$ (within 10% of E^*) by step 300–400. The reference solution has a $3 \times 3 \times 3$ blob structure dictated by the optical lattice (Figure 4). The PWGF solution after $K = 400$ steps shown in Figure 6 is certainly still far from the true ground state, but it is much better than unimodal Beta(5,5) shown in Figure 5. In other words, this 3D test does show that PWGF can recover some interesting structure about the true minimizer of GP energy, though it is quite difficult to use PWGF to obtain a highly accurate solution.

Despite this qualitative mismatch, the PWGF warm start still provides a useful initialization for the H1 flow because the total mass is placed in the correct spatial region ($|x_k| < 4$), even though its fine structure is wrong.

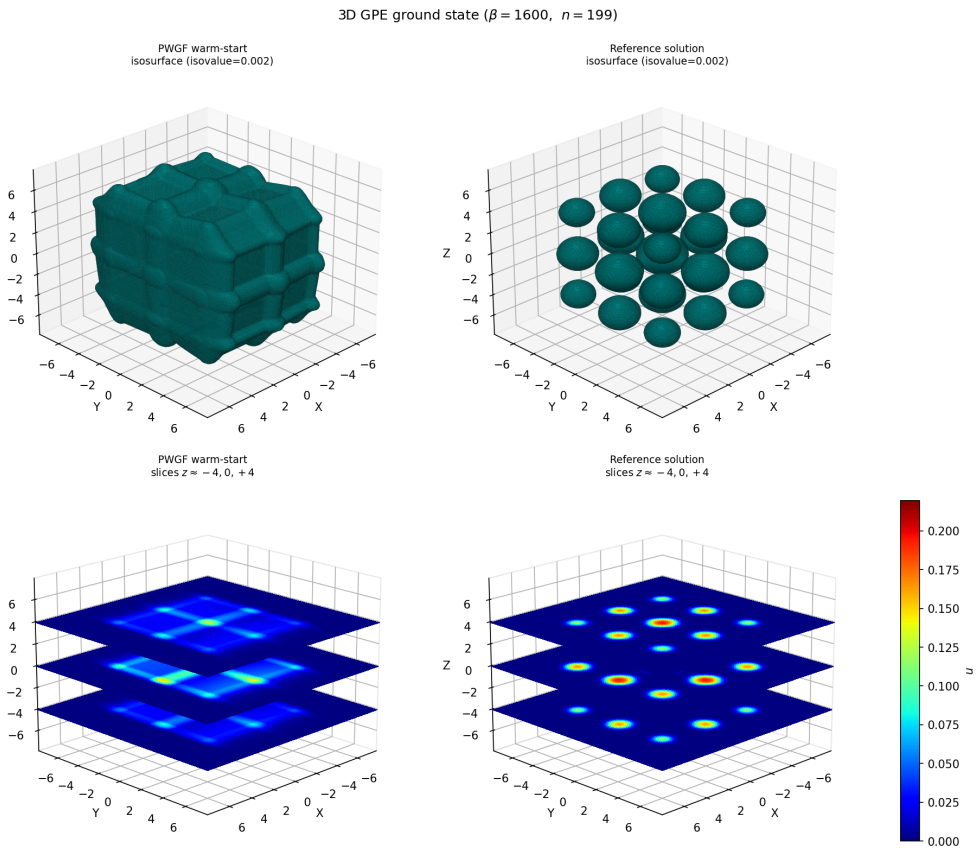

Figure 4: 3D GPE ($\beta = 1600$, $n = 199$). **(a)**–**(b)** Isosurfaces at level 0.002: PWGF solution ($E = 55.26$) and reference solution ($E_h^* = 33.794$, $3 \times 3 \times 3$ blobs). **(c)**–**(d)** Slices at $z \approx -4, 0, +4$; peak ≈ 0.221.

Initialization	$E^{(0)}$	Gap above E^*	$\lvert E^1 - E^*\rvert$
Constant-one	108.40	221%	5.57×10^1
Random ($\lvert\mathcal{N}(0,1)\rvert$, seed 42)	150.92	347%	9.42×10^1
PWGF warm start	**55.24**	**63**%	$\mathbf{1.24 \times 10^1}$

Exact reference: $E^* = 33.80228$, $\lambda^* = 80.89511$. $E^{(0)}$ is the finite-difference energy of each initialization on the n=99 grid.

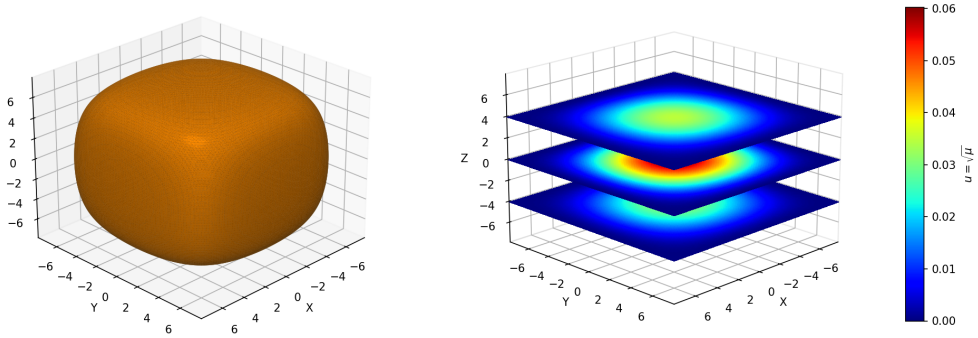

Figure 5: 3D Beta(5,5)^3 density $\mu(\mathbf{z}) = C_1^3 \prod_{k=1}^3 (1 - z_k^2/L^2)^4$ on $(-8, 8)^3$, displayed with the same isovalue and colorbar as Figure 4. **Left:** isosurface at level 0.002 — a single smooth blob centered at the origin, in contrast to the $3 \times 3 \times 3$ multi-blob structure of the true ground state (Figure 4). **Right:** slices at $z \approx -4, 0, +4$.

4.3.3. Warm-start comparison with the H^1 gradient flow

The PWGF output (on a 40^3 grid) is trilinearly interpolated onto the $n = 99$ FD grid and used to initialize the H1 gradient flow:

The PWGF warm start enters the H1 flow ∼4.5× closer to E^* than the constant-one cold start and ∼7.6× closer than the random cold start at step 1. After 10 H1 steps, the warm start reaches $E = 34.48$ (within 2% of E^*), while constant-one reaches 36.04 and random reaches 40.82 (Figure 6).

The advantage is smaller than in 2D (∼7–11×) because the PWGF particle estimate $E_{\text{PWGF}} \approx 37.25$ inflates to $E_{\text{FD}} = 55.24$ on the FD grid (Fisher MC noise, interpolation error, and product-map representation gap), but the initial FD energy is still roughly half that of either cold start.

Re-interpolating the 40^3 PWGF output onto a finer $n = 199$ grid gives the same initial energy ($E^{(0)}_{\text{warm}} = 55.26$); see Figure 7. The PWGF phase runs

in $\approx 240\,\mathrm{s}$ (Macbook Pro with M1 CPU, K=400 steps, M=423 parameters) for the 3D problem.

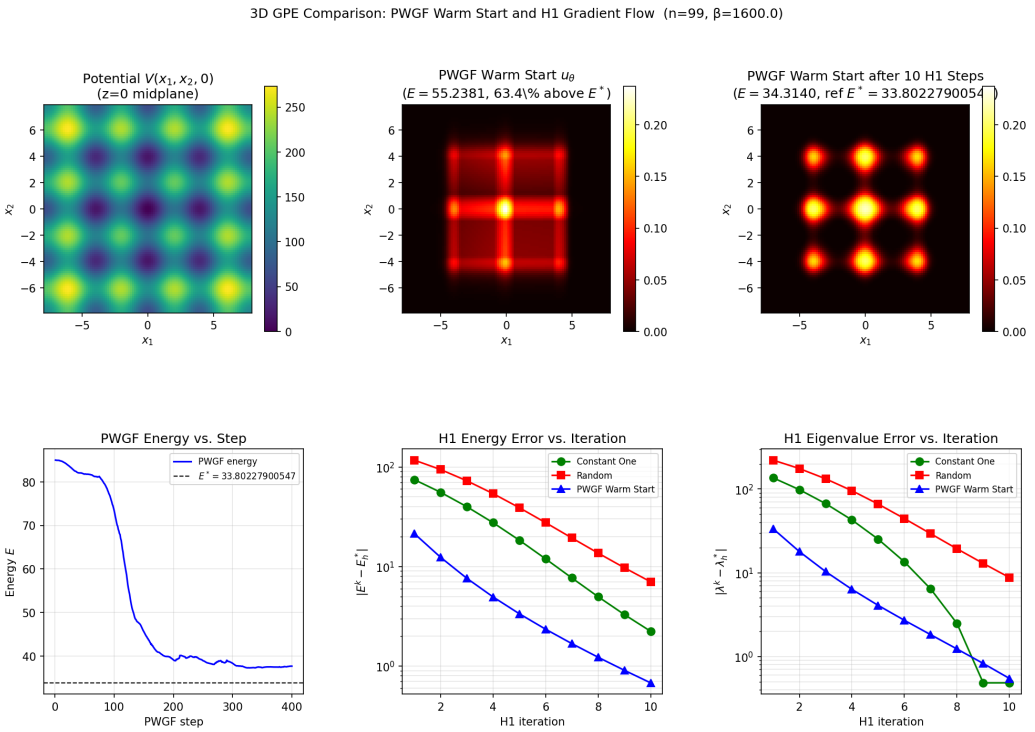

Figure 6: 3D GPE ($\beta = 1600$, $n = 99$). *Top:* potential V midplane, PWGF warm start ($E_{\mathrm{FD}} = 55.2$), and warm start after 10 H1 steps ($E = 34.48$). *Bottom left:* PWGF energy history. *Bottom center/right:* energy and eigenvalue error vs. H1 iteration. Reference: $E^* = 33.802$, $\lambda^* = 80.895$.

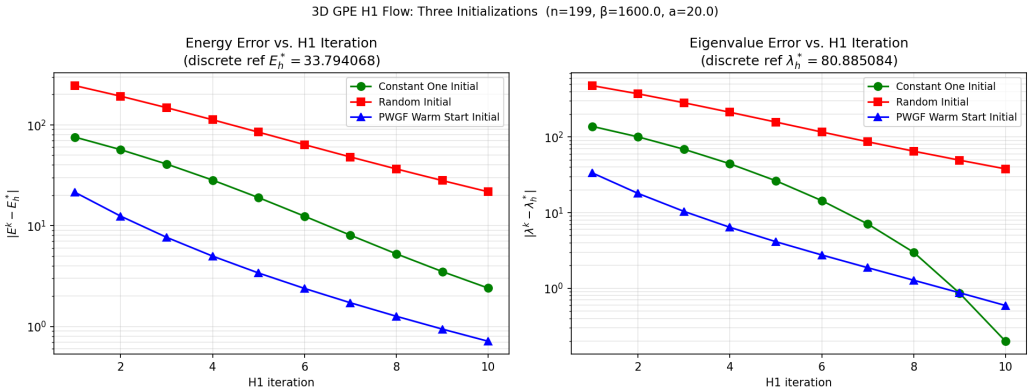

Figure 7: 3D GPE on the $n = 199$ grid ($\beta = 1600$, $h = 0.08$). Energy error $|E^k - E_h^*|$ and eigenvalue error $|\lambda^k - \lambda_h^*|$ vs. H1 iteration; same three initializations as Figure 6. Discrete reference: $E_h^* = 33.794$, $\lambda_h^* = 80.885$.

5. Conclusion

We developed PWGF, a mesh-free method that computes the GPE ground state by natural gradient descent in the parameter space of a boundary-preserving Neural ODE, with the Fisher information evaluated analytically via an augmented ODE. Numerical experiments with different reference densities for three different tests in dimension $d = 1, 2, 3$ show energy gaps of $\sim 4\%$, $\sim 0.4\%$, and $\sim 10\%$ relative to reference solutions. The PWGF output can be used as an effective initial condition in the H^1 Sobolev gradient flow of [23]. Convergence analysis of the parameterized Wasserstein gradient flow remains open and all results in this paper are empirical.

Acknowledgements

This research is partially supported by NSF grants DMS-2208518, DMS-2307465, and DMS-2510829.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article. The Python code for all numerical experiments is available from the authors upon request.

CRediT authorship contribution statement

Xiangxiong Zhang: Software, Formal analysis, Investigation, Visualization, Writing – Original Draft, Writing – Review & Editing. **Haomin Zhou:** Conceptualization, Methodology, Writing – Review & Editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Claude (Anthropic) in order to assist with mathematical discussions, numerical implementation, and drafting of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] R. Altmann, P. Henning, and D. Peterseim, *The J-method for the Gross–Pitaevskii eigenvalue problem*, Numer. Math., 148 (2021), pp. 575–610.

[2] S.-I. Amari, *Natural gradient works efficiently in learning*, Neural Comput., 10 (1998), pp. 251–276.

[3] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, 2nd ed., Lectures in Mathematics ETH Zürich, Birkhäuser, Basel, 2008.

[4] W. Bao and X. Ruan, *Computing ground states of Bose–Einstein condensates with higher order interaction via a regularized density function formulation*, SIAM J. Sci. Comput., 41 (2019), pp. B1284–B1309.

[5] W. Bao and Q. Du, *Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow*, SIAM J. Sci. Comput., 25 (2004), pp. 1674–1697.

[6] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.

[7] E. Cancès, R. Chakir, and Y. Maday, *Numerical analysis of nonlinear eigenvalue problems*, J. Sci. Comput., 45 (2010), pp. 90–117.

[8] E. Cancès, R. Chakir, L. He, and Y. Maday, *Two-grid methods for a class of nonlinear elliptic eigenvalue problems*, IMA J. Numer. Anal., 38 (2018), pp. 605–645.

[9] Z. Chen, J. Lu, Y. Lu, and X. Zhang, *On the convergence of Sobolev gradient flow for the Gross–Pitaevskii eigenvalue problem*, SIAM J. Numer. Anal., 62 (2024), pp. 667–691.

[10] I. Danaila and B. Protas, *Computation of ground states of the Gross–Pitaevskii functional via Riemannian optimization*, SIAM J. Sci. Comput., 39 (2017), pp. B1102–B1129.

[11] E. P. Gross, *Structure of a quantized vortex in boson systems*, Nuovo Cimento, 20 (1961), pp. 454–477.

[12] P. Henning, *The dependency of spectral gaps on the convergence of the inverse iteration for a nonlinear eigenvector problem*, Math. Models Methods Appl. Sci., 33 (2023), pp. 1517–1544.

[13] P. Henning and D. Peterseim, *Sobolev gradient flow for the Gross–Pitaevskii eigenvalue problem: Global convergence and computational efficiency*, SIAM J. Numer. Anal., 58 (2020), pp. 1744–1772.

[14] P. Henning and E. Jarlebring, *The Gross–Pitaevskii equation and eigenvector nonlinearities: Numerical methods and algorithms*, SIAM Rev., 67 (2025), pp. 256–317.

[15] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker–Planck equation*, SIAM J. Math. Anal., 29 (1998), pp. 1–17.

[16] E. H. Lieb, R. Seiringer, and J. Yngvason, *Bosons in a trap: A rigorous derivation of the Gross–Pitaevskii energy functional*, Phys. Rev. A, **61** (2000), 043602.

[17] E. H. Lieb, R. Seiringer, J. P. Solovej, and J. Yngvason, *The Mathematics of the Bose Gas and its Condensation*, Oberwolfach Seminars, vol. 34, Birkhäuser, Basel, 2005.

[18] S. Liu, H. Wu, X. Ye, and H. Zhou, *A parameterized Wasserstein Hamiltonian flow approach for solving the Schrödinger equation*, arXiv:2505.11762 (2025).

[19] H. Wu, S. Liu, X. Ye, and H. Zhou, *Parameterized Wasserstein Hamiltonian flow*, SIAM J. Numer. Anal. **63** (2025), pp. 360–395.

[20] F. Otto, *The geometry of dissipative evolution equations: the porous medium equation*, Comm. Partial Differential Equations, 26 (2001), pp. 101–174.

[21] L. P. Pitaevskii, *Vortex lines in an imperfect Bose gas*, Sov. Phys. JETP, 13 (1961), pp. 451–454.

[22] C. Villani, *Topics in Optimal Transport*, Graduate Studies in Mathematics, vol. 58, American Mathematical Society, Providence, RI, 2003.

[23] Z. Chen, J. Lu, Y. Lu, and X. Zhang, *Fully discretized Sobolev gradient flow for the Gross–Pitaevskii eigenvalue problem*, *Mathematics of Computation* **94** (2025), 2723–2760.

[24] Y. Jin, S. Liu, H. Wu, X. Ye, and H. Zhou, *Parameterized Wasserstein gradient flow*, J. Comput. Phys. **524** (2025), Article 113660.

[25] K. Neklyudov, J. Nys, L. Thiede, J. Carrasquilla, Q. Liu, M. Welling, and A. Makhzani, *Wasserstein Quantum Monte Carlo: A novel approach for solving the quantum many-body Schrödinger equation*, Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 63461–63482.

[26] W. Bao, Z. Chang, and X. Zhao, *Computing ground states of Bose–Einstein condensation by normalized deep neural network*, J. Comput. Phys. **520** (2025), Article 113486.

[27] T. Chu, X. Dai, J. Wu, and A. Zhou, *A gradient flow model for the Gross–Pitaevskii problem: mathematical and numerical analysis*, arXiv:2510.27235 (2025).

[28] D. J. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015, pp. 1530–1538.

[29] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, J. Mach. Learn. Res., 22 (2021), pp. 1–64.

[30] D. Peterseim, J.-F. Pietschmann, J. Püschel, and K. Ruess, *Neural network acceleration of iterative methods for nonlinear Schrödinger eigenvalue problems*, J. Comput. Appl. Math. (2026), Article 117414.