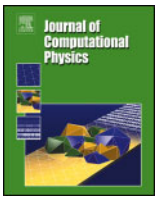
Journal of Computational Physics 513 (2024) 113140

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Convergence of variational Monte Carlo simulation and scale-invariant pre-training

Nilin Abrahamsen [a,*], Zhiyan Ding [b], Gil Goldshlager [b], Lin Lin [b,c]

[a] The Simons Institute for the Theory of Computing, Berkeley, CA 94720, USA
[b] Department of Mathematics, University of California, Berkeley, CA 94720, USA
[c] Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

ARTICLE INFO

Keywords:
Variational Monte Carlo
Gradient descent
Unbiased estimator

ABSTRACT

We provide theoretical convergence bounds for the variational Monte Carlo (VMC) method as applied to optimize neural network wave functions for the electronic structure problem. We study both the energy minimization phase and the supervised pre-training phase that is commonly used prior to energy minimization. For the energy minimization phase, the standard algorithm is scale-invariant by design, and we provide a proof of convergence for this algorithm without modifications. The pre-training stage typically does not feature such scale-invariance. We propose using a scale-invariant loss for the pretraining phase and demonstrate empirically that it leads to faster pre-training.

1. Introduction

A central goal of computational physics and chemistry is to find a *ground state* which minimizes the energy of a system. Electronic structure theory studies the behavior of electrons that govern the chemical properties of molecules and materials. The energy in the electronic structure is a functional defined on the set of quantum wave functions ψ, which are normalized $\mathscr{L}^2$-integrable functions determined up to an arbitrary scalar multiplication factor, known as a *phase*. Equivalently, the wave function can be viewed as an element of the projective space rather than a normalized function.

The *variational Monte Carlo* (VMC) algorithm [1–3] is a powerful method for optimizing the energy of a parameterized wave function through stochastic gradient estimates. The method relies on Monte Carlo samples from the probability distribution defined by the Born rule, which views the normalized squared wave function as a probability density function.

Historically, the number of parameters used in VMC simulations was relatively small, and methods such as stochastic reconfiguration [4,5] and the linear method [6,7] were preferred for parameter optimization. Recently, neural networks have been used to parameterize the wave function in an approach known as *neural quantum states* [8]. Subsequent work has shown that VMC with neural quantum states can match or exceed state-of-the-art high-accuracy quantum simulation methods for a variety of electronic structure problems, including molecules in both first quantization [9–12] and second quantization [13,14], electron gas [15–17], and solids [18].

* Corresponding author.
E-mail addresses: nilin@berkeley.edu (N. Abrahamsen), zding.m@math.berkeley.edu (Z. Ding), ggoldshlager@gmail.com (G. Goldshlager), linlin@math.berkeley.edu (L. Lin).

https://doi.org/10.1016/j.jcp.2024.113140
Received 18 May 2023; Received in revised form 3 May 2024; Accepted 23 May 2024

Available online 27 May 2024

Due to the complexity of quantum wave functions parameterized by neural networks, it is in general not possible to explicitly normalize the parameterized wave function. One of the essential properties of the VMC procedure is thus its ability to usean unnormalized function ψ_θ to represent the corresponding element in the projective space. This is achieved by pulling back the energy functional to be minimized from the projective space to a *scale-invariant* functional on the space of unnormalized wave functions. This scale-invariant functional is the same as the Rayleigh quotient $\langle\psi|\, H\,|\psi\rangle / \langle\psi|\psi\rangle$ where the Hermitian operator H is known as the Hamiltonian. The scale-invariance of the energy functional is reflected in the VMC algorithm where a scale-invariant *local energy* function corresponding to the wave function is evaluated at MCMC-sampled electron configurations. For the ground state problem considered here it suffices to restrict to real-valued wave functions, so we will take all scalars to be real.

The wave function in the VMC algorithm is often initialized using a supervised pre-training stage [11] where it is fitted to a less expressive approximation to the ground state, for example given as a Slater determinant. In contrast to the main VMC algorithm which is trained with an inherently scale-invariant objective, this supervised pre-training typically does not encode the scale-invariant property [11,19]. We propose using a scale-invariant loss function for this supervised stage and demonstrate numerically that this modification leads to faster convergence towards the target in the supervised pre-training setting. We further give a theoretical convergence bound for the proposed scale-invariant supervised optimization algorithm. As an important ingredient in this analysis, we introduce the concept of a *directionally unbiased* stochastic estimator for the gradient.

1.1. Related works

Using scale-invariance in the parameter vector to stabilize SGD training is a well-studied technique. Many structures and methods have been proposed along these lines, such as normalized NN [20–22], $\mathcal{G}$-SGD [23], SGD+WD [24–27], and projected/normalized (S)GD [28,29]. In these works, the parameter $\mathbf{v}$ is stored explicitly and the loss function $\mathcal{L}$ is scale-invariant in the parameter $\mathbf{v}$, which means $\mathcal{L}(\lambda\mathbf{v}) = \mathbf{v}$ for any $\lambda > 0$.

The setting of neural wave functions differs from typical scale-invariant loss functions in that the scale-invariant functional $\mathcal{L}$ is applied after a nonlinear parameterization $\theta \mapsto \psi_\theta$, and the function ψ_θ can be queried but not accessed as an explicit vector. The composed loss function $L(\theta) = \mathcal{L}(\psi_\theta)$ is not scale-invariant in its parameters. We can obtain a relation between the two settings by applying the chain rule involving a functional derivative, but the resulting expression involves factors that we need to estimate by sampling, in contrast with the typical setting of scale-invariance in the parameter space.

1.1.1. SGD on Riemannian manifolds

The training of a scale-invariant objective can be seen as a training process on projective space, which is a special case of learning on Riemannian manifolds. The convergence of SGD on Riemannian manifolds is well-studied; for example [30] shows the asymptotic convergence of SGD on Riemannian manifolds. Further variations of the SGD algorithm on Riemannian manifolds have been proposed and studied, such as SVRG on Riemannian manifolds [31], [32] and ASGD on Riemannian manifolds [33]. To the best of our knowledge, all previous works ensure the parameter stays on the manifold by either taking stochastic gradient steps on the manifold or projecting them back onto it. This is in contrast to our setting where no projection onto a submanifold is needed.

1.1.2. Concurrent theoretical analysis

The concurrent work of [34] proves a similar convergence result for VMC and also takes into account the effect of Markov chain Monte Carlo sampling. However, this work does not consider the setting of supervised pre-training.

2. Variational Monte Carlo

A quantum wave function is a square-integrable function ψ on a space of configurations Ω. A typical configuration is $\Omega = \mathbb{R}^{3N}$ for a system of N electrons.[1] The term "square-integrable function" refers to a function ψ with a finite $\mathscr{L}^2$-norm $\|\psi\| = \sqrt{\int_\Omega |\psi|^2 \mathrm{d}x} < \infty$ with respect to the Lebesgue measure. For any scalar $\lambda \in \mathbb{R}$ and $\lambda \neq 0$, $\lambda\psi$ represents the same physical state as ψ. In the variational Monte Carlo algorithm, ψ_θ is an Ansatz parameterized by θ, and the objective is to find the parameter vector θ such that ψ_θ minimizes the energy

$$\mathcal{L}(\psi) = \frac{\langle\psi|\, H\,|\psi\rangle}{\langle\psi|\psi\rangle}.$$

The *Born rule* associates to a quantum wave function ψ a probability distribution on Ω with density p_ψ given by

$$p_\psi(x) = |\psi(x)|^2 / \|\psi\|^2 . \tag{1}$$

The norm $\|\psi\|$ is unknown to the optimization algorithm, and samples $\{X_i\} \sim p_\psi$ are generated using Markov chain Monte Carlo (MCMC). We will assume that these samples are independent and sampled from the exact distribution p_ψ. In practice, the independence assumption is tested by evaluating the autocorrelation of the MCMC samples.

[1] The spin degrees of freedom are considered as classical and are expressed in the symmetry type of ψ_θ.

The energy of ψ can then be expressed as an expectation with respect to p_ψ: $\mathcal{L}(\psi) = \mathbb{E}_{X\sim p_\psi} \mathcal{E}_\psi(X)$, where

$$\mathcal{E}_\psi(x) = \frac{(H\psi)(x)}{\psi(x)} \tag{2}$$

is called the *local energy*. We note that the local energy is not well-defined in the case where $\psi(x) = 0$, but this is not a practical issue since $p_\psi(x) = 0$ by definition at such points and thus the expectation value is still well-defined.

We use $L(\theta) = \mathcal{L}(\psi_\theta)$ to denote the loss as a function of the parameters.

Problem 2.1 *(VMC).* Minimize

$$L(\theta) = \mathbb{E}_{X\sim p_\theta} \mathcal{E}_\theta(X), \tag{3}$$

over parameter vectors $\theta \in \mathbb{R}^d$, where $\mathcal{E}_\theta(x) = H\psi_\theta(x)/\psi_\theta(x)$ and $p_\theta(x) = |\psi_\theta(x)|^2/\|\psi_\theta\|^2$. We assume access to samples from $\{X_i\} \sim p_\theta$ and query access to ψ_θ, $\nabla_\theta \psi_\theta$, and $\mathcal{E}_\theta$ which are assumed to be C^∞ with respect to θ for any $x \in \Omega$.

When $\|\psi_\theta\| < \infty$, the gradient of the VMC energy functional takes the form (see e.g., equation (9) of [11])

$$\nabla_\theta L(\theta) = 2\mathbb{E}_{X\sim p_\theta}\Big[\big(\mathcal{E}_\theta(X) - L(\theta)\big) \nabla_\theta \log|\psi_\theta(X)| \Big]. \tag{4}$$

Notably, the change in local energy does not contribute to this formula, that is, $\mathbb{E}_{X\sim p_\theta}[\nabla_\theta \mathcal{E}_\theta(X)] = 0$. Instead, the gradient of L is entirely due to the perturbation of the sampling distribution as θ varies. For completeness, we include the derivation of Equation (4) in Appendix A. The condition that ψ_θ is C^∞ with respect to θ is typically satisfied by using the logistic sigmoid activation function.

3. Supervised pre-training

In order to stabilize the VMC training, the neural quantum state ψ_θ in FermiNet [11] and subsequent works [35,36,19] is initialized by fitting the Ansatz to an initial guess ψ at an approximate ground state [11,19]. This target initial state can be taken to be a Slater determinant optimized using the self-consistent field (SCF) method [37]. We therefore consider the problem of minimizing the distance from the line $\{\lambda\psi | \lambda \in \mathbb{R}\}$ to a target function $\varphi \in \mathscr{L}^2(\Omega)$.

That is,

$$\mathcal{L}(\psi) = \min_{\lambda\in\mathbb{R}} \|\lambda\psi - \varphi\|_\rho^2 , \tag{5}$$

where $\|\phi\|_\rho = \sqrt{\int_\Omega |\phi(x)|^2 \mathrm{d}\rho(x)}$ is the norm in $\mathscr{L}^2$-sense on a probability space (Ω, ρ) with probability measure ρ. In practice the pre-training may fit intermediate vector-valued features such as orbitals [19] instead of the scalar-valued wave function. As we note in Section 5.1, this is formally equivalent to fitting a scalar wave function. Moreover, as we show in Section 5.1, exploiting scale-invariance when fitting the orbitals can improve the convergence of the wave function itself as measured by Equation (5).

Here, ρ is typically the Lebesgue measure, but in Section 5.1 we will also consider a case where it is defined by the target function. This loss function is scale-invariant by construction and has a closed-form expression

$$\mathcal{L}(\psi) = \|\varphi\|_\rho^2 - \big(|\langle\varphi, \psi\rangle_\rho|/\|\psi\|_\rho\big)^2 . \tag{6}$$

Note that for normalized target φ, the second term of Equation (6) is the squared sine of the angle between the vectors, $\mathcal{L}(\psi) = \sin^2 \angle(\varphi, \psi)$. Minimizing $\mathcal{L}(\psi)$ is equivalent to minimizing $\mathcal{L}(\psi) = -\langle\varphi, \psi\rangle_\rho/\|\psi\|_\rho$. Again we write the loss as $L(\theta) = \mathcal{L}(\psi_\theta)$ as a function of the variational parameters:

Problem 3.1 *(Supervised learning pre-training).* Given a target function $\varphi \in \mathscr{L}^2(\Omega, \rho)$, access to samples $\{X_i\} \sim \rho$ and query access to φ, ψ_θ, and $\nabla_\theta \psi_\theta$, minimize

$$L(\theta) = -\frac{\langle\varphi, \psi_\theta\rangle_\rho}{\|\psi_\theta\|_\rho} \tag{7}$$

over parameter vectors $\theta \in \mathbb{R}^d$.

3.1. Directionally unbiased gradient estimator

By applying the chain rule we can write the gradient of $L(\theta)$ as

$$\nabla_\theta L(\theta) = [\partial_\psi \mathcal{L}](\psi_\theta) \cdot \nabla_\theta \psi_\theta, \tag{8}$$

where $\partial_\psi \mathcal{L}(\psi_\theta)$ is the functional derivative of the scale-invariant loss. This can be evaluated analogously to the gradient derived in [28] for a scale-invariant loss function. The resulting gradient expression is

$$\nabla_\theta L(\theta) = -\frac{\langle \varphi, \nabla_\theta \psi_\theta \rangle_\rho}{\|\psi_\theta\|_\rho} + \frac{\langle \varphi, \psi_\theta \rangle_\rho \langle \psi_\theta, \nabla_\theta \psi_\theta \rangle_\rho}{\|\psi_\theta\|_\rho^3}. \tag{9}$$

We cannot compute the infinite-dimensional $\mathscr{L}^2$ norms in the denominator of Equation (9) and do not have access to an unbiased estimator for Equation (9). However, in Lemma 4.5, we propose a *directionally unbiased* gradient estimator G, which means that $\mathbb{E}[G]$ is in the positive span of $\nabla_\theta L$. Additionally, we will demonstrate that the directionally unbiased gradient estimator is sufficient for achieving rapid convergence of SGD as long as the norm $\|\psi_\theta\|_\rho$ can be approximately estimated, see detail in Corollary 4.10.

4. Theoretical results

4.1. Convergence result for VMC

In the VMC setting, we assume that the MCMC subroutine is able to sample exactly from the distribution $p_\theta(x) = |\psi_\theta(x)|^2/\|\psi_\theta\|^2$. This sampling oracle makes it possible to construct a cheap and unbiased gradient estimation for the energy functional (3). For simplicity we use exact real-valued arithmetic.

From the formula (4) for the gradient, we derive the following unbiased gradient estimator for $\nabla_\theta L(\theta)$:

Lemma 4.1. *Given $n \geq 2$ i.i.d. samples $\{X_i\} \sim p_\theta$, let $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1} \mathcal{E}_\theta(X_i)$ and define the gradient estimate*

$$G(\theta, X) = \frac{2}{n-1} \sum_{i=1}^{n} \left(\mathcal{E}_\theta(X_i) - \hat{L}(\theta)\right) \nabla_\theta \log|\psi_\theta(X_i)|. \tag{10}$$

Then $G(\theta, X)$ is an unbiased estimator of the gradient of the population energy.

We prove Lemma 4.1 in Appendix A.

Using the unbiased gradient estimator (10) from Lemma 4.1, the VMC algorithm can be formulated as a variant of SGD:

Algorithm 1 Variational Monte Carlo.

Preparation: number of iterations: M; learning rate η_m; initial parameter: θ_0; number of samples each iteration: n;
Running:
$m \leftarrow 0$;
while $m \leq M$ **do**
  Sample $\{X_i^m\}_{i=1}^n$ independently according to the density $p_{\theta^m} \propto |\psi_{\theta^m}|^2$;
  $G_m \leftarrow G(\theta, X) = \frac{2}{n-1}\sum_{i=1}^n (\mathcal{E}_\theta(X_i) - \hat{L}(\theta))\nabla_\theta \log|\psi_\theta(X_i)|$;
  $\theta_{m+1} \leftarrow \theta_m - \eta_m G_m$;
end while
Output: θ_m;

Given any θ, we define $\|\cdot\|_{q,p_\theta}$ as the q-th norm under the measure $p_\theta(x)dx$. The convergence bound for the VMC algorithm assumes a uniform bound on the following quantities:

Condition 4.2. There exists a constant C_ψ such that for any $\theta \in \mathbb{R}^d$,

- (Value bound):

$$\|H\psi_\theta/\psi_\theta\|_{4,p_\theta} \leq C_\psi. \tag{11}$$

- (Gradient bound):

$$\begin{aligned} &\|\nabla_\theta H\psi_\theta/\psi_\theta\|_{2,p_\theta} \leq C_\psi, \\ &\|\nabla_\theta \psi_\theta/\psi_\theta\|_{4,p_\theta} \leq C_\psi. \end{aligned} \tag{12}$$

- (Hessian bound):

$$\|\mathsf{H}_\theta\psi_\theta/\psi_\theta\|_{2,p_\theta} \leq C_\psi, \tag{13}$$

  where $\mathsf{H}_\theta\psi_\theta$ is the hessian of ψ_θ in θ.

Condition 4.2 is scale-invariant in ψ_θ, i.e., if $\psi_\theta(x)$ satisfies Condition 4.2, then $\lambda\psi_\theta(x)$ also satisfies the condition with the same constant λ for any $\lambda \neq 0$. We establish new bounds on the Lipschitz constant of ∇L and the variance of the gradient estimator (see Lemma C.1), subject to the constraints specified in Condition 4.2. These bounds are crucial in demonstrating the convergence of our method.

Now, we are ready to give the convergence result for Algorithm 1:

Theorem 4.3. *Under Condition 4.2 there exists a constant $C > 0$ that only depends on C_ψ such that, when $\eta_m < \frac{1}{C}$ for any $m \geq 0$,*

$$\sum_{m=0}^{M} \eta_m \mathbb{E}\left(\left|\nabla_\theta L(\theta_m)\right|^2\right) \leq 2L(\theta_0) + C\sum_{m=0}^{M} \frac{\eta_m^2}{n}, \tag{14}$$

for any $M > 0$.

The proof of the above theorem is given in Appendix C. The upper bound (14) is important for us to determine η_m so that the parameter θ_m converges to a first-order stationary point in the expectation sense when m is moderately large. Theorem 4.3 implies:

Corollary 4.4. *Under Condition 4.2:*

- *Choosing $\eta_m = \Theta\left(n\epsilon^2\right)$ and $M = \Theta\left(1/(n\epsilon^4)\right)$, we have*

$$\min_{0\leq m\leq M} \mathbb{E}\left(\left|\nabla_\theta L(\theta_m)\right|\right) \leq \epsilon.$$

- *Choosing $\eta_m = \Theta\left(\sqrt{\frac{n}{m+1}}\right)$, we have*

$$\min_{0\leq m\leq M} \mathbb{E}\left(\left|\nabla_\theta L(\theta_m)\right|\right) = O\left((nM)^{-1/4}\right).$$

This bound shows that the convergence of Algorithm 1 is the same as the convergence rate of SGD to first-order stationary point for nonconvex functions, see [38, Theorem 2] or [39, Theorem 5.2.1].

4.1.1. Scale-invariant supervised pre-training

The following lemma shows how we can choose a directionally unbiased gradient estimator. Given samples $\{X_i\}$, write $\langle \psi, \varphi\rangle_n = \frac{1}{n}\sum_{i=1}^n \psi(X_i)\varphi(X_i)$ and $\|\psi\|_n^2 = \frac{1}{n}\sum_{i=1}^n \psi(X_i)^2$.

Lemma 4.5. *Given $n \geq 2$ and i.i.d. samples $X_1, \ldots, X_n \sim \rho$, define coefficients*

$$a_j = -\|\psi_\theta\|_n^2\, \varphi(X_j) + \langle \varphi, \psi_\theta\rangle_n\, \psi_\theta(X_j)$$

and let

$$G = \frac{1}{\|\psi_\theta\|_\rho^3}\frac{1}{n-1}\sum_{j=1}^{n} a_j \nabla_\theta \psi_\theta(X_j). \tag{15}$$

Then $\mathbb{E}_X(G) = \nabla_\theta L(\theta)$.

We give the derivation of Lemma 4.5 in Appendix B. Given an estimator $\tilde{Z}$ for $\|\psi_\theta\|_\rho$ which is independent of $X_1, \ldots, X_n$, we then choose the following as our directionally unbiased gradient estimator:

$$G(\theta, X) = \frac{1}{\tilde{Z}^3}\frac{1}{n-1}\sum_{j=1}^{n} a_j \nabla_\theta \psi_\theta(X_j), \tag{16}$$

where $\{a_j\}_{j=1}^n$ are defined in Lemma 4.5.

Remark 4.6. To obtain the independent norm estimate $\tilde{Z}$ we can take $2n$ samples $X_1, \ldots, X_{2n}$ at each iteration and let $\tilde{Z} = (\frac{1}{n}\sum_{i=n+1}^{2n} |\psi_\theta(X_i)|^2)^{1/2}$ in (16). However, our numerics in Section 5.1 suggest it is sufficient in practice to estimate $\|\psi_\theta\|_\rho$ as $\tilde{Z} = (\frac{1}{n}\sum_{i=1}^{n} |\psi_\theta(X_i)|^2)^{1/2}$ without sampling an additional independent minibatch.

Remark 4.7. Some care must be taken when constructing a plug-in estimator for the gradient. For example, given an estimate $\tilde{Z}$ of $\|\psi_\theta\|_\rho$ and samples $X_1, X_2 \sim \rho$, Equation (9) suggests a plug-in estimate of $\nabla_\theta L_\theta$ given by

$$-\frac{\varphi(X_1)\nabla_\theta\psi(X_1)}{\tilde{Z}} + \frac{\varphi(X_2)\psi(X_2)\psi(X_1)\nabla_\theta\psi(X_1)}{\tilde{Z}^3}. \tag{17}$$

However, this estimator is not directionally unbiased and does not achieve the convergence shown in this paper. This is due to the fact that it is *unbalanced* in the sense that the exponent of $\tilde{Z}$ is different for each term. More specifically, taking expectation on (17) gives

$$-\frac{\langle \varphi, \nabla\psi_\theta\rangle}{\tilde{Z}} + \frac{\langle \varphi, \psi_\theta\rangle\langle \psi_\theta, \nabla\psi_\theta\rangle}{\tilde{Z}^3} \neq \lambda\nabla_\theta L(\theta), \quad \forall \lambda \in \mathbb{R}.$$

The detailed algorithm is summarized in Algorithm 2.

Algorithm 2 Stochastic gradient descent algorithm for supervised learning.

Preparation: φ_n; $\psi_\theta(x)$; number of iterations: M; learning rate η_m; initial parameter: θ_0; n
Running:
$m \leftarrow 0$;
while $m \le M$ **do**
  Sample $\{X_i^m\}_{i=1}^n$ independently from ρ;
  $\widetilde{Z}_m \leftarrow$ an estimation of $\|\psi_{\theta_m}\|_\rho$;
  $a_i \leftarrow -\|\psi_{\theta_m}\|_n^2 \varphi(X_i^m) + \langle \varphi, \psi_{\theta_m} \rangle_n \psi_{\theta_m}(X_i^m)$
  $G_m \leftarrow G(\theta, X) = \frac{1}{\widetilde{Z}_m^3} \frac{1}{n-1} \sum_{i=1}^n a_i \nabla_\theta \psi_{\theta_m}(X_i^m)$;
  $\theta_{m+1} \leftarrow \theta_m - \eta_m G_m$;
end while
Output: θ_m;

The choice of learning rate η_m in the pre-training setting and the decay rate of the loss function depends on the ratio $\|\psi_{\theta_m}\|_\rho / \widetilde{Z}$. However, for our results it suffices that this ratio is bounded above and below by fixed constants. As mentioned above we may estimate $\tilde{Z} = (\frac{1}{n} \sum_{i=1}^n |\psi_\theta(X_i)|^2)^{1/2}$ at each iteration using an independent sample $X_{n+1}, \dots, X_{2n} \perp\!\!\!\perp X_1, \dots, X_n$. Alternatively we may take inspiration from variance reduction techniques [40–42], to update $\tilde{Z}$ using a large batch once every K steps, letting $\tilde{Z}_m = (\frac{1}{K} \sum_{i=1}^K |\psi_\theta(X_i)|^2)^{1/2}$ when $m/K \equiv 0 \,(\mathrm{mod}\, K)$ and $\widetilde{Z}_m = \widetilde{Z}_{m-1}$ otherwise.

Our convergence bound for the supervised pre-training assumes uniform bounds similar to Condition 4.2.

Condition 4.8. There exists a constant C_ψ such that for any $\theta \in \mathbb{R}^d$

- (Value bound):

$$\left\| \psi_\theta^2 \right\|_\rho^{1/2} / \|\psi_\theta\|_\rho \le C_\psi \,. \tag{18}$$

- (Gradient bound):

$$\left\| \left| \nabla_\theta \psi_\theta \right|^2 \right\|_\rho^{1/2} / \|\psi_\theta\|_\rho \le C_\psi \,. \tag{19}$$

- (Hessian bound):

$$\left\| \left\| \mathsf{H}_\theta \psi_\theta \right\|_2^2 \right\|_\rho^{1/2} / \|\psi_\theta\|_\rho \le C_\psi \,, \tag{20}$$

  where $\mathsf{H}_\theta \psi$ is the hessian of ψ.

Under Condition 4.8, we establish the Lipschitz property of $\nabla_\theta L$ and provide bounds for the variance of the gradient estimator (see Lemma D.1). These results play a crucial role in demonstrating the convergence of our method. Condition 4.8 is scale-invariant in ψ_θ, meaning that the same condition holds with the same constant for any $\lambda \psi_\theta$, where $\lambda \neq 0$.

Theorem 4.9. *Assume Condition 4.8, $|\varphi_n| \le C_\varphi$, and*

$$\frac{1}{C_r} \le \frac{\|\psi_{\theta_m}\|_\rho}{\widetilde{Z}_m} \le C_r \,, \quad \text{a.s.} \tag{21}$$

for all $m \in \mathbb{N}$, where θ_m comes from Algorithm 2. Then there exists a constant C that only depends on C_r, C_ψ, C_φ such that, when $\eta_m < \frac{1}{C}$ for any $m \ge 0$,

$$\sum_{m=0}^M \eta_m \mathbb{E}\left(\left| \nabla_\theta L(\theta_m) \right|^2 \right) \le 2 C_r^2 L(\theta_0) + C \sum_{m=0}^M \frac{\eta_m^2}{n} \,, \tag{22}$$

for all $M > 0$.

The above theorem is proved in Appendix D. In the proof, we first bound the $\mathbb{E}|G_m|^2$ and show that $\nabla_\theta L$ has a uniform Lipschitz constant. Then, the decay rate of the loss function in each step can be lower bounded by $|\nabla_\theta L|^2$, which finally gives us an upper bound of $|\nabla_\theta L|^2$ as shown in (22).

Using Theorem 4.9 it is straightforward to show the following corollary:

Corollary 4.10. *Under Condition 4.8 and (21),*

- *Choosing* $\eta_m = \Theta\left(n\epsilon^2\right)$ *and* $M = \Theta\left(1/(n\epsilon^4)\right)$*, we have*

$$\min_{0\le m\le M} \mathbb{E}\left(\left|\nabla_\theta L(\theta_m)\right|\right) \le \epsilon.$$

- *Choosing* $\eta_m = \Theta\left(\sqrt{\frac{n}{m+1}}\right)$*, we have*

$$\min_{0\le m\le M} \mathbb{E}\left(\left|\nabla_\theta L(\theta_m)\right|\right) = O\left((nM)^{-1/4}\right).$$

We note that this bound matches the standard $O(M^{-1/4})$ convergence rate of SGD to first-order stationary point for non-convex functions.

5. Numerical results

5.1. Pre-training with scale-invariant loss

To evaluate the use of a scale-invariant loss during pre-training we modify the training loss used in the pre-training stage of [19] to be scale-invariant with respect to the trained Ansatz. We tested the scale-invariant loss in a version of the FermiNet code [43] where the electron configurations were sampled from the target state as in [19]. Specifically, the wave function is given by a sum of determinants and the target state is a Slater determinant:

$$\psi_\theta(x) = \sum_{k=1}^{d}\left[\det(y_{ij}^{(k)})_{i,j=1}^{N}\right], \qquad \varphi(x) = \det\left[\left(\phi_j(x_i)\right)_{i,j=1}^{N}\right],$$

where N is the number of electrons and the tensor y is the output of the parameterized permutation-equivariant neural network. In our experiment we use the standard FermiNet Ansatz [11] for the network architecture. The pre-training of [19] uses the loss function

$$\sum_{k=1}^{d}\sum_{i,j=1}^{N} |y_{i,j}^{(k)} - \phi_j(x_i)|^2 \tag{23}$$

on the matrices. To test the use of a scale-invariant loss function during training we replace the training loss of Equation (23) with the sum of column-wise scale-invariant $\sin^2$ distances. We note that in practice we did not find it necessary to estimate the denominator using an independent sample as in Remark 4.6. Note also that we can view the fitting of a (column) vector-valued function $\Omega \ni x \mapsto (y_i)_{i=1}^N$ as fitting a scalar-valued function $y(x,i)$ defined on $\Omega \times \{1,\dots,N\}$.

$$L(\theta) = \sum_{k=1}^{d}\sum_{j=1}^{N}\left(1 - \frac{|y_{\cdot,j}^{(k)} \cdot \phi_j(\cdot)|^2}{\|y_{\cdot,j}^{(k)}\|^2}\right), \tag{24}$$

where $\phi_j(\cdot) = (\phi_j(x_1),\dots,\phi_j(x_n))^T$, that is,

$$L(\theta) = \sum_{k=1}^{d}\sum_{j=1}^{N}\left(1 - \frac{|\sum_{i=1}^{N} y_{i,j}^{(k)}\phi_j(x_i)|^2}{\sum_{i=1}^{N}|y_{i,j}^{(k)}|^2}\right). \tag{25}$$

To evaluate the performance of the modified pre-training procedure we plot the angle between the target Slater determinant-based wave function and the neural network wave function during each training procedure (Fig. 1). The system shown is the lithium atom (details in Appendix F).

5.2. Empirical convergence of VMC algorithm

As an example, we demonstrate the convergence of the VMC Algorithm 1 on the Hydrogen square (H_4) model depicted in Fig. 2. This system involves only four particles but is strongly correlated and known to be difficult to simulate accurately. We define ψ_θ using the FermiNet ansatz [11,35] and run 200,000 training steps using stochastic gradient descent with a learning rate schedule $\eta_m = \frac{0.05}{\sqrt{1+m/10000}}$.

With this learning rate schedule, Corollary 4.4 implies that the running minimum of $\mathbb{E}(|\nabla_\theta L(\theta_m)|)$ should converge as $O(M^{-1/4})$ as the optimization progresses. To verify this bound, we plot in Fig. 3 the running minimum of $|G_m|$, which is our numerical proxy for $|\nabla_\theta L(\theta)|$ as per Lemma 4.1. We compare two distinct VMC runs using 10 and 1000 walkers, respectively, to explore how the convergence varies with the accuracy of the gradient estimate. We estimate the convergence rate by measuring the overall slope of the log-log plot, starting from step 200 to avoid the initial pre-asymptotic period. We find that the gradient of the loss converges roughly as $M^{-0.27}$ when using 10 walkers, closely matching the expected bound. With 1000 walkers the convergence goes as $M^{-0.38}$, which is somewhat faster than the theoretical bound, as might be expected given that the estimate of the gradient is very accurate with so many walkers. With 1000 walkers, the convergence is $M^{-0.38}$. This rate surpasses the theoretical bound given in Corollary 4.4.

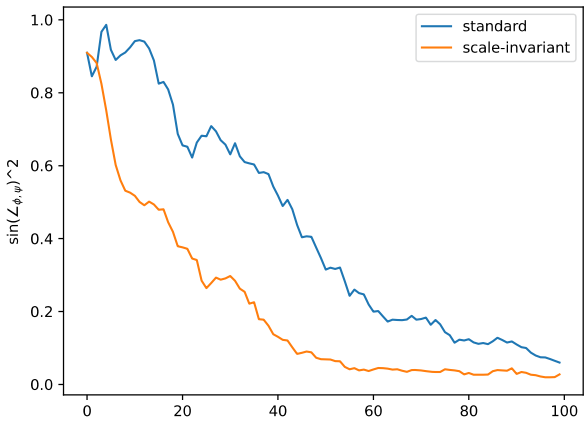


Fig. 1. Convergence of supervised pre-training using the scale-invariant training loss (orange) vs. the training loss of [19] (blue). For both optimizers the plotted quantity is the sine of the angle between the target state and the trained state, where the angle is defined in $\mathscr{L}^2(\mathbb{R}^{3n}, \rho = |\varphi|^2)$ with respect to the measure induced by the target state density.

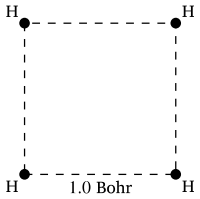


Fig. 2. Atomic configuration for the square H_4 model.

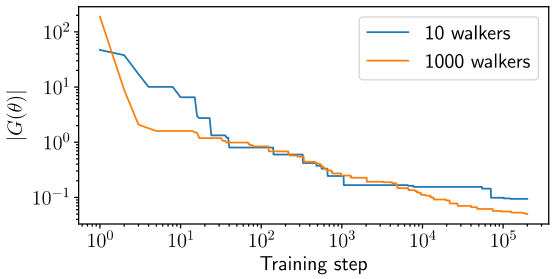


Fig. 3. Convergence of VMC run on the H_4 square. The running minimum is taken to smooth out the data and match the form of Corollary 4.4.

Although a rigorous theoretical understanding of this phenomenon is currently lacking, we hypothesize that with this number of walkers, relatively accurate gradient estimates lead to training dynamics similar to (non-stochastic) gradient descent, giving a faster convergence rate than our theoretical bound in Corollary 4.4.

Since the bounding of the Lipschitz constant of the gradient of the loss function is critical to our convergence proof, we supply in Fig. 4 a numerical estimate of this quantity during the VMC run. Here we show data only for the case of 1000 walkers, as the Lipschitz constant estimate is extremely noisy with 10 walkers. The data suggest that the value of the Lipschitz constant is well below 1000 for this simulation.

6. Conclusion

We consider two optimization problems that arise in neural network variational Monte Carlo simulations of electronic structure: energy minimization and supervised pre-training. We provide theoretical convergence bounds for both cases. For the setting of supervised pre-training, we note that the standard algorithms do not incorporate the property of scale-invariance. We propose using a scale-invariant loss function for the supervised pre-training and demonstrate numerically that incorporating scale-invariance accelerates the process of pre-training.

SGD is only the simplest stochastic optimization method. Over the last two decades, there has been increasing interest in developing more efficient and scalable optimization methods for VMC simulations [44–46], and our analysis may be a starting point for analyzing such methods. It may also be possible to generalize this work from a high-dimensional sphere to a Grassmann manifold,

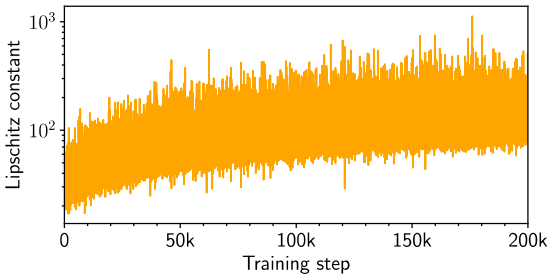


Fig. 4. Lipschitz constant for VMC run on the H_4 square with 1000 walkers. The constant is numerically approximated using the formula $|G(\theta_{m+1}) - G(\theta_m)|/|\theta_{m+1} - \theta_m|$.

parameterized by a neural network up to a gauge matrix. This setting could be applicable to VMC simulations of *excited states* of quantum systems.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CRediT authorship contribution statement

Nilin Abrahamsen: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Zhiyan Ding:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Gil Goldshlager:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Lin Lin:** Writing – review & editing, Writing – original draft, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Zhiyan Ding reports financial support was provided by US National Science Foundation (NSF) through grant number OMA-2016245. Gil Goldshlager reports financial support was provided by US Department of Energy under Award Number DE-SC0023112. Lin Lin reports financial support was provided by US Department of Energy under Award Number DE-SC0022364.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work was supported by the Simons Foundation under Award No. 825053 (N.A.), and by the Challenge Institute for Quantum Computation (CIQC) funded by National Science Foundation (NSF) through grant number OMA-2016245 (Z.D.). This work is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112 (G.G.). This material is also based upon work supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research and Office of Basic Energy Sciences, Scientific Discovery through Advanced Computing (SciDAC) program under Award Number DE-SC0022364 (L.L.). L.L. is a Simons Investigator in Mathematics. This research used the Savio computational cluster resource provided by the Berkeley Research Computing program at the University of California, Berkeley (supported by the UC Berkeley Chancellor, Vice Chancellor for Research, and Chief Information Officer).

Appendix A. Proofs of gradient estimators

Proof of (4). Let $q_\theta(x) = |\psi_\theta(x)|^2$, and $Q_\theta = \|\psi_\theta\|^2$.

We write (3) as

$$L(\theta) = \langle q_\theta, \mathcal{E}_\theta \rangle / Q_\theta \,.$$

Then by the product formula,

$$\nabla_\theta L(\theta) = \frac{\nabla_\theta \langle q_\theta, \mathcal{E}_\theta \rangle}{Q_\theta} - \frac{\langle q_\theta, \mathcal{E}_\theta \rangle \nabla_\theta Q_\theta}{Q_\theta^2} =: A - B. \tag{A.1}$$

Continuing, we have

$$\begin{aligned} A &= \langle \mathcal{E}_\theta, \nabla_\theta q_\theta \rangle / Q_\theta + \langle q_\theta, \nabla_\theta \mathcal{E}_\theta \rangle / Q_\theta \\ &= \langle \mathcal{E}_\theta, \frac{\nabla_\theta q_\theta}{q_\theta} \frac{q_\theta}{Q_\theta} \rangle + \langle \frac{q_\theta}{Q_\theta}, \nabla_\theta \mathcal{E}_\theta \rangle \\ &= \mathbb{E}_{p_\theta}[\mathcal{E}_\theta \nabla_\theta \log q_\theta] + \mathbb{E}_{p_\theta}[\nabla_\theta \mathcal{E}_\theta], \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} B &= \langle \frac{q_\theta}{Q_\theta}, \mathcal{E}_\theta \rangle \frac{\nabla_\theta Q_\theta}{Q_\theta} \\ &= L(\theta) \langle \frac{q_\theta}{Q_\theta}, \frac{\nabla_\theta q_\theta}{q_\theta} \rangle = L(\theta) \mathbb{E}_{p_\theta}[\nabla_\theta \log q_\theta]. \end{aligned} \tag{A.3}$$

Substitute (A.2), (A.3) into (A.1) to obtain

$$\begin{aligned} \nabla_\theta L(\theta) &= \mathbb{E}_{X \sim p_\theta} \left[(\mathcal{E}_\theta(X) - L(\theta)) \nabla_\theta \log q_\theta(X) + \nabla_\theta \mathcal{E}_\theta(X) \right] \\ &= 2\mathbb{E}_{X \sim p_\theta} \left[\left(\mathcal{E}_\theta(X) - L(\theta) \right) \nabla_\theta \log |\psi_\theta(X)| + \nabla_\theta \mathcal{E}_\theta(X) \right]. \end{aligned} \tag{A.4}$$

It remains to show that $\mathbb{E}_{X \sim p_\theta}[\partial_i \mathcal{E}_\theta(X)] = 0$ where ∂_i is differentiation with respect to θ_i. Write

$$\partial_i \mathcal{E}_\theta = \partial_i \frac{H\psi_\theta}{\psi_\theta} = \frac{\partial_i H\psi_\theta \cdot \psi_\theta - H\psi_\theta \cdot \partial_i \psi_\theta}{\psi_\theta^2}. \tag{A.5}$$

So, because H is symmetric:

$$\mathbb{E}_{X \sim p_\theta} \left[\partial_i \mathcal{E}_\theta \right] = \frac{\langle H \partial_i \psi_\theta, \psi_\theta \rangle - \langle H\psi_\theta, \partial_i \psi_\theta \rangle}{\|\psi_\theta\|^2} = 0, \quad \square$$

Proof of Lemma 4.1.

$$\begin{aligned} G(\theta, X) =& \frac{2}{n-1} \sum_{i=1}^{n} \mathcal{E}_\theta(X_i) \nabla_\theta \log |\psi_\theta(X_i)| \\ & - \frac{2}{n(n-1)} \sum_{j=1}^{n} \sum_{i=1}^{n} \mathcal{E}_\theta(X_j) \nabla_\theta \log |\psi_\theta(X_i)| \\ =& \frac{2}{n} \sum_{i=1}^{n} \mathcal{E}_\theta(X_i) \nabla_\theta \log |\psi_\theta(X_i)| \\ & - \frac{2}{n^2 - n} \sum_{i \neq j} \mathcal{E}_\theta(X_i) \nabla_\theta \log |\psi_\theta(X_j)|. \end{aligned}$$

Here, the terms $i = j$ in the expansion of the rightmost term cancel with the first sum. But now each sum is the average of terms with the correct expectation $\left(2\mathbb{E}_{X \sim p_\theta} \left[\mathcal{E}_\theta(X) \nabla_\theta \log |\psi_\theta(X)| \right] \right.$ and $\left. 2\mathcal{L}_\theta \mathbb{E}_{X \sim p_\theta} \nabla_\theta \log |\psi_\theta(X)| \right.$ respectively). $\square$

Appendix B. The directionally unbiased gradient estimator for supervised learning

Given $n \geq 2$ i.i.d. samples $\{X_i\} \sim \rho$. We write $\psi_i = \psi_\theta(X_i)$, $\varphi_i = \varphi(X_i)$, and $\nabla \psi_i = \nabla_\theta \psi_\theta(X_i)$.

$$\langle \psi, \varphi \rangle_n = \frac{1}{n} \sum_{i=1}^{n} \psi_i \varphi_i, \qquad \|\psi\|_n^2 = \frac{1}{n} \sum_{i=1}^{n} \psi_i^2. \tag{B.1}$$

Lemma B.1. *Given samples $X_1, \ldots, X_n$ from ρ, let*

$$G_n = \frac{1}{n-1} \sum_{j=1}^{n} a_j \nabla \psi_j, \tag{B.2}$$

$$a_j = -\|\psi\|_n^2 \varphi_j + \langle \varphi, \psi \rangle_n \psi_j. \tag{B.3}$$

Then G_n is an unbiased estimator for $G = \|\psi_\theta\|_\rho^3 \nabla_\theta L(\theta)$.

Proof of Lemma B.1. We notice

$$G = -\|\psi_\theta\|_\rho^2 \langle \varphi, \nabla \psi_\theta \rangle + \langle \varphi, \psi_\theta \rangle \langle \psi_\theta, \nabla \psi_\theta \rangle.$$

For each $i \neq j$ we have an unbiased estimator for G given by

$$-|\psi_i|^2 \varphi_j \nabla \psi_j + \varphi_i \psi_i \psi_j \nabla \psi_j.$$

Taking the average over all pairs $i \neq j$ gives another unbiased estimator for G:

$$-\frac{1}{n(n-1)} \sum_{i \neq j} (\psi_i^2 \varphi_j \nabla \psi_j - \varphi_i \psi_i \psi_j \nabla \psi_j).$$

We may add the terms $i = j$ without changing the value because all such terms evaluate to 0. □

Appendix C. Proof of Theorem 4.3

To prove Theorem 4.3, we first show the following lemma:

Lemma C.1. *Under Condition 4.2 there exists a uniform constant C' such that for any $\theta, \widetilde{\theta} \in \mathbb{R}^d$,*

$$\left|\nabla_\theta L(\theta)\right| \leq 4C_\psi^2, \tag{C.1}$$

$$\left|\nabla_\theta L(\theta) - \nabla_\theta L\left(\widetilde{\theta}\right)\right| \leq C'(C_\psi^4 + 1)|\theta - \widetilde{\theta}|, \tag{C.2}$$

and

$$\mathbb{E}_{\{X_i\}_{i=1}^n}\left(|G(\theta, X)|^2\right) \leq |\nabla_\theta L(\theta)|^2 + \frac{C'(C_\psi^4 + 1)}{n}, \tag{C.3}$$

where $G(\theta, X)$ is defined in (10).

The above lemma is important for the proof of Theorem 4.3. First, inequality (C.3) gives an upper bound for the gradient estimator. Using this upper bound, we can show that each iteration of Algorithm 1 is close to the classical gradient descent when the learning rate η is small enough. Second, (C.2) means that the hessian of L is bounded. This ensures that the classical gradient descent has a faster convergence rate to the first-order stationary point, which further implies the fast convergence of Algorithm 1.

Proof of Lemma C.1. Define

$$Z_\theta = \sqrt{\int_\Omega |\psi_\theta(x)|^2 \, dx}.$$

First, to prove (C.1), using (4), we have

$$\begin{aligned}\left|\nabla_\theta L(\theta)\right| &= 2\mathbb{E}_{X \sim p_\theta}\left|\left(\mathcal{E}_\theta(X) - L(\theta)\right) \frac{\nabla_\theta \psi_\theta(X)}{\psi_\theta(X)}\right| \\ &\leq 2\left(\mathbb{E}_{X \sim p_\theta}\left|\mathcal{E}_\theta(X) - L(\theta)\right|^2\right)^{1/2} \left(\mathbb{E}_{X \sim p_\theta}\left|\frac{\nabla_\theta \psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)^{1/2} \leq 4C_\psi^2,\end{aligned}$$

where we use Hölder's inequality in the first inequality, (11) and the second inequality of (12) in the last inequality.

Next, to prove (C.2). Using the formula of $\mathcal{E}(\theta)$ (equation (2)), we write

$$\begin{aligned}\|\mathsf{H}_\theta L(\theta)\| = &\underbrace{\frac{2\left|\nabla_\theta Z_\theta\right|}{Z_\theta^3} \int_\Omega \left|H\psi_\theta(x) \nabla_\theta \psi_\theta(x)\right| dx}_{\text{(I)}} + \underbrace{\frac{1}{Z_\theta^2} \int_\Omega \left\|\nabla_\theta H\psi_\theta(x) \nabla_\theta \psi_\theta(x)\right\| dx}_{\text{(II)}} \\ &+ \underbrace{\frac{1}{Z_\theta^2} \int_\Omega \left|H\psi_\theta(x)\right| \left\|\mathsf{H}_\theta \psi_\theta(x)\right\| dx}_{\text{(III)}} + \left\|\nabla_\theta \left(\frac{1}{Z_\theta^2} \int_\Omega L(\theta) \psi_\theta(x) \nabla_\theta \psi_\theta(x) dx\right)\right\|.\end{aligned}$$

We first deal with Terms (I), (II), (III) separately:

- For Term (I), we notice

$$\left|\nabla_\theta Z_\theta\right| = \left|\frac{\int \nabla_\theta \psi_\theta(x)\psi_\theta(x)dx}{\sqrt{\int |\psi_\theta(x)|^2 dx}}\right| \le \sqrt{\int |\nabla_\theta \psi_\theta(x)|^2 dx},$$

  where we use Hölder's inequality to bound the numerator in the inequality. This implies

$$\begin{aligned}
&\text{Term (I)} \le \frac{2\sqrt{\int |\nabla_\theta \psi_\theta(x)|^2 dx}}{\sqrt{\int |\psi_\theta(x)|^2 dx}} \int_\Omega \frac{\left|H\psi_\theta(x)\nabla_\theta\psi_\theta(x)\right|}{Z_\theta^2} dx \\
\le& 2\left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \frac{\sqrt{\int_\Omega |H\psi_\theta(x)|^2 dx}\sqrt{\int_\Omega |\nabla_\theta\psi_\theta(x)|^2 dx}}{Z_\theta^2} \\
=& 2\left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{H\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \\
\le& 2C_\psi^3 .
\end{aligned}$$

  Here, we use the Hölder's inequality in the second inequality, and (11)-(13) in the last inequality.
- For Term (II),

$$\begin{aligned}
&\frac{1}{Z_\theta^2}\int_\Omega \left\|\nabla_\theta H\psi_\theta(x)\nabla_\theta\psi_\theta(x)\right\| dx \le \mathbb{E}_{X\sim p_\theta}\left(\frac{\left|\nabla_\theta H\psi_\theta(X)\right|\left|\nabla_\theta\psi_\theta(X)\right|}{\left|\psi_\theta(X)\right|^2}\right) \\
\le& \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta H\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \le C_\psi^2
\end{aligned}$$

  where we use Hölder's inequality, (11)-(13) in the last two inequalities.
- For Term (III),

$$\begin{aligned}
&\frac{1}{Z_\theta^2}\int_\Omega \left\|H\psi_\theta(x)\mathsf{H}_\theta\psi_\theta(x)\right\| dx \le \mathbb{E}_{X\sim p_\theta}\left(\frac{\left|H\psi_\theta(X)\right|\left\|\mathsf{H}_\theta\psi_\theta(x)\right\|}{\left|\psi_\theta(X)\right|^2}\right) \\
\le& \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{H\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\mathsf{H}_\theta\psi_\theta(x)}{\psi_\theta(X)}\right|^2\right)\right)^{1/2} \le C_\psi^2
\end{aligned}$$

  where we use Hölder's inequality, (11)-(13) in the last two inequalities.

Combining the above three inequalities, we have

$$\text{Term (I)} + \text{Term (II)} + \text{Term (III)} \le 2C_\psi^2(C_\psi + 1).$$

Using a similar calculation, we can also show

$$\left\|\nabla_\theta\left(\frac{1}{Z_\theta^2}\int_\Omega L(\theta)\psi_\theta(x)\nabla_\theta\psi_\theta(x)dx\right)\right\| \le C(C_\psi^4 + 1),$$

where C is a uniform constant. Thus, we have the hessian of L can be bounded, meaning

$$\left\|\mathsf{H}_\theta L(\theta)\right\| \le C(C_\psi^4 + 1),$$

which proves (C.2) by the mean-value theorem.

Finally, to prove (C.3), noticing

$$G(\theta, X) = \frac{2}{n}\sum_{i=1}^n \mathcal{E}_\theta(X_i)\nabla_\theta \log|\psi_\theta(X_i)| - \frac{2}{n^2-n}\sum_{i\ne j}\mathcal{E}_\theta(X_i)\nabla_\theta\log|\psi_\theta(X_j)|,$$

and

$$\mathbb{E}_{\{X_i\}_{i=1}^n}(G(\theta, X)) = \nabla_\theta L(\theta),$$

we have

$$\begin{aligned}
&\mathbb{E}_{\{X_i\}_{i=1}^n}\left(|G(\theta,X)|^2\right)\\
\leq&|\nabla_\theta L(\theta)|^2+\frac{8}{n^2}\sum_{i=1}^n\mathbb{E}_{X_i\sim p_\theta}\left(\left|\mathcal{E}_\theta\left(X_i\right)\nabla_\theta\log|\psi_\theta\left(X_i\right)|-\mathbb{E}_{X\sim p_\theta}\left[\mathcal{E}_\theta(X)\nabla_\theta\log|\psi_\theta(X)|\right]\right|^2\right)\\
&+\frac{8}{(n^2-n)^2}\mathbb{E}\left(\sum_{i\neq j}\mathcal{E}_\theta\left(X_{i_1}\right)\nabla_\theta\log|\psi_\theta\left(X_{j_1}\right)|-\mathcal{L}_\theta\mathbb{E}_{X\sim p_\theta}\nabla_\theta\log|\psi_\theta(X)|\right)^2\\
\leq&|\nabla_\theta L(\theta)|^2+\frac{8}{n}\mathbb{E}_{X\sim p_\theta}\left|\mathcal{E}_\theta\left(X\right)\nabla_\theta\log|\psi_\theta\left(X\right)|-\mathbb{E}_{X\sim p_\theta}\left[\mathcal{E}_\theta(X)\nabla_\theta\log|\psi_\theta(X)|\right]\right|^2\\
&+\frac{8}{(n^2-n)^2}\left(\sum_{i_1=i_2\text{ or }j_1=j_2}1\right)\mathbb{E}_{(X_1,X_2)\sim p_\theta^{\otimes 2}}\left|\mathcal{E}_\theta\left(X_1\right)\nabla_\theta\log|\psi_\theta\left(X_2\right)|-\mathcal{L}_\theta\mathbb{E}_{X\sim p_\theta}\nabla_\theta\log|\psi_\theta(X)|\right|^2\\
\leq&|\nabla_\theta L(\theta)|^2+\frac{8}{n}\mathbb{E}_{X\sim p_\theta}\left|\mathcal{E}_\theta\left(X\right)\nabla_\theta\log|\psi_\theta\left(X\right)|\right|^2+\frac{C}{n}\mathbb{E}_{(X_1,X_2)\sim p_\theta^{\otimes 2}}\left|\mathcal{E}_\theta\left(X_1\right)\nabla_\theta\log|\psi_\theta\left(X_2\right)|\right|^2
\end{aligned},$$

where we use X_i and X_j are independent in the first two inequalities. Using (11) and the second inequality of (12), it is straightforward to show:

$$\mathbb{E}_{X\sim p_\theta}\left|\mathcal{E}_\theta\left(X\right)\nabla_\theta\log|\psi_\theta\left(X\right)|\right|^2\leq\left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{H\psi_\theta(X)}{\psi_\theta(X)}\right|^4\right)\right)^{1/2}\left(\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^4\right)\right)^{1/2}\leq C_\psi^4,$$

and

$$\mathbb{E}_{(X_1,X_2)\sim p_\theta^{\otimes 2}}\left|\mathcal{E}_\theta\left(X_1\right)\nabla_\theta\log|\psi_\theta\left(X_2\right)|\right|^2\leq\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{H\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\mathbb{E}_{X\sim p_\theta}\left(\left|\frac{\nabla_\theta\psi_\theta(X)}{\psi_\theta(X)}\right|^2\right)\leq C_\psi^4.$$

This concludes the proof of (C.3). □

Now, we are ready to prove Theorem 4.3:

Proof of Theorem 4.3. Denote the probability filtration

$$\mathcal{F}_m=\sigma(X_i^j,1\leq i\leq n,1\leq j\leq m).$$

According to the algorithm, we have

$$\theta_{m+1}=\theta_m-\eta_m G_m.$$

Plugging θ_{m+1} into $L(\theta)$, we have

$$L(\theta_{m+1})\leq L(\theta_m)-\eta_m\nabla_\theta L(\theta_m)\cdot G_m+\frac{C\eta_m^2}{2}|G_m|^2$$

where we use (C.2) and C is a constant that only depends on C_ψ. This implies

$$\mathbb{E}\left(L(\theta_{m+1})|\mathcal{F}_{m-1}\right)\leq L(\theta_m)-\eta_m\left|\nabla_\theta L(\theta_m)\right|^2+\frac{C\eta_m^2}{2}\mathbb{E}(|G_m|^2|\mathcal{F}_{m-1})$$

where we use $\mathbb{E}_{\{X_i\}_{i=1}^n}(G(\theta,X))=\nabla_\theta L(\theta)$. Furthermore, using (C.3) and $\eta_m<\frac{1}{C}$, we have

$$\begin{aligned}
\mathbb{E}\left(L(\theta_{m+1})|\mathcal{F}_{m-1}\right)\leq&L(\theta_m)-\left(\eta_m-C\eta_m^2/2\right)\left|\nabla_\theta L(\theta_m)\right|^2+\frac{C\eta_m^2}{2n}\\
\leq&L(\theta_m)-\frac{\eta_m}{2}\left|\nabla_\theta L(\theta_m)\right|^2+\frac{C\eta_m^2}{2n}
\end{aligned}$$

Taking the average in $m=1,2,3,\cdots,M-1$, we prove (14). □

Appendix D. Proof of Theorem 4.9

In this section, we prove Theorem 4.9. We first show the following lemma:

Lemma D.1. *Under Condition 4.8 there exists a constant C' that only depends on C_ψ,C_r,C_ϕ such that for any $\theta,\widetilde{\theta}\in\mathbb{R}^d$,*

$$\left|\nabla_\theta L(\theta)-\nabla_\theta L\left(\widetilde{\theta}\right)\right|\leq C'|\theta-\widetilde{\theta}|,\tag{D.1}$$

and

$$\mathbb{E}_{\{X_i\}_{i=1}^n}\left(|G(\theta,X)|^2\right) \le \frac{Z_\theta^6}{Z^6}\left(|\nabla_\theta L(\theta)|^2 + \frac{C'}{n}\right), \tag{D.2}$$

where $G(\theta,X)$ *is defined in* (16).

Proof of Lemma D.1. The proof is very similar to the proof of Lemma C.1 after setting $H\psi_\theta(x) = g(x)\left(\langle g(x), \psi_\theta(x)\rangle\right)$. □

Now, we are ready to prove Theorem 4.9.

Proof of Theorem 4.9. According to the algorithm, we have

$$\theta_{m+1} = \theta_m - \eta_m G_m .$$

Plugging θ_{m+1} into $L(\theta)$, we have

$$L(\theta_{m+1}) \le L(\theta_m) - \eta_m \nabla_\theta L(\theta_m)\cdot G_m + \frac{C\eta_m^2}{2}|G_m|^2$$

where we use (D.1). Here C is a constant that only depends on C_ψ. Because

$$\mathbb{E}_{X_1^m,\cdots,X_n^m}(G_m) = \frac{Z_m^3}{\widetilde{Z}_m^3}\nabla_\theta L(\theta_m), \tag{D.3}$$

we obtain

$$\begin{aligned}\mathbb{E}\left(L(\theta_{m+1})|\mathcal{L}_{m-1}\right) \le & L(\theta_m) - \eta_m\left|\nabla_\theta L(\theta_m)\right|^2\frac{Z_m^3}{\widetilde{Z}_m^3} \\ & + \frac{C\eta_m^2}{2}\mathbb{E}(|G_m|^2|\mathcal{L}_{m-1})\end{aligned} \tag{D.4}$$

Using (D.2) and (21),

$$\mathbb{E}(|G_m|^2|\mathcal{L}_{m-1}) \le C\left(|\nabla_\theta L(\theta_m)|^2 + \frac{C'}{n}\right), \tag{D.5}$$

where C is a constant depends on C_ψ, C_r, C_ϕ.

Plugging (D.5) into (D.4), using (21), and choosing η_m small enough, we have

$$\mathbb{E}\left(L(\theta_{m+1})\right) \le L(\theta_m) - \frac{\eta_m}{2C_r^2}\left|\nabla_\theta L(\theta_m)\right|^2 + \frac{C\eta_m^2}{n} \tag{D.6}$$

Taking the average in $m = 1,2,3,\cdots,M-1$, we prove (22). □

Appendix E. Lipschitz of the Z_θ

Lemma E.1. *Assume Condition 4.8, let* $\theta,\widetilde{\theta}\in\mathbb{R}^d$*, and suppose*

$$\left|\widetilde{\theta}-\theta\right| \le C_r .$$

Then there exists a constant C' *that only depends on* C_ψ, C_r, C_ϕ *such that*

$$\frac{\|\psi_{\widetilde{\theta}}\|_\rho}{\|\psi_\theta\|_\rho} \le \exp(C'C_r). \tag{E.1}$$

Proof of Lemma E.1. Fixing $\theta,\widetilde{\theta}$, we define

$$R(t) = \frac{\left\|\psi_{\theta+t(\widetilde{\theta}-\theta)}\right\|_\rho^2}{\|\psi_\theta\|_\rho^2}.$$

Then

$$
\begin{aligned}
|R'(t)| &\le \left| \frac{2\left\langle \nabla_\theta \Psi_{\theta+t(\widetilde{\theta}-\theta)}, \Psi_{\theta+t(\widetilde{\theta}-\theta)} \right\rangle_\rho}{\|\psi_\theta\|_\rho^2} \right| |\widetilde{\theta}-\theta| \\
&\le \left| \frac{2\left\langle \nabla_\theta \Psi_{\theta+t(\widetilde{\theta}-\theta)}, \Psi_{\theta+t(\widetilde{\theta}-\theta)} \right\rangle_\rho}{\left\|\Psi_{\theta+t(\widetilde{\theta}-\theta)}\right\|_\rho^2} \right| \left| \frac{\left\|\Psi_{\theta+t(\widetilde{\theta}-\theta)}\right\|_\rho^2}{\|\psi_\theta\|_\rho^2} \right| |\widetilde{\theta}-\theta| \\
&\le C|\widetilde{\theta}-\theta| R(t),
\end{aligned}
$$

where we use (18) and (19) in the last inequality. Because $R(0)=1$, using Grönwall's inequality, we obtain

$$
\frac{\|\psi_{\widetilde{\theta}}\|_\rho}{\|\psi_\theta\|_\rho} = R^{1/2}(1) \le \exp(C|\theta-\widetilde{\theta}|) \le \exp(CC_r). \quad \square
$$

Appendix F. Additional details on scale-invariant supervised training

We created a fork [47] of the FermiNet repository [43] and implemented the scale-invariant loss function Equation (24) in the columns of the backflow tensors. We used the straight-through gradient estimator [48] to train the network using either the standard loss or our modified scale-invariant loss while plotting the angle between the wave functions in each case Fig. 1.

The commands for each (using the fork [47] of FermiNet) were:

- Standard (blue)

```
ferminet --config ferminet/configs/atom.py \
--config.system.atom Li \
--config.batch_size 256 \
--config.pretrain.iterations 10
```

- Scale-invariant (orange)

```
ferminet --config ferminet/configs/atom.py \
--config.system.atom Li \
--config.batch_size 256 \
--config.pretrain.iterations 10 \
--config.pretrain.SI
```

References

[1] W.M.C. Foulkes, L. Mitas, R.J. Needs, G. Rajagopal, Quantum Monte Carlo simulations of solids, Rev. Mod. Phys. 73 (2001) 33–83, https://doi.org/10.1103/RevModPhys.73.33.

[2] J. Toulouse, R. Assaraf, C.J. Umrigar, Introduction to the variational and diffusion Monte Carlo methods, in: P.E. Hoggan, T. Ozdogan (Eds.), Electron Correlation in Molecules – Ab Initio Beyond Gaussian Quantum Chemistry, in: Advances in Quantum Chemistry, vol. 73, Academic Press, 2016, pp. 285–314, Chapter fifteen, https://www.sciencedirect.com/science/article/pii/S0065327615000386.

[3] F. Becca, S. Sorella, Quantum Monte Carlo Approaches for Correlated Systems, Cambridge University Press, 2017.

[4] S. Sorella, Green function Monte Carlo with stochastic reconfiguration, Phys. Rev. Lett. 80 (1998) 4558–4561, https://doi.org/10.1103/PhysRevLett.80.4558, https://link.aps.org/doi/10.1103/PhysRevLett.80.4558.

[5] S. Sorella, Generalized Lanczos algorithm for variational quantum Monte Carlo, Phys. Rev. B 64 (2001) 024512, https://doi.org/10.1103/PhysRevB.64.024512, https://link.aps.org/doi/10.1103/PhysRevB.64.024512.

[6] M.P. Nightingale, V. Melik-Alaverdian, Optimization of ground- and excited-state wave functions and van der Waals clusters, Phys. Rev. Lett. 87 (2001) 043401, https://doi.org/10.1103/PhysRevLett.87.043401, https://link.aps.org/doi/10.1103/PhysRevLett.87.043401.

[7] J. Toulouse, C.J. Umrigar, Optimization of quantum Monte Carlo wave functions by energy minimization, J. Chem. Phys. 126 (8) (2007) 084102.

[8] G. Carleo, M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science 355 (6325) (2017) 602–606, https://doi.org/10.1126/science.aag2302.

[9] J. Han, L. Zhang, W. E, Solving many-electron Schrödinger equation using deep neural networks, J. Comput. Phys. 399 (2019) 108929, https://doi.org/10.1016/j.jcp.2019.108929.

[10] J. Hermann, Z. Schätzle, F. Noé, Deep-neural-network solution of the electronic Schrödinger equation, Nat. Chem. 12 (10) (2020) 891–897, https://doi.org/10.1038/s41557-020-0544-y.

[11] D. Pfau, J.S. Spencer, A.G. Matthews, W.M.C. Foulkes, Ab initio solution of the many-electron Schrödinger equation with deep neural networks, Phys. Rev. Res. 2 (3) (2020) 033429.

[12] J. Hermann, J. Spencer, K. Choo, A. Mezzacapo, W.M.C. Foulkes, D. Pfau, G. Carleo, F. Noé, Ab-initio quantum chemistry with neural-network wavefunctions, https://doi.org/10.48550/ARXIV.2208.12590, https://arxiv.org/abs/2208.12590, 2022.

[13] K. Choo, A. Mezzacapo, G. Carleo, Fermionic neural-network states for ab-initio electronic structure, Nat. Commun. 11 (1) (2020) 2368, https://doi.org/10.1038/s41467-020-15724-9.

[14] T.D. Barrett, A. Malyshev, A. Lvovsky, Autoregressive neural-network wavefunctions for ab initio quantum chemistry, Nat. Mach. Intell. 4 (4) (2022) 351–358.

[15] G. Cassella, H. Sutterud, S. Azadi, N.D. Drummond, D. Pfau, J.S. Spencer, W.M.C. Foulkes, Discovering quantum phase transitions with fermionic neural networks, https://doi.org/10.48550/ARXIV.2202.05183, https://arxiv.org/abs/2202.05183, 2022.

[16] G. Pescia, J. Nys, J. Kim, A. Lovato, G. Carleo, Message-passing neural quantum states for the homogeneous electron gas, arXiv:2305.07240, 2023.

[17] M. Wilson, S. Moroni, M. Holzmann, N. Gao, F. Wudarski, T. Vegge, A. Bhowmik, Neural network ansatz for periodic wave functions and the homogeneous electron gas, Phys. Rev. B 107 (23) (2023) 235139.
[18] X. Li, Z. Li, J. Chen, Ab initio calculation of real solids via neural network ansatz, Nat. Commun. 13 (1) (2022) 7895.
[19] I. von Glehn, J.S. Spencer, D. Pfau, A self-attention ansatz for ab-initio quantum chemistry, https://doi.org/10.48550/ARXIV.2211.13672, https://arxiv.org/abs/2211.13672, 2022.
[20] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, 2015.
[21] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.
[22] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv:1607.06450, 2016.
[23] Q. Meng, S. Zheng, H. Zhang, W. Chen, Z.-M. Ma, T.-Y. Liu, G-SGD: optimizing reLU neural networks in its positively scale-invariant space, in: International Conference on Learning Representations, 2019.
[24] T. van Laarhoven, L2 regularization versus batch and weight normalization, arXiv:1706.05350, 2017.
[25] S. Arora, Z. Li, K. Lyu, Theoretical analysis of auto rate-tuning by batch normalization, in: International Conference on Learning Representations, 2019.
[26] Z. Li, S. Bhojanapalli, M. Zaheer, S. Reddi, S. Kumar, Robust training of neural networks using scale invariant architectures, in: Proceedings of the 39th International Conference on Machine Learning, PMLR, 2022.
[27] R. Wan, Z. Zhu, X. Zhang, J. Sun, Spherical motion dynamics: learning dynamics of normalized neural network using sgd and weight decay, in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 6380–6391.
[28] T. Salimans, D.P. Kingma, Weight normalization: a simple reparameterization to accelerate training of deep neural networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016.
[29] M. Kodryan, E. Lobacheva, M. Nakhodnov, D.P. Vetrov, Training scale-invariant neural networks on the sphere can happen in three regimes, in: Advances in Neural Information Processing Systems, 2022.
[30] S. Bonnabel, Stochastic gradient descent on Riemannian manifolds, IEEE Trans. Autom. Control 58 (9) (2013) 2217–2229, https://doi.org/10.1109/TAC.2013.2254619.
[31] H. Zhang, S.J. Reddi, S. Sra, Riemannian svrg: fast stochastic optimization on Riemannian manifolds, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 4599–4607.
[32] H. Sato, H. Kasai, B. Mishra, Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport, SIAM J. Optim. 29 (2) (2019) 1444–1472.
[33] N. Tripuraneni, N. Flammarion, F. Bach, M.I. Jordan, Averaging stochastic gradient descent on Riemannian manifolds, in: Proceedings of the 31st Conference on Learning Theory, PMLR, 2018, pp. 650–687, ISSN: 2640-3498, https://proceedings.mlr.press/v75/tripuraneni18a.html.
[34] T. Li, F. Chen, H. Chen, Z. Wen, Provable convergence of variational Monte Carlo methods, arXiv:2303.10599, 2023.
[35] J.S. Spencer, D. Pfau, A. Botev, W.M.C. Foulkes, Better, faster fermionic neural networks, https://doi.org/10.48550/ARXIV.2011.07125, https://arxiv.org/abs/2011.07125, 2020.
[36] L. Gerard, M. Scherbela, P. Marquetand, P. Grohs, Gold-standard solutions to the Schrödinger equation using deep learning: how much physics do we need?, https://doi.org/10.48550/ARXIV.2205.09438, https://arxiv.org/abs/2205.09438, 2022.
[37] A. Szabo, N. Ostlund, Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory, Dover Books on Chemistry, Dover Publications, 1996, https://books.google.com/books?id=6mV9gYzEkgIC.
[38] A. Khaled, P. Richtárik, Better theory for sgd in the nonconvex world, arXiv:2002.03329, 2020.
[39] G. Garrigos, R.M. Gower, Handbook of convergence theorems for (stochastic) gradient methods, arXiv:2301.11235, 2024.
[40] M. Schmidt, N. Roux, F. Bach, Minimizing finite sums with the stochastic average gradient, Math. Program. 162 (2013), https://doi.org/10.1007/s10107-016-1030-6.
[41] A. Defazio, F. Bach, S. Lacoste-Julien, SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives, Adv. Neural Inf. Process. Syst. 2 (2014).
[42] R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, pp. 315–323.
[43] D.P. James, S. Spencer, F. Contributors, FermiNet (forked from commit fdaca31) (2023). URL, http://github.com/deepmind/ferminet.
[44] A.W. Sandvik, G. Vidal, Variational quantum Monte Carlo simulations with tensor-network states, Phys. Rev. Lett. 99 (2007) 220602, https://doi.org/10.1103/PhysRevLett.99.220602, https://link.aps.org/doi/10.1103/PhysRevLett.99.220602.
[45] E. Neuscamman, C.J. Umrigar, G.K.-L. Chan, Optimizing large parameter sets in variational quantum Monte Carlo, Phys. Rev. B 85 (2012) 045103, https://doi.org/10.1103/PhysRevB.85.045103, https://link.aps.org/doi/10.1103/PhysRevB.85.045103.
[46] L. Otis, E. Neuscamman, Complementary first and second derivative methods for ansatz optimization in variational Monte Carlo, Phys. Chem. Chem. Phys. 21 (2019) 14491–14510, https://doi.org/10.1039/C9CP02269D.
[47] D.P. James, S. Spencer, F. Contributors, Fork of FermiNet (forked from commit fdaca31), https://github.com/nilin/fork_of_ferminet/tree/sample_orbitals, 2023.
[48] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, J. Xin, Understanding straight-through estimator in training activation quantized neural nets, arXiv:1903.05662, 2019.