A SCORE-BASED MODEL FOR LEARNING NEURAL WAVEFUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantum Monte Carlo coupled with neural network wavefunctions has shown success in finding the ground state of quantum many-body systems. The existing optimization approaches compute the energy by sampling local energy from an explicit probability distribution given by the wavefunction. In this work, we provide a new optimization framework for obtaining properties of quantum many-body ground state using score-based neural networks. This new framework does not require explicit probability distribution and performs the sampling via Langevin dynamics. Our method is based on the key observation that the local energy is directly related to the score, defined as the gradient of the logarithmic wavefunction. Inspired by the score matching and the diffusion Monte Carlo methods, we derive a weighted score matching objective, which guides our score-based models to correctly converge to the ground state. We first validate our approach with experiments on quantum harmonic traps, and further results show that it can accurately learn the ground states of atomic systems. By implicitly modeling the high-dimensional data distribution, our work paves the way toward a more efficient representation of quantum systems.

1 INTRODUCTION

Understanding the properties of quantum systems lies at the core of many scientific disciplines such as condensed matter physics, material science, and quantum chemistry. A quantum system is characterized by its ground state wavefunction, formally obtained by solving the Schrödinger equation. However, because of the exponentially large Hilbert space, direct solving the Schrödinger equation for quantum systems with many particles is impractical. Owning its strong dimension reduction abilities, deep learning stands as a strong candidate to approximately solve the Schrödinger equation and extract properties of quantum systems with the desired accuracy. For example, under the supervised learning setting, deep learning methods have been successfully applied to predict the quantum properties of molecular systems based on training data from density functional theory (DFT) calculation (Schütt et al., 2017; Gasteiger et al., 2020; Liu et al., 2022). However, supervised methods rely on expensive computational simulations to generate a large amount of training data, and the accuracy of these methods is fundamentally limited by the data quality. Furthermore, DFT calculations involve various approximations and are not guaranteed to reach the true ground state. A common scheme for approximately solving the Schrödinger equation is the variational principle. The variational principle optimizes a trial wavefunction to reach the ground state by minimizing its energy as much as possible, which can be done efficiently using quantum Monte Carlo. Such a method is called variational Monte Carlo, whose accuracy relies on the expressive power of the trial wavefunction. Recently, the integration of deep learning and variational Monte Carlo has unleashed the potential of both methods (Carleo & Troyer, 2017). Powered by the efficient sampling and optimization framework of quantum Monte Carlo and the universal approximation ability of deep neural networks, neural wavefunctions can accurately model the quantum states, and remarkable improvement has been achieved (Pfau et al., 2020; Hermann et al., 2020).

Modeling a wavefunction is conceptually similar to modeling a probability density. Existing methods model the wavefunction explicitly by training a neural network to directly output the wavefunction values. However, numerous examples in machine learning have shown that implicitly modeling the data distribution provides better representation (Kingma & Welling, 2014; Goodfellow et al., 2014; Ho et al., 2020). Properly chosen latent embedding can better describe the structure of the

data space. As our direct reference, score-based methods have demonstrated their strong ability in generative modeling (Song & Ermon, 2019; Song et al., 2020). The score is defined as the gradient of the log probability. For example, realistic images can be generated from random noise by following the dynamics defined by the score. In this paper, we show that the quantum wavefunction can be represented by score models and be optimized within the QMC framework.

Our motivation to connect the score-based model with quantum Monte Carlo is based on an interesting observation between the energy computation and the score-based formulation. In QMC, the energy of the system is averaged over the local energy of plausible quantum states. Our observation is that the local energy only involves the gradient of the logarithmic wavefunction, which we define as the score of the wavefunction. As a result, to minimize the energy, the score must be explicitly computed. On the other hand, the actual wavefunction values are only used for sampling and optimization. To this end, we propose a new optimization framework for QMC where sampling and optimization are also achieved by using the score function alone, eliminating the need to explicitly compute the wavefunction value. In our proposed score-based framework, the sampling is done via Langevin dynamics and the optimization is done through a new loss function inspired by diffusion Monte Carlo.

Our score-based method enables the possibility of performing QMC computation with only the score function, which is infeasible in existing optimization frameworks. The direct benefit is that by predicting the gradient, we omit the necessity to recompute it from the wavefunction. Moreover, the score function can be interpreted as the force of the quantum system, by implicitly modeling the distribution with the score function, the dynamics of the quantum system could be better captured. Our experimental results show that with our score-based optimization framework, the ground state of a quantum system can be accurately learned.

2 Background and Related Work

2.1 Quantum many-body wavefunction in continuous space

Let $\boldsymbol{x} \in \mathbb{R}^{N\times d}$ be the coordinates of N particles in d-dimension, the quantum state of the system is defined by its wavefunction $\psi : \mathbb{R}^{N\times d} \to \mathbb{R}$. By definition ψ is normalized ($\int_{\boldsymbol{x}} |\psi(\boldsymbol{x})|^2 = 1$) and $|\psi(\boldsymbol{x})|^2$ gives the probability density of observing $\boldsymbol{x}$. The wavefunction ψ is the solution to the time-independent Schrödinger equation:

$$\hat{H}\psi(\boldsymbol{x}) = E\psi(\boldsymbol{x}), \tag{1}$$

where $\hat{H}$ is an operator called the Hamiltonian, and E is a scalar that gives the energy of the system. The Hamiltonian is defined as:

$$\hat{H}\psi(\boldsymbol{x}) = -\frac{1}{2}\sum_i \nabla_i^2\psi(\boldsymbol{x}) + V(\boldsymbol{x})\psi(\boldsymbol{x}), \tag{2}$$

where the index i goes over all of the $N \times d$ dimensions in the summation. The first term in the Hamiltonian takes the sum of the second-order partial derivatives of the wavefunction and is related to the kinetic energy of the system. The second term in the Hamiltonian multiplies the wavefunction by a scalar value and is related to the potential energy of the system. The kinetic term is intrinsic to the Schrödinger equation and always takes the same form, whereas the potential function $V : \mathbb{R}^{N\times d} \to \mathbb{R}$ varies for different physics problems. Note that although a wavefunction can be complex-valued in general, the solution to Equation 1 is real-valued because $\hat{H}$ is real.

Our objective is to find the ground state ψ_0, which is the solution associated with the lowest energy E_0. Note that the Hamiltonian $\hat{H}$ is a linear operator. Hence the solutions to Equation 1 are spanned by the eigenfunctions of $\hat{H}$. Therefore we essentially try to find the eigen pair (ψ_0, E_0) with the smallest eigenvalue.

In our notation, the coordinates $\boldsymbol{x}$ can be either viewed as N d-dimensional vectors or as a flattened $N \cdot d$ dimensional vector. In the remaining of this paper, depending on the context, we may use the bold font $\boldsymbol{x}_i$ to denote the i-th particle or use the regular font x_i to denote the i-th scalar component of the flattened vector.

2.2 VARIATIONAL MONTE CARLO

The variational Monte Carlo (VMC) method uses a parameterized function $\psi_{\boldsymbol{\theta}} : \mathbb{R}^{N\times d} \to \mathbb{R}$ (called the Ansatz) to model the wavefunction, where $\boldsymbol{\theta}$ denotes the parameters to be optimized. The normalization of $\psi_{\boldsymbol{\theta}}$ is not required. The energy expectation of $\psi_{\boldsymbol{\theta}}$ is computed as:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{\int \psi_{\boldsymbol{\theta}}(\boldsymbol{x})\hat{H}\psi_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x}}{\int \psi_{\boldsymbol{\theta}}(\boldsymbol{x})\psi_{\boldsymbol{\theta}}(\boldsymbol{x})d\boldsymbol{x}} = \frac{\int \psi_{\boldsymbol{\theta}}(\boldsymbol{x})^2 \frac{\hat{H}\psi_{\boldsymbol{\theta}}(\boldsymbol{x})}{\psi_{\boldsymbol{\theta}}(x)}d\boldsymbol{x}}{\int \psi_{\boldsymbol{\theta}}(\boldsymbol{x})^2 d\boldsymbol{x}} = \mathbb{E}_{\boldsymbol{x}\sim\psi^2_{\boldsymbol{\theta}}} E_L(\boldsymbol{x};\boldsymbol{\theta}), \tag{3}$$

where $E_L(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\hat{H}\psi_{\boldsymbol{\theta}}(\boldsymbol{x})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{x})}$ is called the *local energy* of $\boldsymbol{x}$. The expectation is evaluated numerically on sparse samples. Markov Chain Monte Carlo (MCMC) sampling is employed to drive sample distribution to converge to the target density $\psi^2_{\boldsymbol{\theta}}$.

We can approach the ground state wavefunction by minimizing the energy expectation with gradient descent. The unbiased gradient of $\mathcal{L}(\boldsymbol{\theta})$ w.r.t. the parameters $\boldsymbol{\theta}$ is given by:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{x}) = 2\mathbb{E}_{\boldsymbol{x}\sim\psi^2_{\boldsymbol{\theta}}}\left[\left(E_L(\boldsymbol{x};\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{x}\sim\psi^2_{\boldsymbol{\theta}}}[E_L(\boldsymbol{x};\boldsymbol{\theta})]\right)\nabla_{\boldsymbol{\theta}}\log|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})|\right], \tag{4}$$

where the expectations are evaluated by average over samples. The derivation of this loss makes use of the fact that $\hat{H}$ is Hermitian (Ceperley et al., 1977). A detailed derivation can be found in Appendix E of Lin et al. (2021). The optimized energy expectation value gives the approximated ground state energy, and the eigenvalue formulation ensures the estimation to be variational, that is, the energy expectation defined by the ansatz is always above the true ground state energy. Other properties of the quantum systems can also be estimated by taking expectations over the corresponding operators.

Neural quantum state based on the restrictive Boltzmann machine was initially studied for simple quantum spin models on lattices (Carleo & Troyer, 2017). More complicated neural network architectures, such as convolutional neural network and graph neural network are then extended to more complicated spin systems to capture frustration due to the lattice structure and next nearest neighboring interaction (Kochkov et al., 2021; Fu et al., 2022). Similar ideas are also studied for ab-initio simulation in quantum chemistry (Han et al., 2019) to study properties of small molecules. Recently, FermiNet Pfau et al. (2020) and PauliNet Hermann et al. (2020) greatly improve the accuracy of neural wavefunction and apply to larger molecular systems. Follow-up works contribute to improved performance (Gerard et al., 2022), joint training for multiple geometries (Gao & Günnemann, 2021; Scherbela et al., 2022) or solving for excited state (Entwistle et al., 2022). All of the existing methods use neural networks to explicitly model the wavefunction, while in this work we propose to implicitly model the quantum state using the score function.

2.3 SCORE-BASED NEURAL WAVEFUNCTIONS

For the Hamiltonian $\hat{H}$ defined in Equation 2 the local energy can be expressed in terms of the logarithmic wavefunction as:

$$E_L(\boldsymbol{x};\boldsymbol{\theta}) = -\frac{1}{2}\sum_i \left(\frac{\partial^2 \log|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})|}{\partial x_i^2} + \left(\frac{\partial \log|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})|}{\partial x_i}\right)^2\right) + V(\boldsymbol{x}). \tag{5}$$

One can easily prove this expression by using the fact that $\frac{\partial \log|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})|}{\partial x_i} = \frac{1}{x_i}\frac{\partial \psi_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial x_i}$. If we take a closer look at this expression, we can notice that the local energy depends only on the gradient of the logarithmic wavefunction. In fact, if we define the function $\boldsymbol{s}_{\boldsymbol{\theta}} : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$ such that $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})_i = \frac{\partial \log|\psi_{\boldsymbol{\theta}}(\boldsymbol{x})|}{\partial x_i}$, we can rewrite the local energy as:

$$E_L(\boldsymbol{x};\boldsymbol{\theta}) = -\frac{1}{2}\left(\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})) + \|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2\right) + V(\boldsymbol{x}) \tag{6}$$

The function $\boldsymbol{s}_{\boldsymbol{\theta}}$ is called the *quantum force* or the *drift* in quantum Monte Carlo. Here we follow the convention from the machine learning community and call it the *score*. The direct relationship between the local energy and the score motivates us to ask the question: can we represent the wavefunction using only the score function? As a direct benefit, this can avoid computing the first-order derivatives in evaluating the local energy and consequently provide a more succinct representation.

To answer this question, two questions need to be answered: (1) is the score alone sufficient to model the wavefunction? and (2) how to optimize the parameters to minimize the energy expectation? Score matching gives a definite answer to the first question. For the second question, we seek inspiration from diffusion Monte Carlo.

2.4 DENSITY ESTIMATION VIA SCORE MATCHING

Score matching (Hyvärinen & Dayan, 2005) provides a way to model the sample distribution with the score function. Assume we model the sample distribution with a non-normalized parameterized function $f_{\boldsymbol{\theta}}$. The probability density of sample $\boldsymbol{x}$ is given by $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})/Z(\boldsymbol{\theta})$ where $Z(\boldsymbol{\theta})$ is the normalization term, defined as $Z(\boldsymbol{\theta}) = \int_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$. One major difficulty is that in most cases we don't know how to compute $Z(\boldsymbol{\theta})$. The idea of score matching is that instead of dealing directly with $p_{\boldsymbol{\theta}}$ we seek to estimate the score, defined as $2\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$. Note that the score does not depend on the normalization term since $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \log Z(\boldsymbol{\theta})$ and $Z(\boldsymbol{\theta})$ is independent of any specific $\boldsymbol{x}$. Note that since in our case $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \psi_{\boldsymbol{\theta}}(\boldsymbol{x})^2$, there is a multiplicative factor of 2 between the score defined here and the score defined for wavefunction.

The key insight of score matching is that one can make $2\boldsymbol{s}_{\boldsymbol{\theta}}$ converge to the true score (of the sample distribution) without explicitly knowing the true score. This is achieved by minimizing the implicit score matching (ISM) objective, defined as follows:

$$\mathrm{ISM}(\boldsymbol{\theta}) = 2\mathbb{E}_{\boldsymbol{x}\sim p_{\text{sample}}} \left[\mathrm{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})) + \|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 \right]. \tag{7}$$

We can see that ISM objective is related to the kinetic part of the local energy by a multiplicative factor of 4.

Score-based methods have shown great success in generative modeling (Song & Ermon, 2019; Song et al., 2020). The effectiveness of implicit density modeling has also been demonstrated by other successful generative models, such as VAEs (Kingma & Welling, 2014), GANs Goodfellow et al. (2014), and diffusion models Ho et al. (2020).

2.5 DIFFUSION MONTE CARLO

Diffusion Monte Carlo (DMC) (Toulouse et al., 2016; Ceperley, 2004) employs score-based diffusion to improve upon VMC. In VMC, the quality of the approximation is limited by the capacity of the ansatz. DMC additionally assigns a weight for each sample in a way that the weighted distribution gives a better approximation of the ground state. In DMC's formulation, each sample is defined by a walker, and each walker is assigned a weight. At each iteration, the walker moves randomly, and the weight is increased or decreased according to the move. In the importance-sampled DMC, a trial function ψ_T is used. At each iteration, the walkers randomly diffuse by following score of the trial function and the weights are updated as:

$$w(\boldsymbol{x}') = w(\boldsymbol{x}) \cdot \exp\left(-\alpha\left(\frac{E_L(\boldsymbol{x}') + E_L(\boldsymbol{x})}{2} - E_T\right)\right), \tag{8}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are the coordinates before and after the move, w is the weight, α is the step size of the move, E_L is the local energy computed with the trial function ψ_T and E_T is a trial energy updated at each iteration to control the scale of overall weight.

DMC is based on the imaginary time evolution where the repeated diffusing and weighting procedure projects out the ground state from the trial wavefunction. At convergence, the ground state energy is estimated by the weighted average of local energies.

Ceperley & Alder (1980) applies DMC to electonic gas. Ren et al. (2022) and Wilson et al. (2021) apply fixed-node DMC starting from the optimized FermiNet ansatz.

3 METHOD

In this section, we discuss how to use a parameterized function (a neural network) to model the score directly and how to correctly find the ground state by optimizing the parameters of the neural

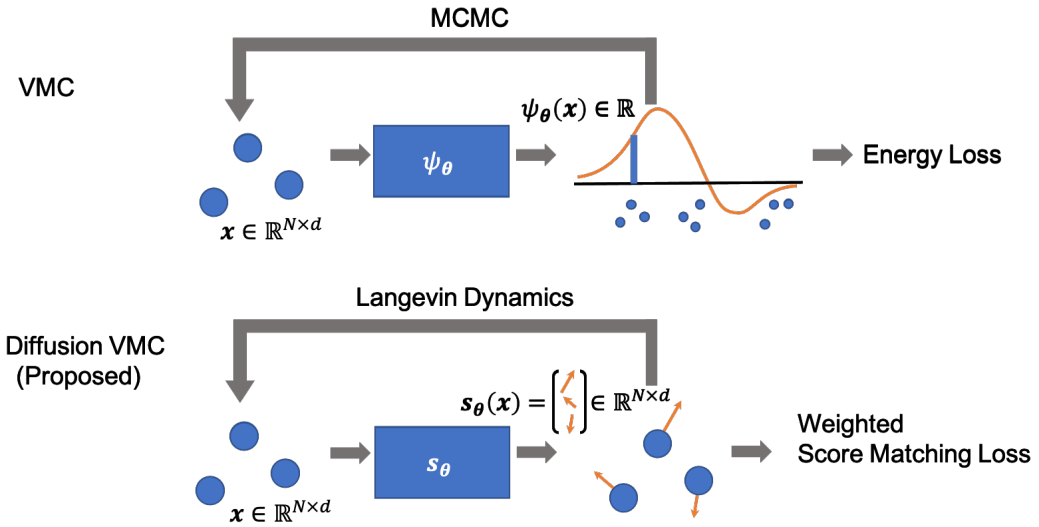

Figure 1: Comparison between the pipeline of VMC and our proposed DVMC. In VMC, the wavefunction is explicitly modeled with a neural network, who directly outputs the wavefunction value. The sampling is carried out with MCMC and the optimization is achieved by minimizing the energy loss. In the proposed DVMC we instead model the score of the wavefunction. The samples are generated via Langevin dynamics and the proposed weighted score matching objective is employed to optimize the score neural network.

wavefunction. Existing methods cannot achieve this goal. Although using score-based method to perform sampling and energy evaluation in VMC is straightforward, it is unclear how to do optimization. This is because the loss function of VMC (Equation 4) explicitly requires the wavefunction. On the other hand, diffusion Monte Carlo works naturally with score-based formulation, but no parameter update step is involved in the diffusion procedure. It should be noted that naively taking the gradient of average energy of the current samples will fail because the underlying distribution will change when the parameters of score network are updated, hence the gradient must also consider the distribution shift.

To this end, our score-based optimization framework follows the gradient descent formulation in VMC, and we design a new loss function inspired by DMC. As we mix the flavors of these two methods, we call our new optimization framework the *Diffusion Variational Monte Carlo* (DVMC). The pipeline of VMC and our proposed DVMC are shown in Figure 1.

3.1 Score-based Neural Wavefunction Ansatz

We use a parameterized score function to implicitly model the wavefunction. Formally, for systems with N particles in d dimension, the score function $\boldsymbol{s}_\theta : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$ maps the set of input coordinates to the set of output scores, which are vectors having the dimensions as the input. Intuitively, after training, the output score should tell the particles the direction toward regions with higher density.

In our case, we are dealing with indistinguishable particles in quantum mechanics. So exchanging two particles will not change the probability density. Formally,

$$\psi(\dots, \boldsymbol{x}_i, \dots, \boldsymbol{x}_j, \dots)^2 = \psi(\dots, \boldsymbol{x}_j, \dots, \boldsymbol{x}_i, \dots)^2. \tag{9}$$

As a result, the score, which is the gradient of logarithmic wavefunction, is permutation equivariant. Formally,

$$\nabla_{\boldsymbol{x}_i} \log|\psi(\dots, \boldsymbol{x}_i, \dots, \boldsymbol{x}_j, \dots)| = \nabla_{\boldsymbol{x}_j} \log|\psi(\dots, \boldsymbol{x}_j, \dots, \boldsymbol{x}_i, \dots)|. \tag{10}$$

Essentially the equivariance means that if two input particles exchange their positions, their corresponding output score vectors will also exchange positions. The proof is straightforward. To save

space we show it for two 1-d particles:

$$\partial_1 \log|\psi(x_1, x_2)| = \lim_{\Delta x \to 0} \frac{\log|\psi(x_1 + \Delta x, x_2)| - \log|\psi(x_1, x_2)|}{\Delta x} \tag{11}$$

$$= \lim_{\Delta x \to 0} \frac{\log|\psi(x_2, x_1 + \Delta x)| - \log|\psi(x_2, x_1)|}{\Delta x} = \partial_2 \log|\psi(x_2, x_1)|. \tag{12}$$

We parameterize the score function with a neural network. The equivariance can be easily achieved by considering the input coordinates as a set.

3.2 Sampling via Langevin Dynamics

To generate samples that follow the distribution implicitly defined by the score, we use Langevin dynamics, which is similar in the score-based generative models (Song & Ermon, 2019). Given the samples at the current time step x_t, the coordinates for the new samples are computed as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \sqrt{\alpha}\boldsymbol{\epsilon} + \alpha \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t), \tag{13}$$

where α is the hyperparameter that defines the step size, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}^{Nd})$ is a random vector sampled from the standard multivariate Gaussian distribution. The process is the same as in DMC and can be understood as first doing a random diffusion, then drifting by following the score vector. We can prove that when α is small, the distribution converges to the distribution defined by the score function.

When doing Langevin dynamics, an accept/reject procedure can be employed to alleviate the finite time error. Although omitted in the context of generative modeling Song & Ermon (2019), this is a standard step in DMC. The original rejection step computes the ratio

$$p = \frac{\exp\left(-\frac{1}{2\alpha}\|\boldsymbol{x} - \boldsymbol{x}' - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}')\alpha)\|^2\right)\psi_{\boldsymbol{\theta}}(\boldsymbol{x}')^2}{\exp\left(-\frac{1}{2\alpha}\|\boldsymbol{x}' - \boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\alpha\|^2\right)\psi_{\boldsymbol{\theta}}(\boldsymbol{x})^2}. \tag{14}$$

However, this expression involves the wavefunction value. By assuming $\log \psi_{\boldsymbol{\theta}}$ to be constant between $\boldsymbol{x}$ and $\boldsymbol{x}'$, we can show that the ration can be approximated only in terms of the score as:

$$p \approx \exp\left(\frac{\alpha}{2}(\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 - \|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}')\|^2)\right). \tag{15}$$

3.3 Neural wavefunction optimization

Inspired by score matching and diffusion Monte Carlo, we propose a weighted score matching objective. In diffusion Monte Carlo, at each step, we create a weighted distribution that is closer to the ground state. To make the score network evolve toward the ground state, we let the score network match the weighted distribution. This can be done by minimizing a weighted score matching objective where the samples with higher weight will contribute more to the score matching objective, mimicking the duplication of that sample. We compute the weight in a similar way as in DMC. To make the loss depend only on the current samples, we make two simplifications to Equation 8. First, we do not accumulate weights across multiple iterations. Second, we assume the local energies before and after the move are close, so that we have $\frac{E_L(\boldsymbol{x}') + E_L(\boldsymbol{x})}{2} \approx E_L(\boldsymbol{x})$. Since we update the parameters stochastically with gradient descent, the direction of the loss is more important than the exact value, so we further drop the dependence on the step size α. With these simplifications, given the batch size of M, the weighted score matching (WSM) objective is defined as:

$$\langle E_L \rangle = \frac{1}{M}\sum_{i=1}^{M} E_L(\boldsymbol{x}_i) \tag{16}$$

$$E_{\text{diff}}(\boldsymbol{x}_i) = E_L(\boldsymbol{x}_i) - \langle E_L \rangle \tag{17}$$

$$\text{WSM}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \frac{\exp\left(-\beta E_{\text{diff}}(\boldsymbol{x}_i)\right)}{\sum_{i=1}^{M}\exp\left(-\beta E_{\text{diff}}(\boldsymbol{x}_i)\right)}\left[\text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{1}{2}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|^2\right] \tag{18}$$

$$= \sum_{i=1}^{M} \text{softmax}\left(-\beta E_{\text{diff}}(\boldsymbol{x}_i)\right)_i \left[\text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{1}{2}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|^2\right], \tag{19}$$

where the softmax function is defined as $\mathrm{softmax}(\boldsymbol{a})_i = \frac{\exp(a_i)}{\sum_i exp(a_i)}$. We subtract the average local energy to improve the numerical stability. This weighting scheme is similar to the attention mechanism in machine learning where the attention scores are based on the local energies. Due to the zero-variance property of the ground state, the weights should in principle converge to 1 at the ground state. Empirically we observe that the convergence is improved if we further normalize E_{diff} by dividing its standard deviation, which will lead to more imbalanced weights when approaching the ground state. We then have the scaled-WSM defined as:

$$\text{Scaled-WSM}(\boldsymbol{\theta}) = \sum_{i=1}^{M} \mathrm{softmax}\left(-\beta \frac{E_{\text{diff}}(\boldsymbol{x}_i)}{\mathrm{Std}\left(E_{\text{diff}}\right)}\right)_i \left[\mathrm{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{1}{2}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\|^2\right]. \tag{20}$$

The proposed loss shares the same behavior with the energy gradient. In both cases, we try to increase the sample density in regions with lower local energy and decrease the sample density in regions with higher local energy.

3.4 The Diffusion Variational Monte Carlo Algorithm

The overall procedure is similar to VMC. We first perform several steps of Langevin dynamics to equilibrate the sampling. Then we compute the weighted score matching loss and update the network parameters via gradient descent. The algorithm is summarized in Algorithm 1.

Algorithm 1 Diffusion Variational Monte Carlo (DVMC)

Input: Randomly initialized sample $\boldsymbol{x}$, score network $\boldsymbol{s}_{\boldsymbol{\theta}}$, step size α, number of iterations T, number of Langevin dynamics step N_{ld}
Output: New sample $\boldsymbol{x}$, optimized score network $\boldsymbol{s}_{\boldsymbol{\theta}}$
  for $t = 1$ to T **do**
    for $i = 1$ to N_{ld} **do**
      $\boldsymbol{x}' = \boldsymbol{x} + \sqrt{\alpha}\boldsymbol{\epsilon} + \alpha \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ ▷ Langevin dynamics (Section 3.2)
      $p_i = \exp\left(\frac{\alpha}{2}(\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 - \|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}')\|^2)\right)$
      Sample $z \sim U(0, 1)$
      if $p > z$ **then**
        $\boldsymbol{x} = \boldsymbol{x}'$ ▷ Accept the move
      end if
    end for
    loss $= \text{Scaled-WSM}(\boldsymbol{\theta})$ ▷ Weighted score matching objective (Section 3.3)
    Update $\boldsymbol{\theta}$ via gradient descent to minimize the loss.
  end for

4 Experiments

In this paper, we first show that our score-based method correctly finds the ground state by solving the harmonic trap. Then we showcase the applicability of our method on simple atomic systems, i.e., interacting fermions with coulomb potential.

4.1 Wavefunction Ansatz for Bosons and Fermions

Since the probability density $\psi^2(\cdot)$ is invariant under exchange of the particle positions (Equation 9), there are two cases for the wavefunction value. The first case is that the wavefunction does not change its sign, i.e., $\psi(\dots, \boldsymbol{x}_i, \dots, \boldsymbol{x}_j, \dots) = \psi(\dots, \boldsymbol{x}_j, \dots, \boldsymbol{x}_i, \dots)$, such particles are called *bosons*. The second case is that the wavefunction change its sign, i.e., $\psi(\dots, \boldsymbol{x}_i, \dots, \boldsymbol{x}_j, \dots) = -\psi(\dots, \boldsymbol{x}_j, \dots, \boldsymbol{x}_i, \dots)$, such particles are called *fermions*. These different symmetries will fundamentally change the ground state. The fermion ground state must consider the antisymmetric constraint and has higher ground state energy than the boson ground state.

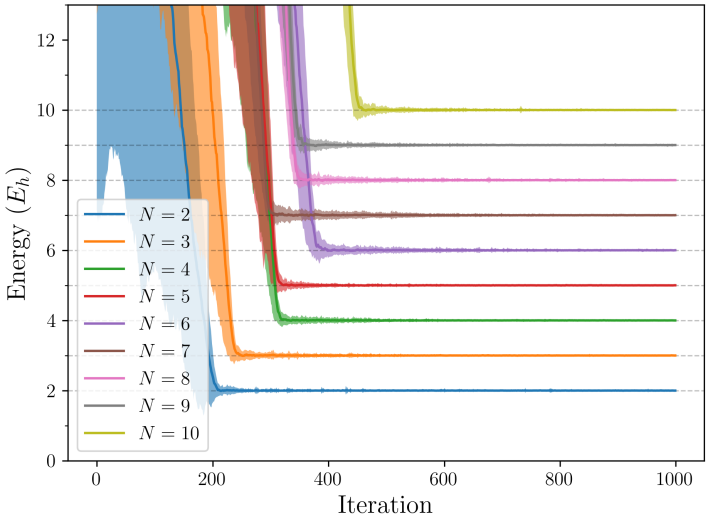

Figure 2: Training energy for Bosons in quantum harmonic trap. N denotes the number of particles. All runs converge smoothly to ground state.

Bosons We use a feed-forward neural network composed of multi-layer perceptrons (MLPs) which satisfies the permutation equivariance. The input feature is the coordinates and the distance to the center. Each particle is simultaneously transformed into two feature vectors with two different MLPs. We perform average pooling for the first feature vector over all particles and obtain a global feature. The global feature is then concatenated with the second feature vector. A final MLP is used to predict the score of each particle. All MLPs are shared among different particles.

Fermions The antisymmetric property of fermions is very challenging to model (Ceperley, 1991). Different from bosons, due to the change of sign, the fermion wavefunction has both positive regions and negative regions. The region where the wavefunction equals zero is called the nodal surface (or the node). In 1-d, the nodal surface is exactly the set where two particles coincide. However, for higher dimension the nodal surface can be arbitrary. The fermion sign structure gives rise some intrinsic difficulties to parametrize the score function.The score function diverges inversely proportionally to the distance away from the nodal surface Umrigar et al. (1993). Since neural networks essentially models continuous functions, using feed forward neural networks to model the score function is prohibitive. In fact, even the networks manage to approximately model the discontinuity, in order to compute the local energy we also requires the derivatives of the score to be accurate, which is more difficult.

Nevertheless, we are still able to illustrate how our score-based framework works with fermions. We can do so by modeling the score as the gradient of an antisymmetric wavefunction ansatz. In our experiments with fermions, we use the score computed from FermiNet Pfau et al. (2020) where Slater determinants are employed to ensure the antisymmetry. Although by doing so the wavefunction value is practically computed, our goal is to evaluate the our method in this more challenging fermion setting. Also as discussed above, currently this stands as the only feasible option.

4.2 Boson: Quantum Harmonic Trap

The Schrödinger equation with harmonic potential

$$V_{\text{QHO}}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2. \tag{21}$$

describes a quantum harmonic oscillator, one of the most famous quantum systems that is solvable. The energy levels and eigenstates are known analytically. In particular, for n bosons in the d

dimensional harmonic potential, the ground state energy and wavefunction are

$$E = \frac{nd}{2}, \quad |\psi\rangle = \frac{1}{2^{nd/2}} \exp(-\frac{1}{2}\|\boldsymbol{x}\|^2) \tag{22}$$

Therefore, the quantum harmonic oscillator provides an analytical benchmark for our numerical method. Furthermore, despite its simplicity, the quantum harmonic oscillator is of important experimental relevance and can be realized in cold atom systems by trapping atoms using lasers Dalfovo et al. (1999).

We conduct simulation with various number of particles. The results are shown in Figure 2. All runs correctly converge to the ground state.

4.3 Fermion: Atomic systems

For atoms, we work in the Born-Oppenheimer where we assume the atoms are fixed in space and only the electrons are allowed to move. Hence, our inputs are the coordinates of the electrons. The potential due to the Coulomb interactions is:

$$V_{\text{Coulumb}}(\boldsymbol{x}) = \sum_{i>j} \frac{1}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} - \sum_{iI} \frac{Z_I}{\|\boldsymbol{x}_i - \boldsymbol{R}_I\|} + \sum_{I>J} \frac{Z_I Z_J}{\|\boldsymbol{R}_I - \boldsymbol{R}_J\|}. \tag{23}$$

As mentioned above, we compute the score by taking the gradient of FermiNet Pfau et al. (2020). Following Lin et al. (2021), four atoms are tested, including Boron, Carbon, Nitrogen and Oxygen. To show the effect of step size, We trained and tested the score network with α=1e-3 and α=1e-2. We and optimize for 200k iterations for all systems. We use a batch size of 256 for Boron and Carbon, and 512 for Nitrogen and Oxygen. We do Langevin dynamics for 20 steps between two parameter update. We use β=2 for Oxygen and β=1 for all other atoms. The norm of the score is clipped at 20 to increase numerical stability.

	Ours α=1e-3	Ours α=1e-2	FermiNet	HF	Reference
B	-24.64893(22)	-24.65233(3)	-24.65370(3)	-24.53316	-24.65391
C	-37.84414(13)	-37.83445(5)	-37.84471(5)	-37.6938	-37.8450
N	-54.58171(29)	-54.53580(12)	-54.58882(6)	-54.4047	-54.5892
O	-75.06587(63)	-74.94880(709)	-75.06655(7)	-74.8192	-75.0673

Table 1: Experimental results on Atoms in Hartree (E_h). Underline denotes our results within chemical accuracy. References taken from Lin et al. (2021).

The results are summarized on Table 1. The chemical accuracy is defined as 1.594 mE_h (Pfau et al., 2020). Except for the Nitrogen atom, all atoms can enter the chemical accuracy under one setting, although the effect of step size is mixed. In many cases a higher level of variance is observed. The reason may be due to the difficulty in optimizing the gradient network. Moreover, FermiNet works with the second order optimizer KFAC Martens & Grosse (2015), which may not be suitable for our loss function.

5 Discussion and Conclusion

Inspired by the connection between the score-based formulation of local energy, we explore the possibility to implicitly model the quantum wavefunction. With the weighted score matching objective, the proposed DVMC enables the possibility to optimize the score network toward the ground state. Experiments show that our proposed method can accurately find the ground state for both bosons and fermions.

Author Contributions

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

David Ceperley, Geoffrey V Chester, and Malvin H Kalos. Monte carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081, 1977.

David M Ceperley. Fermion nodes. *Journal of statistical physics*, 63(5):1237–1267, 1991.

David M Ceperley. Introduction to quantum monte carlo methods applied to the electron gas. In *The Electron Liquid Paradigm in Condensed Matter Physics*, pp. 3–42. IOS Press, 2004.

David M Ceperley and Berni J Alder. Ground state of the electron gas by a stochastic method. *Physical review letters*, 45(7):566, 1980.

Franco Dalfovo, Stefano Giorgini, Lev P Pitaevskii, and Sandro Stringari. Theory of bose-einstein condensation in trapped gases. *Reviews of modern physics*, 71(3):463, 1999.

Mike Entwistle, Zeno Schätzle, Paolo A Erdman, Jan Hermann, and Frank Noé. Electronic excited states in deep variational monte carlo. *arXiv preprint arXiv:2203.09472*, 2022.

Cong Fu, Xuan Zhang, Huixin Zhang, Hongyi Ling, Shenglong Xu, and Shuiwang Ji. Lattice convolutional networks for learning ground states of quantum many-body systems. *arXiv preprint arXiv:2206.07370*, 2022.

Nicholas Gao and Stephan Günnemann. Ab-initio potential energy surfaces by pairing gnns with neural wave functions. In *International Conference on Learning Representations*, 2021.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

Leon Gerard, Michael Scherbela, Philipp Marquetand, and Philipp Grohs. Gold-standard solutions to the schrödinger equation using deep learning: How much physics do we need? *arXiv preprint arXiv:2205.09438*, 2022.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. 2014.

Jiequn Han, Linfeng Zhang, and E Weinan. Solving many-electron schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, 2019.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

Dmitrii Kochkov, Tobias Pfaff, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Bryan K Clark. Learning ground states of quantum hamiltonians with graph networks. *arXiv preprint arXiv:2110.06390*, 2021.

Jeffmin Lin, Gil Goldshlager, and Lin Lin. Explicitly antisymmetrized neural network layers for variational monte carlo simulation. *arXiv preprint arXiv:2112.03491*, 2021.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=givsRXsOt9r.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

D. Pfau, J.S. Spencer, A.G. de G. Matthews, and W.M.C. Foulkes. Ab-initio solution of the many-electron schrödinger equation with deep neural networks. *Phys. Rev. Research*, 2:033429, 2020. doi: 10.1103/PhysRevResearch.2.033429. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429.

Weiluo Ren, Weizhong Fu, and Ji Chen. Towards the ground state of molecules via diffusion monte carlo on neural networks. *arXiv preprint arXiv:2204.13903*, 2022.

Michael Scherbela, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs. Solving the electronic schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nature Computational Science*, 2(5):331–341, 2022.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Julien Toulouse, Roland Assaraf, and Cyrus J Umrigar. Introduction to the variational and diffusion monte carlo methods. In *Advances in Quantum Chemistry*, volume 73, pp. 285–314. Elsevier, 2016.

CJ Umrigar, MP Nightingale, and KJ Runge. A diffusion monte carlo algorithm with very small time-step errors. *The Journal of chemical physics*, 99(4):2865–2890, 1993.

Max Wilson, Nicholas Gao, Filip Wudarski, Eleanor Rieffel, and Norm M Tubman. Simulations of state-of-the-art fermionic neural network wave functions with diffusion monte carlo. *arXiv preprint arXiv:2103.12570*, 2021.