Accepted to J. Chem. Phys. 10.1063/5.0157512



# DeepQMC: an open-source software suite for variational optimization of deep-learning molecular wave functions.

Z. Schätzle [1, a)], P. B. Szabó [1, a)], M. Mezera [1], J. Hermann [2], and F. Noé [1, 2, 3, 4]

[1)] *FU Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany*

[2)] *Microsoft Research AI4Science, Karl-Liebknecht Str. 32, 10178 Berlin, Germany*

[3)] *FU Berlin, Department of Physics, Arnimallee 14, 14195 Berlin, Germany*

[4)] *Rice University, Department of Chemistry, Houston, TX 77005, USA*

(Dated: 17 July 2023)

Computing accurate yet efficient approximations to the solutions of the electronic Schrödinger equation has been a paramount challenge of computational chemistry for decades. Quantum Monte Carlo methods are a promising avenue of development as their core algorithm exhibits a number of favorable properties: it is highly parallel, and scales favorably with the considered system size, with an accuracy that is limited only by the choice of the wave function ansatz. The recently introduced machine-learned parametrizations of quantum Monte Carlo ansatzes rely on the efficiency of neural networks as universal function approximators to achieve state of the art accuracy on a variety of molecular systems. With interest in the field growing rapidly, there is a clear need for easy to use, modular, and extendable software libraries facilitating the development and adoption of this new class of methods. In this contribution, the DEEPQMC program package is introduced, in an attempt to provide a common framework for future investigations by unifying many of the currently available deep-learning quantum Monte Carlo architectures. Furthermore, the manuscript provides a brief introduction to the methodology of variational quantum Monte Carlo in real space, highlights some technical challenges of optimizing neural network wave functions, and presents example black-box applications of the program package. We thereby intend to make this novel field accessible to a broader class of practitioners both from the quantum chemistry as well as the machine learning communities.

[a)] Z. Schätzle and P. B. Szabó contributed equally to this work.

Accepted to J. Chem. Phys. 10.1063/5.0157512



## I. INTRODUCTION

Recently, the application of machine learning to a wide range of problems from the natural sciences has proven to be highly successful. Computational chemistry is a field of particular activity: machine learning force fields model complicated quantum mechanical effects at the resolution of atoms, while machine learned functionals elevate density functional theory to unprecedented accuracy [1–3]. These approaches utilize supervised training to learn from accurate quantum mechanical reference calculations, and make predictions for unseen configurations. While this results in fast yet accurate approximations to the quantum many-body problem, it inherently depends on high quality training data, which represents a major bottleneck of these methods.

An orthogonal way to incorporate machine learning into computational chemistry is its application to improve ab-initio calculations. Notably, over the course of the last years a new family of deep-learning quantum Monte Carlo (deep QMC) methods has developed, incorporating advancements from the field of machine learning [4]. Common to the deep QMC methods is the utilization of neural networks to parametrize highly expressive ansatzes, efficiently approximating the solutions of the time-independent electronic Schrödinger equation, thereby providing a complete description of the system's electronic properties. Originating from spin lattices[5], deep-learning ansatzes were soon applied to molecules in real-space[6]. With the development of PauliNet[7] and FermiNet[8], the accuracy of neural-network wave functions became the state of the art within variational Monte Carlo. Subsequent works have further increased the accuracy of these ansatzes[9–12], extended them to the simulation of excited states[13] as well as periodic systems[14,15], combined them with pseudo-potentials[16], used them in the calculation of interatomic forces[17], utilized them in diffusion Monte Carlo simulations[18,19], and extended them to share parameters across multiple molecular geometries[20–22] or distinct molecules[23,24].

Although the method of optimizing deep-learning wave function ansatzes using variational quantum Monte Carlo was developed only a few years ago, it already competes with some of the most accurate traditional quantum chemistry methods on molecules with up to $\sim$100 electrons. Exhibiting competitive scaling with the number of electrons, it has the potential to be extended to larger systems in the near future. Achieving this will no doubt require further method development as well as efficient implementations of the core algorithms, creating the need for open source libraries that facilitate experimentation and contribution from the community.

Accompanying the above summarized research, various software libraries for variational op-

Accepted to J. Chem. Phys. 10.1063/5.0157512



timization of deep-learning wave functions have been released[25–28]. While NETKET[25] provides a general implementation of variational optimization of machine learning wave functions mainly for lattice systems with recent extensions to continuous space, the research for molecular machine learning wave functions was carried out across various repositories and is lacking a unified framework. The presented DEEPQMC program package aims to provide a unified implementation of the developments in the field of deep-learning molecular wave functions. It intends to be easy to use out of the box, while allowing full control and flexibility over the ansatz and variational training for advanced users. The library is designed to be modular, facilitating the rapid development and testing of individual components, and easing the implementation of new features. It makes use of the composable function transformations and just-in-time compilation provided by the JAX library[29] to express performant GPU accelerated algorithms using concise Python[30] syntax. Neural network models are encapsulated in higher-level objects, using the `haiku` deep-learning library[31]. The project is open-source and distributed online under the MIT license[26].

## II. THEORY

### A. The electronic structure problem

In computational chemistry, molecular systems are often described by the non-relativistic molecular Hamiltonian using the Born–Oppenheimer approximation:

$$\hat{H} = \sum_{i=1}^{N} \Big( -\frac{1}{2}\Delta_{\mathbf{r}_i} - \sum_{I=1}^{M} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} + \sum_{j=1}^{i-1} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \Big), \tag{1}$$

where $\mathbf{r}_i$ denotes the position of the $i$th electron, while $Z_I$ and $\mathbf{R}_I$ are the charge and position of the $I$th nucleus, respectively. To determine the electronic structure of these molecular systems, one must solve the associated time independent Schrödinger equation

$$\hat{H}\psi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = E\psi(\mathbf{x}_1, \ldots, \mathbf{x}_N), \tag{2}$$

where $\mathbf{x}_i = (\mathbf{r}_i, \sigma_i)$ comprise the positions of the electrons and their spin. A solution is an eigenfunction of the Hamiltonian, the electronic wave function $\psi$, and its corresponding energy eigenvalue $E$. With the exact wave function at hand, any observable electronic property of the system can in principle be computed, as the wave function gives a complete description of the system's electronic state. Since electrons have half-integer spin, their wave functions must be antisymmetric

Accepted to J. Chem. Phys. 10.1063/5.0157512



with respect to electron exchanges

$$\begin{aligned}\psi(\ldots,\mathbf{x}_i,\ldots,\mathbf{x}_j,\ldots) = \\ -\psi(\ldots,\mathbf{x}_j,\ldots,\mathbf{x}_i,\ldots)\,.\end{aligned} \tag{3}$$

While general wave functions are complex-valued, the solutions of the time independent Schrödinger equation can be chosen as real without loss of generality, due to the hermiticity of the molecular Hamiltonian. Therefore in all of the following discussions, as well as in the DEEPQMC library, only real valued wave functions are considered.

## B. Variational optimization

Even with the aforementioned approximations, the electronic Schrödinger equation involving the molecular Hamiltonian can only be solved analytically for hydrogenic atoms – the special case of the two-body problem. This makes computational quantum chemistry a mainly numerical field, where different methods yield approximate solutions at varying trade-offs of accuracy and computational cost. The class of variational quantum chemistry methods phrases the solution of the Schrödinger equation as a minimization problem. The ground state of the Hamiltonian is approximated by optimizing the parameters $\boldsymbol{\theta}$ of a trial wave function (ansatz) $\psi_{\boldsymbol{\theta}}$, to minimize the expectation value of the Hamiltonian

$$\boldsymbol{\theta}' = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \langle \hat{H} \rangle_{\psi_{\boldsymbol{\theta}}}. \tag{4}$$

This objective is rooted in the variational principle of quantum mechanics, which states that the ground state energy of the Hamiltonian is a lower bound for the energy expectation value of any wave function from the associated antisymmetric Hilbert space $H^-$

$$E_0 \leq \min_{\psi} \langle \hat{H} \rangle_{\psi} \quad \psi \in H^-. \tag{5}$$

The variational methods can be categorized based on the means of calculating the expectation value $\langle \cdot \rangle$, and choice of ansatz $\psi_{\boldsymbol{\theta}}$.

The DEEPQMC program package implements VMC in real space (first quantization) with neural network wave functions. In VMC, expectation values are estimated through a stochastic

Accepted to J. Chem. Phys. 10.1063/5.0157512



sampling of electron configurations

$$\begin{aligned}\langle \hat{H} \rangle_{\psi_\theta} &= \frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \\ &= \frac{\int d\mathbf{r}_1, ..., d\mathbf{r}_N \psi^*_\theta(\mathbf{r}_1, ..., \mathbf{r}_N) \hat{H} \psi_\theta(\mathbf{r}_1, ..., \mathbf{r}_N)}{\int d\mathbf{r}_1, ..., d\mathbf{r}_N |\psi_\theta(\mathbf{r}_1, ..., \mathbf{r}_N)|^2} \\ &= \frac{\int d\mathbf{r}_1, ..., d\mathbf{r}_N |\psi_\theta(\mathbf{r}_1, ..., \mathbf{r}_N)|^2 E_{\text{loc}}[\psi_\theta](\mathbf{r})}{\int d\mathbf{r}_1, ..., d\mathbf{r}_N |\psi_\theta(\mathbf{r}_1, ..., \mathbf{r}_N)|^2} \\ &= \mathbb{E}_{\mathbf{r} \sim |\psi_\theta|^2} \left[ E_{\text{loc}}[\psi_\theta](\mathbf{r}) \right] \\ &\approx \frac{1}{n} \sum_{\mathbf{r} \sim |\psi_\theta|^2}^{n} E_{\text{loc}}[\psi_\theta](\mathbf{r}) \,.\end{aligned} \tag{6}$$

Because the molecular Hamiltonian does not depend on the spin, it is possible to compute the energy using the spatial wave function $\psi(\mathbf{r}_1, ..., \mathbf{r}_N)$, where fixed spins are assigned to the electrons and spin-up and spin-down electrons are treated as distinguishable[32]. The convention is to sort the electrons by spin and consider the first $N^\uparrow$ electrons to have spin-up and the remaining $N^\downarrow = N - N^\uparrow$ electrons to possess spin-down.

In practice, a VMC simulation then consists of choosing an ansatz (see Section III), and optimizing it in an alternating scheme of sampling and parameter updates. The expectation value in (6) is approximated by sampling the probability density given by the square of the wave function (see Section V), followed by a parameter update using the gradient of the expectation value (see Section IV).

### C. Neural network wave functions

Being exact in principle, the choice of the wave function ansatz is crucial for the efficiency of a VMC simulation. Recently, neural network parametrizations of real-space molecular wave functions were introduced by PauliNet[7], and FermiNet[8]. They both rely on generalized Slater determinants, that augment the single particle orbitals of conventional Slater determinants with many-body correlation,

$$\psi_\theta(\mathbf{r}_1, ..., \mathbf{r}_N) = \textstyle\sum_p c_p \det[\mathbf{A}^p(\mathbf{r})] \,, \tag{7}$$

$$A^p_{ik} = \phi^p_k(\mathbf{r}_i, \{\mathbf{r}^\uparrow\}, \{\mathbf{r}^\downarrow\}) \times \varphi^p_k(\mathbf{r}_i) \,. \tag{8}$$

Here, $\phi^p_k$ are many-body orbitals, and $\varphi^p_k$ are single particle envelopes that ensure the correct asymptotic behavior of the wave function with increasing distance from the nuclei. The set notation $\{\cdot\}$ is to be understood as a permutation invariant dependence on the spin-up electrons $\mathbf{r}^\uparrow$,

Accepted to J. Chem. Phys. 10.1063/5.0157512



and spin-down electrons $\mathbf{r}^{\downarrow}$, respectively. The ansatz may be a linear combination of multiple generalized Slater determinants, which are distinguished with the $p$ index. The form of $\phi_k^p$ in (8) is closely related to the backflow transformation[33], which introduces quasi-particles to obtain many-body orbital functions. The key observation motivating this augmentation is that the anti-symmetry of Slater determinants constructed from many-body orbitals is preserved as long as the orbital functions are equivariant with the exchange of electrons,

$$P_{ij}^{\parallel}\phi_k(\mathbf{r}_i, \{\mathbf{r}^{\uparrow}\}, \{\mathbf{r}^{\downarrow}\}) = \phi_k(\mathbf{r}_j, \{\mathbf{r}^{\uparrow}\}, \{\mathbf{r}^{\downarrow}\}), \tag{9}$$

where $P_{ij}^{\parallel}$ is the operator exchanging same-spin electrons $i$ and $j$.

Most of the currently used deep-learning molecular wave functions[7,8,11] share the functional form of (7), and (8), and differ only in the parametrization of the many-body orbitals $\phi_k^p$ and single-particle envelopes $\varphi_k^p$. DEEPQMC aims to provide a general framework for variational optimization of deep-learning molecular wave functions, facilitating the investigation of the design space spanned by the PauliNet, FermiNet, and DeepErwin neural network ansatzes.

### D. Pseudopotentials

Despite the favorable asymptotic scaling of VMC with the number of electrons, systems containing heavy nuclei remain challenging due to a variety of reasons. The high energy of electrons near these nuclei complicates simulations by spoiling the optimization and reducing the effectiveness of MCMC sampling. Furthermore, the kinetic energy of these electrons reaches the relativistic regime, requiring the treatment of relativistic effects that are not accounted for in the standard non-relativistic molecular Hamiltonian of (1). On the other hand, while the core regions of heavily charged nuclei contribute dominantly to the total energy, they are typically unchanged during chemical processes and thus have little effect on the chemically relevant relative energies. Therefore most quantum chemistry methods targeted at computing relative energies reduce the above outlined difficulties, by treating the outer (valence) electrons separately from the inner (core) electrons.

The approach most suited for implementation in the context of variational optimization of deep-learning wave functions is the use of pseudopotentials, which has been previously demonstrated by Li et al.[16]. In this method, the core electrons are excluded from the explicit calculation and replaced by additional terms in the Hamiltonian, to simulate their influence on the remaining $N_{\mathrm{v}}$

The Journal of Chemical Physics

AIP Publishing

valence electrons. The modified Hamiltonian reads as

$$\hat{H} = \sum_{i=1}^{N_\mathrm{v}} \left( -\frac{1}{2}\Delta_{\mathbf{r}_i} + \sum_{j=1}^{i-1} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \right) + \hat{V}_\mathrm{PP} \,. \tag{10}$$

The $\hat{V}_\mathrm{PP}$ pseudopotential term is in turn decomposed to local and non-local parts

$$\hat{V}_\mathrm{PP} = \sum_{I=1}^{M} \sum_{i=1}^{N_\mathrm{v}} \left( V^I(r_{iI}) + \sum_{l=0}^{l_\mathrm{max}} W_l^I(r_{iI}) \hat{P}_l^{iI} \right) , \tag{11}$$

where $r_{iI} = |\mathbf{r}_i - \mathbf{R}_I|$, $V^I$ and $W_l^I$ are sets of scalar functions describing the local and non-local pseudopotential contributions, while $\hat{P}_l^{iI} = \sum_{m=-l}^{m=l} |lm\rangle_{iI} \langle lm|_{iI}$ is a projection operator of the $i$-th electron on spherical harmonics centered on the $I$-th nucleus. To evaluate the contribution of the non-local part of the pseudopotential (second term of (11)) one considers integrals of the form

$$\begin{aligned} \frac{\langle \mathbf{r}| W_l^I \hat{P}_l^{iI} |\psi\rangle}{\langle \mathbf{r}|\psi\rangle} &= W_l^I(r_{iI}) \sum_{m=-l}^{l} Y_{lm}(\Omega_{iI}) \\ &\times \int Y_{lm}(\Omega'_{iI})^* \frac{\psi(\mathbf{r}_1,\ldots,(r_{iI},\Omega'_{iI}),\ldots,\mathbf{r}_N)}{\psi(\mathbf{r}_1,\ldots,(r_{iI},\Omega_{iI}),\ldots,\mathbf{r}_N)} \, \mathrm{d}\Omega'_{iI} \end{aligned} \tag{12}$$

where $Y_{lm}$ is a spherical harmonic and $(r_{iI}, \Omega_{iI})$ denotes the position vector of the $i$-th electron $\mathbf{r}_i$, expressed in spherical coordinates centered on nucleus $I$. Following the first implementation of pseudopotentials for deep-learning molecular wave functions by Li et al.[16], the above integral is approximated using an icosahedral quadrature of 12-points.

The scalar functions $V^I$ and $W_l^I$ are typically pre-computed by expanding them in a Gaussian basis, and fitting the expansion parameters directly to a database of reference energies. The DEEPQMC program package currently includes the widely used BFD[34], and the most recent ccECP[35] pseudopotentials, with an application of the latter presented in Section VIII B.

## III. WAVE FUNCTION DESIGN SPACE

DEEPQMC implements a variety of options to obtain the equivariant many-body orbitals $\phi_k^p$ and the accompanying envelopes $\varphi_k^p$, covering PauliNet, FermiNet, DeepErwin and their derivatives. In the following, the main architectural concepts of these wave function ansatzes will be described in more detail. For ease of use DEEPQMC provides predefined configuration files to obtain the above mentioned ansatzes, while allowing their interpolation through a manual choice of hyperparameters.

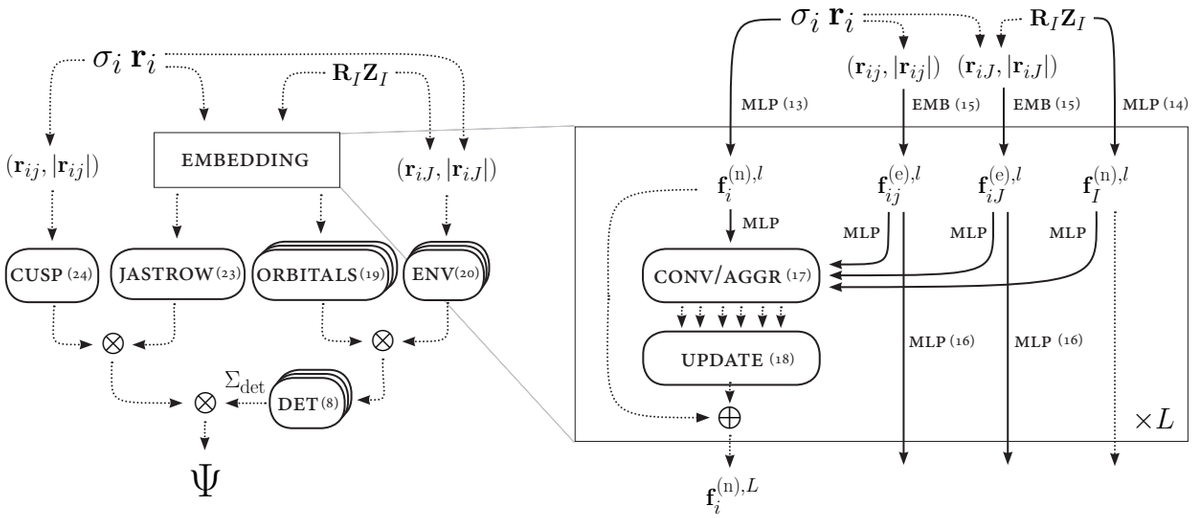


FIG. 1. **Sketch of a general neural network wave function ansatz and its graph neural network architecture.** This sketch comprises the implemented design space for the neural network wave function (left) and the GNN architecture (right). Solid lines can carry MLPs and dotted lines correspond to forwarding without further change. Numbers in parentheses refer to the corresponding equations of the main text. The choice among the drawn connections, the depths of the associated MLPs as well as the aggregation and update rules distinguish the previously published works PauliNet, FermiNet, and DeepErwin.

## A. Graph Neural Networks

Central to the neural network wave function ansatzes is the computation of equivariant many-body embedding vectors for the electrons, which are used downstream to obtain the entries of the generalized Slater determinant. Many strategies of obtaining these embeddings can be unified in the framework of graph neural networks (GNNs). GNNs are well suited to model functions on graphs that exhibit certain permutational symmetries, and can be adapted to describe electrons of molecules in real-space.

An electronic configuration of a molecule can be encoded as a graph, where the nodes are electrons and nuclei, and the connecting edges carry pairwise features, e.g. difference vectors. GNNs are functions of these graphs, yielding high-dimensional latent space embeddings of the nodes. The electronic node embeddings are initialized with single-electron features and iteratively updated to incorporate many-body information of the electronic environment. Using graph convolutions, the updates are invariant under the exchange of electrons in the environment, and the

conditions of (9) are fulfilled.

The most relevant aspects of the GNN architecture implemented in DEEPQMC are sketched on the right pane of Figure 1, and are discussed in detail in the following. Electron positions (spins) are denoted with $\mathbf{r}$ ($\sigma$), while $\mathbf{R}$ and $Z$ indicate nuclei positions and charges. Node and edge quantities are denoted with the superscripts (n) and (e), respectively. Furthermore, $l$ indexes the GNN interaction layers, $\boldsymbol{\theta}$ denotes functions parametrized by MLPs, and $t$ runs over the different edge types (those between electrons of identical or opposite spins, or between electrons and nuclei), node types (electron or nuclei nodes) or message types. Lastly, vertical brackets indicate the different options implemented in DEEPQMC for the computation of various quantities.

***The graph representation:***

A graph is a natural way to denote the electronic configuration of a molecule in real-space. The nodes of the graph represent particles (electrons and nuclei), carrying information such as spin or nuclear charge. The edges support the difference vectors between the particles, resulting in a representation invariant under global translation. Note that using internal coordinates that are invariant under global rotation may not be sufficient to represent all wave functions (simple counterexamples are atomic wave functions with P symmetry), and can only be employed with a careful treatment of the spatial symmetries.

To implement a variety of wave function ansatzes, DEEPQMC provides configuration options to define the specifics of the graph construction outlined above. Most importantly, the nodes corresponding to nuclei, and their respective nuclei-electron edges can optionally be excluded from the graph. In this case, electron-nuclei information can still be introduced to the GNN, by initializing the electron embeddings using a concatenation of the difference vectors between the positions of the given electron and all nuclei (see the second case of (13)).

***Node features:***

The output of DEEPQMC GNNs are electron node embeddings $\mathbf{f}_i^{(\mathrm{n})}$ which are subsequently used to generate the many-body orbitals that constitute the entries of the generalized Slater determinants and an optional trainable Jastrow factor. To enforce the equivariance of these quantities with respect to the exchange of electrons, the initialization of the electron embeddings has to be

Accepted to J. Chem. Phys. 10.1063/5.0157512





The Journal of Chemical Physics
AIP Publishing

chosen appropriately. In DEEPQMC one can either use identical embeddings for all electrons of the same spin (invariant under permutation of same-spin electrons), or a concatenation of the electron nuclei difference vectors (equivariant under electron permutations)

$$\mathbf{f}_i^{(\mathrm{n}),0,\mathrm{el}} = \begin{cases} \mathscr{E}_{\boldsymbol{\theta}}^{(\mathrm{n}),\mathrm{el}}(\sigma_i) \\ \mathscr{E}_{\boldsymbol{\theta}}^{(\mathrm{n}),\mathrm{el}}(\mathbf{r}_i - \mathbf{R_1}, \ldots, \mathbf{r}_i - \mathbf{R_N}) \,, \end{cases} \tag{13}$$

where $\mathscr{E}_{\boldsymbol{\theta}}^{(\mathrm{n}),\mathrm{el}}$ are parameterized node embedding functions implemented through MLPs.

If the GNN is chosen to explicitly consider electron-nuclei interactions, the embeddings associated with the nuclear nodes have to be initialized besides the electronic embeddings. DEEPQMC implements fixed nuclear node embeddings $\mathbf{f}_I^{(\mathrm{n})}$, that either distinguish all nuclei or depend on the respective atom type:

$$\mathbf{f}_I^{(\mathrm{n}),0,\mathrm{nuc}} = \begin{cases} \mathscr{E}_{\boldsymbol{\theta}}^{(\mathrm{n}),\mathrm{nuc}}(I) \\ \mathscr{E}_{\boldsymbol{\theta}}^{(\mathrm{n}),\mathrm{nuc}}(Z_I) \,. \end{cases} \tag{14}$$

***Edge features :***

The edges of the graph hold the pairwise differences of node positions ($\mathbf{r}_{ij}$) and their embeddings are consequently initialized as

$$\mathbf{f}_{ij}^{(\mathrm{e}),0,t^{(\mathrm{e})}} = \mathscr{E}^{(\mathrm{e}),t^{(\mathrm{e})}}(\mathbf{r}_{ij}) \,, \tag{15}$$

where $\mathscr{E}^{(\mathrm{e}),t^{(\mathrm{e})}}$ is an edge type dependent featurization based on the pairwise differences. This may correspond to directly feeding the difference vectors, using the pairwise distances, expanding in a basis of Gaussians or working with rescaled difference vectors amongst other options. In later interaction layers, the original edge embeddings are either reused or iteratively updated,

$$\mathbf{f}_{ij}^{(\mathrm{e}),l,t^{(\mathrm{e})}} = \begin{cases} \mathbf{f}_{ij}^{(\mathrm{e}),0,t^{(\mathrm{e})}} \\ u_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}(\mathbf{f}_{ij}^{(\mathrm{e}),l-1,t^{(\mathrm{e})}}) \,, \end{cases} \tag{16}$$

with the latter option making use of a parametrized update function $u_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}$, thus increasing the effective depth of the architecture at the cost of additional MLPs. The parameters of the update function $u_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}$ may be shared across different edge types.

Accepted to J. Chem. Phys. 10.1063/5.0157512



***Message generation:***

The electron embeddings are updated in each interaction layer by aggregating messages passed along the graph edges. These messages are constructed via an elementwise product between functions of the sending node and edge embeddings (graph convolution),

$$\mathbf{m}_i^{l,t^{(\mathrm{m})}} = \sum_{j\in t^{(\mathrm{n})}} w_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}(\mathbf{f}_{ij}^{(\mathrm{e}),l,t^{(\mathrm{e})}}) \cdot h_{\boldsymbol{\theta}}^{l,t^{(\mathrm{n})}}(\mathbf{f}_j^{(\mathrm{n}),l,t^{(\mathrm{n})}}) \,. \tag{17}$$

The superscript $t^{(\mathrm{n})}$ on the node features specifies the subset of sending nodes and the superscript $t^{(\mathrm{e})}$ on the edge features depends on the type of the sending and the receiving nodes respectively. The choice of how to distinguish electron-electron messages based on their spin (relative spin of sending and receiving electron, spin of sending electron, or no distinction between messages from spin-up and spin-down electrons) is another hyperparameter of the GNN. Optionally, the above sum over the edges can be normalized by dividing it with the number of edges. Note that messages depending only on node (edge) information can be obtained by setting the function $w_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}$ ($h_{\boldsymbol{\theta}}^{l,t^{(\mathrm{n})}}$) to return identity. The superscript $t^{(\mathrm{m})}$ runs over all the constructed messages, which may include different choices of $w_{\boldsymbol{\theta}}^{l,t^{(\mathrm{e})}}$ and $h_{\boldsymbol{\theta}}^{l,t^{(\mathrm{n})}}$.

***Electron embedding update:***

To obtain updated electron embeddings, messages from various edge types are combined and added to a residual connection. DEEPQMC implements a few protocols for the combination of messages, that can be summarized as follows

$$\mathbf{f}_i^{(\mathrm{n}),l+1,\mathrm{el}} = \mathbf{f}_i^{(\mathrm{n}),l,\mathrm{el}} + \begin{cases} \sum_{t^{(\mathrm{m})}} g_{\boldsymbol{\theta}}^{l,t^{(\mathrm{m})}}\big(\mathbf{m}_i^{l,t^{(\mathrm{m})}}\big) \\ g_{\boldsymbol{\theta}}^{l}\big(\sum_{t^{(\mathrm{m})}} \mathbf{m}_i^{l,t^{(\mathrm{m})}}\big) \\ g_{\boldsymbol{\theta}}^{l}\big(\bigoplus_{t^{(\mathrm{m})}} \mathbf{m}_i^{l,t^{(\mathrm{m})}}\big)\,, \end{cases} \tag{18}$$

where $\bigoplus$ refers to the concatenation of the messages. Additionally to the messages constructed according to equation (17) the message types $t^{(\mathrm{m})}$ can include a residual connection $\mathbf{f}_i^{(\mathrm{n}),l}$ such that the trainable self-interaction of FermiNet and DeepErwin can be reproduced.

In the above outlined general GNN framework a wide variety of ansatzes can be obtained. Furthermore, the implementation of DEEPQMC and its GNN framework focus on facilitating rapid extensions with new ansatz variants either by exploration within the existing hyperparameter space or by extending it with new features.

Accepted to J. Chem. Phys. 10.1063/5.0157512



### B. Orbital construction

The entries of the generalized Slater determinant in (8) are obtained as products of many-body orbitals $\phi_k^p$ and envelopes $\varphi_k^p$. The many-body orbitals are functions of the final equivariant electron embeddings

$$\phi_k^p(\mathbf{r}_i, \{\mathbf{r}^\uparrow\}, \{\mathbf{r}^\downarrow\}) = \kappa_{\boldsymbol{\theta}}(\mathbf{f}_i^{(\mathrm{n}),L}), \tag{19}$$

where $\kappa_{\boldsymbol{\theta}}$ is an MLP applied electronwise, projecting the embedding dimension to the required number of orbitals. For the $\varphi_k^p$ envelopes, DEEPQMC implements linear combinations of exponentials centered on the nuclei

$$\varphi_k^p(\mathbf{r}_i) = \textstyle\sum_I \sum_{\beta_I} \omega_{k\beta_I}^p \exp\left(-||\Sigma_{k\beta_I}^p(\mathbf{r}_i - \mathbf{R}_I)||^\alpha\right), \tag{20}$$

where $\omega_{k\beta_I}^p$ and $\Sigma_{k\beta_I}^p$ are trainable parameters and $\beta_I$ indexes the basis function centered on atom $I$. The hyperparameter $\alpha \in (1,2)$ represents the choice of Slater type orbitals with $\alpha = 1$, and Gaussian type orbitals (GTOs) with $\alpha = 2$. DEEPQMC provides an option to restrict the envelopes to be isotropic ($\Sigma_{k\beta_I}^p := \sigma_{k\beta_I}^p \cdot \mathbf{I}$). The GTOs can be initialized from the molecular orbital coefficients of reference solutions with standard quantum chemistry basis sets obtained in PySCF[36].

### C. Determinant construction

Because the antisymmetry of the wave function is required with respect to the exchange of same-spin particles only, Slater determinants in VMC are typically considered block diagonal and are factored into a spin-up and a spin-down component

$$\psi_{\boldsymbol{\theta}} = \textstyle\sum_p c_p \det\left[\mathbf{A}^{\uparrow,p}(\mathbf{r})\right] \det\left[\mathbf{A}^{\downarrow,p}(\mathbf{r})\right]. \tag{21}$$

Additionally, DEEPQMC implements the full determinant option explored by Lin et al.[10], which constructs a single determinant using both spin-up and spin-down electrons

$$\psi_{\boldsymbol{\theta}} = \textstyle\sum_p c_p \det\left[\mathbf{A}^{\uparrow\downarrow,p}(\mathbf{r})\right]. \tag{22}$$

It's noted that since the many-body orbitals are not equivariant under the exchange of opposite spin electrons, the full determinant ansatz is still not antisymmetric with respect to these permutations. Instead, a full determinant can practically be understood as an expansion in multiple

Accepted to J. Chem. Phys. 10.1063/5.0157512



spin-factorized determinants, e.g. by relying on the generalized Laplace expansion of determinants to expand $\det\left[\mathbf{A}^{\uparrow\downarrow,p}(\mathbf{r})\right]$ according to the rows corresponding to spin-up electrons

$$\det\left[\mathbf{A}^{\uparrow\downarrow}(\mathbf{r})\right] = \sum_S \varepsilon_S \det\left[\mathbf{A}^{\uparrow,S}(\mathbf{r})\right] \det\left[\mathbf{A}^{\downarrow,\bar{S}}(\mathbf{r})\right]. \tag{23}$$

Here, $S$ runs over all subsets of the orbitals that contain as many elements as the number of spin-up electrons, $\bar{S}$ stands for the complement subset of $S$, $\mathbf{A}^{\uparrow,S}(\mathbf{r})$ denotes the submatrix of $\mathbf{A}^{\uparrow\downarrow}(\mathbf{r})$ formed from the orbitals in $S$ and the spin-up electrons, while $\varepsilon_S$ is the sign of the permutation defined by the subset $S$. For the block diagonal matrices of (21), the determinants for all subsets of spin-up orbitals containing off-diagonal elements evaluate to zero and the sum in (23) reduces to a single product of a spin-up and spin-down determinant. Note that since the many-body orbitals defined in (19) are not equivariant with respect to exchanges of electrons with opposite spins, the terms on the right hand side of (23) with different $S$s will in general be unrelated. In practice, it is conceivable that due to the concrete form of parametrization of the many-body orbitals, there remains some structure in the set of factorized determinants, that makes the full determinant more effective than using an equivalent number of spin-factorized determinants formed from independent orbitals.

### D. Jastrow factor and cusp correction

The antisymmetry of the wave function is retained when multiplying it with a global correction term symmetric under the exchange of the same spin particles. This symmetric correction, traditionally called a Jastrow factor, is well suited to introduce known asymptotics to the ansatz. DEEPQMC implements a learnable Jastrow factor $\mathrm{e}^J$, where $J$ is computed from the permutation invariant sum of many-body electron embeddings

$$J = \eta_\theta\Big(\sum_i \mathbf{f}_i^{(\mathrm{n}),L}\Big), \tag{24}$$

with $\eta_\theta$ again being implemented by an MLP. Furthermore, DEEPQMC provides a fixed Jastrow factor that implements the known asymptotic behavior [37] when two electrons approach

$$\gamma(\mathbf{r}) = \sum_{i<j} -\frac{\alpha c_{ij}}{1+\alpha|\mathbf{r}_i - \mathbf{r}_j|}, \tag{25}$$

where $c_{ij}$ is $\frac{1}{4}$ if the electrons $i$ and $j$ are of the same spin and $\frac{1}{2}$ if the electrons possess opposite spins and the hyperparameter alpha scales the width of the correction term. If cuspless Gaussian

envelopes are used, a similar cusp correction can be employed for the nuclei

$$\gamma(\mathbf{r},\mathbf{R}) = \sum_{i,I} \frac{\alpha Z_I}{1+\alpha|\mathbf{r}_i - \mathbf{R}_I|}\,, \tag{26}$$

serving as a simple replacement for the more involved technique utilized by Hermann et al.[7].

### E. Log-representation of the wave function

The output of the (unnormalized) neural network wave functions typically spans many orders of magnitude, potentially resulting in instabilities due to finite floating-point precision. In order to improve numerical stability, DEEPQMC represents wave functions in the log-domain

$$\psi = \big(\mathrm{sign}(\psi), \log(|\psi|)\big)\,. \tag{27}$$

We mitigate over- and underflow problems during the computation of the determinant by performing it directly in the log-domain using the appropriate `slogdet` function provided by JAX. In order to perform the summation over multiple determinants $\phi^p$ we apply the log-sum-exp trick

$$\log\Big(\big|\sum_p \phi^p\big|\Big) = \max\{\log(|\phi^p|)\} + \log\Big(\big|\sum_p \mathrm{sign}(\phi^p)\exp\big(\log(|\phi^p|) - \max\{\log(|\phi^p|)\}\big)\big|\Big)\,. \tag{28}$$

Note that for the variational principle to remain valid, it is sufficient to ensure the antisymmetry of the trial wave function, and its explicit sign is not needed for the evaluation of any of the quantities involved in the optimization (30).

## IV. TRAINING

In this section, some technical aspects of the variational optimization of deep-learning trial wave functions are discussed. While these ansatzes are trained within the standard VMC framework, the characteristics of their optimization differ markedly from other VMC ansatzes, mainly due to the greatly increased parameter count introduced by neural networks. On the other hand, it is also distinct from most other deep-learning settings owing to the unusual complexity of the loss function and the self-supervised setting, where training data is generated in parallel to the optimization.

### A. Loss function and gradient trick

As discussed in Section II B, VMC relies on the variational principle by optimizing the wave function ansatz to minimize the expectation value of the local energies. From a machine learning perspective, this translates to considering the loss function

$$\mathscr{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{r}\sim|\psi_{\boldsymbol{\theta}}|^2}\left[E_{\mathrm{loc}}[\psi_{\boldsymbol{\theta}}](\mathbf{r})\right]. \tag{29}$$

Naively computing the gradient of this loss would involve taking derivatives of the local energies $E_{\mathrm{loc}}[\psi_{\boldsymbol{\theta}}]$ with respect to the ansatz parameters $\boldsymbol{\theta}$. However, evaluating the local energy already involves second derivatives of the wave function with respect to the electron coordinates due to the Laplacian in (1). Consequently, this naive gradient computation would necessitate taking mixed third derivatives of the ansatz.

To reduce the computational costs and numerical instabilities associated with higher order derivatives, a different unbiased estimator of the loss gradient is utilized,

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}}\mathscr{L}(\boldsymbol{\theta}) = 2\mathbb{E}_{\mathbf{r}\sim|\psi_{\boldsymbol{\theta}}|^2}\left[\left(E_{\mathrm{loc}}[\psi_{\boldsymbol{\theta}}](\mathbf{r}) - \mathscr{L}(\boldsymbol{\theta})\right)\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log|\psi_{\boldsymbol{\theta}}|\right]. \tag{30}$$

The derivation of this estimator exploits the hermiticity of the Hamiltonian and is given in full detail by Inui et al.[38]. It replaces the derivatives of the local energy with a simple gradient of the wave function, therefore it is expected to be more efficient and numerically stable to evaluate than the direct gradient computation.

### B. Local energy evaluation

The evaluation of the local energies of the wave function ansatz is by far the most computationally demanding part of the training (and evaluation)

$$\begin{aligned} E_{\mathrm{loc}}[\psi_{\boldsymbol{\theta}}](\mathbf{r}) &= -\frac{1}{2}\sum_i\left(\frac{\Delta_{\mathbf{r}_i}\psi_{\boldsymbol{\theta}}(\mathbf{r})}{\psi_{\boldsymbol{\theta}}(\mathbf{r})}\right) + V(\mathbf{r}) \\ &= -\frac{1}{2}\sum_i\left(\Delta_{\mathbf{r}_i}\log|\psi_{\boldsymbol{\theta}}(\mathbf{r})| + \left(\nabla_{\mathbf{r}_i}\log|\psi_{\boldsymbol{\theta}}(\mathbf{r})|\right)^2\right) + V(\mathbf{r}). \end{aligned} \tag{31}$$

While the potential energy term is very cheap to evaluate, the cost of the Laplacian within the kinetic energy term scales steeply with the number of electrons. In this step, one obtains second derivatives of the wave function with respect to the electron coordinates. We obtain these derivatives of the wave function by applying automatic differentiation in backward-forward mode,

which we confirmed to be a good choice in the setting of molecular wave functions. Further discussions regarding the memory bottleneck associated with the Laplacian and details regarding the implementation choices are presented in the Appendix A.

### C. Pretraining

Choosing initial values for ansatz parameters is a non-trivial question common to many computational chemistry methods. One need only think of the sensitivity of the self-consistent iterations to the initial guess in Hartree–Fock (HF) and related methods[39–41]. The case of deep-learning VMC ansatzes is no different – a random initialization of the neural network parameters according to some of the widely adopted schemes of the machine learning community can lead to the optimization diverging or converging to local minima. This problem becomes increasingly severe with growing system size, presumably due to the higher-dimensional, more complex wave functions of larger molecules and their intricate nodal structure.

A practical solution to this issue is the initialization of the wave function based on a cheap reference solution. To that end DEEPQMC interfaces with PYSCF[36], enabling the initialization of wave functions from the coefficients of a preceding HF or multi-configurational self consistent field (MCSCF) calculation. While this allows the direct initialization of the neural network wave function ansatz as introduced by Hermann et al.[7], subsequent work suggested that explicitly incorporating an approximate reference wave function in the model can deteriorate performance[11]. Instead a short, supervised pretraining with respect to a reference solution before the self-supervised variational optimization is recommended. In this step, the many-body orbitals of the ansatz are trained to match the reference by minimizing the pretraining loss

$$\mathscr{L}_p(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{r}\sim|\psi_{\boldsymbol{\theta}}|^2}\Big[\sum_{ki}\big(\varphi_k^{\mathrm{ref}}(\mathbf{r}_i) - \phi_k(\mathbf{r}_i,\{\mathbf{r}^{\uparrow}\},\{\mathbf{r}^{\downarrow}\})\times\varphi_k(\mathbf{r}_i)\big)^2\Big]\,, \tag{32}$$

where $\varphi_k^{\mathrm{ref}}$ are the occupied orbitals of the HF/MCSCF wave function. Unlike the variational loss of (29), computing $\mathscr{L}_p$ does not involve evaluating the Laplacian of the ansatz, which means that pretraining requires significantly less computational resources than variational training. Initialization with pretrained orbitals, as introduced by Pfau et al.[8], improves the convergence properties of the variational training and, if well balanced with the subsequent variational optimization, can even slightly boost the final accuracy, as Gerard et al. recently demonstrated[11].

Accepted to J. Chem. Phys. 10.1063/5.0157512



### D. Gradient clipping

Despite the utilization of sophisticated gradient estimators and pretraining, the convergence of the variational optimization is still often hindered by outliers in the training batches of local energies. The existence of these outliers is not surprising, considering that the electrostatic energy is singular when two particles coincide, while the kinetic energy is singular at the nodes of the wave function – energy contributions that the shape of the wave function precisely levels out in later stages of the training. While the outliers are valid contributions to the energy expectation value, their presence can inject a lot of noise into the gradient estimates. To reduce their contribution to the parameter update, the loss and its gradient (given in (29) and (30)) are evaluated using clipped local energies $\hat{E}_{\mathrm{loc}}^{\mu,\sigma}$, where $\mu$ is the center and $\sigma$ is the half-width of the clipping window.

Regarding concrete choices for $\mu$ and $\sigma$, some empirical findings have been reported in the related literature. Investigating transition metal atoms using pseudopotentials, Li and coworkers report[16], that choosing $\sigma = 10 \times \mathrm{std}(E_{\mathrm{loc}})$ significantly outperforms all other options they've considered. More recently, von Glehn et al.[12] found that centering the clipping window at the median of local energies, and using the mean absolute deviation from the median to determine $\sigma$, improves the training of multiple deep-learning ansatzes. Considering the practical importance of the clipping mechanism, DEEPQMC implements the algorithm of von Glehn et al.[12], along with an analogous logarithmically scaling "soft" clipping scheme introduced by Hermann et al.[7], but also offers full flexibility to the user in specifying custom clipping functions.

Finally, it should be highlighted that the local energies are only to be clipped for computing the gradient of the loss during optimization. Since clipping can introduce a bias to the estimate of the energy expectation value, variational energy estimates can only be obtained from unclipped local energies.

### E. Optimizer

Utilizing natural gradient descent optimization[42] or Kronecker-factored approximations thereof[43] has proven to be a crucial ingredient to the success of variational optimization of deep-learning wave functions on molecular systems[8,11,12,44,45]. Consequently, DEEPQMC makes use of the Kronecker-Factored Approximate Curvature (KFAC) optimizer implementation of A. Botev and J. Martens[46]. To showcase the importance of the choice of the optimizer, the performance of

Accepted to J. Chem. Phys. 10.1063/5.0157512



KFAC is compared to the commonly employed first-order optimizer AdamW[47], on variational trainings on six test systems. The obtained training energy curves are plotted in Figure 2. To account for the 10–25% longer per iteration run time of KFAC compared to AdamW, the wall clock time of the training (instead of the usual iteration count) is shown on the horizontal axes. The results show that the slightly increased per-iteration cost is offset by the significantly improved per-iteration convergence speed of the KFAC optimizer. Furthermore, it is found that the increase in the relative cost of KFAC over AdamW optimization is smaller for systems with larger numbers of electrons. In practice, this means that the last percents of correlation energy can be recovered much more efficiently with KFAC, resulting in improved final energies for a given computational budget.

The effectiveness of natural gradient descent in this setting can be rationalized through its connection to the stochastic reconfiguration method[8,48], known from traditional variational quantum Monte Carlo optimization[49,50].

These higher order methods utilize the Fisher information of the unnormalized density associated with the wave function as a preconditioner to the gradients. KFAC extends the application range of natural gradient descent by low-rank approximating the Fisher information, facilitating its computation for neural network wave functions with large numbers of model parameters.

Instead of following the steepest descent in parameter space, an optimization step with the preconditioned gradient is in the direction of steepest descent in distribution space, with distance defined by the Kullback–Leibler (KL) divergence[51]. Considering that in VMC the predicted quantity $\psi_{\boldsymbol{\theta}}(\mathbf{r})$ directly defines the distribution $p(\mathbf{r}|\boldsymbol{\theta}) \propto |\psi_{\boldsymbol{\theta}}(\mathbf{r})|^2$, one concludes that a natural gradient step is in the direction of maximal loss decrease for a given KL divergence between $\psi_{\boldsymbol{\theta}}$ and $\psi_{\boldsymbol{\theta}+\mathrm{d}\boldsymbol{\theta}}$. This is an advantageous property, as relying on the KL divergence results in updates that are independent of the way $\psi_{\boldsymbol{\theta}}$ is parameterized, as opposed to the steepest descent where the Euclidean metric introduces strong dependence.

## V. SAMPLING

An important characteristic of VMC is that the data (electron positions) used to fit the model is generated in tandem with the optimization, by sampling the probability distribution of the electronic degrees of freedom defined by the square of the wave function. This sampling task comes with its own challenges, due to its tight coupling with the training. For the variational principle to

Accepted to J. Chem. Phys. 10.1063/5.0157512



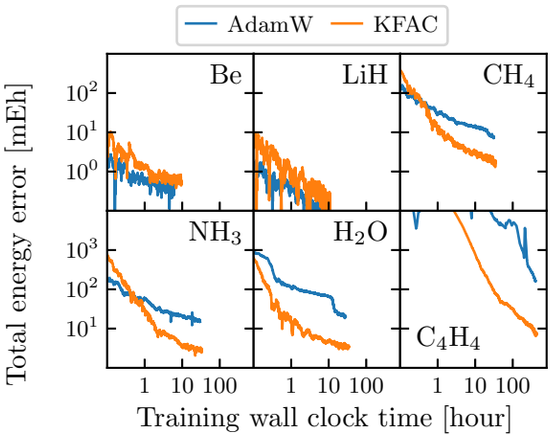


FIG. 2. **Comparing the performance of the AdamW and KFAC optimizers.** Total energy errors during the training process are shown for beryllium, lithium hydride, methane, ammonia, water, and cyclobutadiene. The horizontal axes show the wall clock time of the training, measured on a single Nvidia GTX 1080 Ti GPU. To obtain smooth training curves, the exponential moving average of the training energy is plotted. While on smaller systems (Be and LiH) AdamW converges slightly faster, due to its lower per-iteration cost, on the larger systems the benefit of using KFAC is clear.

remain valid, the samples used to evaluate (6) must be equilibrated according to the distribution $\mathbf{r} \sim |\psi_{\boldsymbol{\theta}}(\mathbf{r})|^2$. Furthermore, since $\psi_{\boldsymbol{\theta}}$ is updated in every training iteration, the sampling must account for the corresponding changes in the distribution of the electron positions. To carry out this demanding sampling task in a computationally efficient manner, the DEEPQMC program package offers two optimized Markov chain Monte Carlo (MCMC) algorithms. Along with the Random Walk MCMC algorithm[52,53], referred to as Metropolis sampler, the Metropolis-Adjusted Langevin Algorithm[54] (MALA), referred to as Langevin sampler, is also implemented, that proposes walker updates using overdamped Langevin dynamics. The implemented MALA includes the correction proposed by Hermann et al.[7], which scales the electron step sizes around the nuclei to avoid "overshooting" the latter. Additionally, changes of the wave function during training can be accounted for by re-equilibration after each gradient step or using a batch reweighting scheme. In the following sections, these MCMC samplers along with the above described sampling difficulties are characterized in more detail.

Accepted to J. Chem. Phys. 10.1063/5.0157512



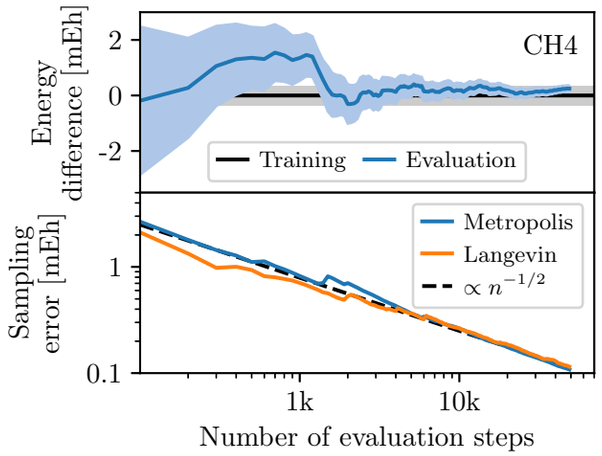


FIG. 3. **Top: Evaluation of a trained wave function ansatz, bottom: sampling error for the evaluation of a trained ansatz.** The convergence of the energy expectation value is depicted during the evaluation of an optimized CH4 ansatz using two thousand MCMC walkers. In the top pane, sampling is performed with the Random Walk MCMC algorithm. The evaluated energy is compared to the final training energy, with shaded areas showing error estimates. In the bottom pane, the convergence of the sampling errors of evaluations with the Metropolis sampler and the Langevin sampler are compared. The sampling errors converge as $n^{-1/2}$ with the number of samples $n$, as expected from the central limit theorem.

## A. Energy convergence

First, the convergence of the energy expectation value estimate is investigated, when sampling an unchanging, previously trained ansatz. In order to draw $n = n_\mathrm{b} \times n_\mathrm{s}$ electron samples $\{\mathbf{r}\}_{ij}$, distributed according to $|\psi_{\boldsymbol{\theta}}(\mathbf{r})|^2$, a batch of $n_\mathrm{b}$ many walkers is propagated for $n_\mathrm{s}$ MCMC steps. Based on the electron samples the energy expectation value is estimated as

$$\langle E \rangle = \frac{1}{n} \sum_{i=1}^{n_\mathrm{b}} \sum_{j=1}^{n_\mathrm{s}} E_\mathrm{loc}[\psi_{\boldsymbol{\theta}}](\{\mathbf{r}\}_{ij}) \,. \tag{33}$$

Following the central limit theorem[32,55], the sampling error of such estimates decays proportional to $n^{-1/2}$. To approximate the sampling error, we utilize the nonoverlapping batch means estimator, as reviewed by Flegal et al.[55]. We first obtain independent estimates of the energy by averaging the local energies over the walker trajectories (batches)

$$\langle E \rangle^i = \frac{1}{n_\mathrm{s}} \sum_{j=1}^{n_\mathrm{s}} E_\mathrm{loc}[\psi_{\boldsymbol{\theta}}](\{\mathbf{r}\}_{ij}) \,. \tag{34}$$

Considering these batches, the sampling error is then estimated as

$$\langle \sigma_E \rangle = \sqrt{\frac{\sum_{i=1}^{n_\mathrm{b}} (\langle E \rangle^i - \langle E \rangle)^2}{n_\mathrm{b}(n_\mathrm{b}-1)}} \,. \tag{35}$$

The convergence of the energy estimate and its error bar throughout the evaluation of an ansatz trained on the $CH_4$ molecule is plotted in Figure 3. In the top pane, the final value of the exponential walking mean of the training energies, and its estimated error are also shown with a horizontal line and shaded area. It can be seen from this plot that the energy estimate of the evaluation converges gradually towards the final training energy as expected, while its sampling error converges towards zero. Note that due to the parameter updates during the optimization, the energy estimate from the training is an unreliable estimate and a thorough evaluation of the energy expectation value requires sampling the ansatz with fixed parameters.

On the bottom pane of Figure 3, the convergence of the estimated sampling error is compared between the Metropolis sampler and the Langevin sampler. Importantly, the expected $n^{-1/2}$ convergence behavior is observed for both methods. Comparing the two algorithms, it can be seen that the error of MALA converges slightly faster than that of Random Walk MCMC, indicating a lower degree of correlation between the subsequent positions of the walkers of the Langevin sampler.

## B. Decorrelated sampling

To characterize the phenomenon of correlated samples hinted at in Section V A, autocorrelation functions of the local energy samples are investigated. The autocorrelation function of the local energies sampled by a single MCMC chain is defined as:

$$\rho_{E_\mathrm{loc}}(t) = \int_{-\infty}^{\infty} \left(E_\mathrm{loc}^{t'+t} - \mu_{E_\mathrm{loc}}\right)\left(E_\mathrm{loc}^{t'} - \mu_{E_\mathrm{loc}}\right) \mathrm{d}t' \,, \tag{36}$$

where $E_\mathrm{loc}^t$ denotes the local energy sampled at step $t$, and $\mu_{E_\mathrm{loc}}$ is the mean of the local energies over the entire trajectory. The autocorrelation time of the local energy is then computed as $\tau' = 2\int_0^\infty \rho_{E_\mathrm{loc}}(t)\mathrm{d}t$. Finally, $\tau$ is obtained by taking the mean of $\tau'$s over all propagated MCMC chains, providing a simple measure of local energy autocorrelation.

The autocorrelation times for five atoms of increasing size and the cyclobutadiene molecule are plotted on the upper pane of Figure 4, for both the Metropolis and the Langevin sampler. The

Accepted to J. Chem. Phys. 10.1063/5.0157512



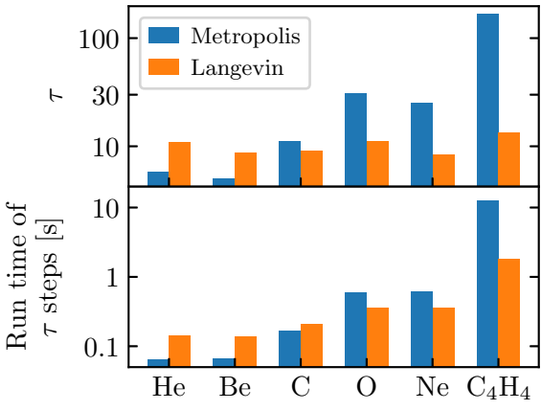


FIG. 4. **Autocorrelation times of the local energy samples.** The top pane shows the MCMC sampling autocorrelation time $\tau$, as defined in Section V B, for a sequence of atoms and the ground state of cyclobutadiene. The bottom pane shows the run time of performing $\tau$ sampling steps for the same systems. The electrons are sampled using either the Metropolis sampler or the Langevin sampler. Following the suggestion of Sokal[56], the autocorrelation times are estimated using MCMC chains of length $\approx 5\tau$.

general trend of longer autocorrelation times for larger systems can be observed for the Metropolis sampler. One of the main causes of this trend is the increasing nuclear charge, which induces higher and higher peaks in the distribution of the electrons near the nuclei. These pronounced peaks necessitate shorter update proposal radii, ultimately resulting in a higher correlation between subsequent samples. Furthermore, the autocorrelation time is expected to grow with the increasing complexity of the wave functions and their nodal surfaces. On the other hand, the Langevin sampler seems less affected by this trend, delivering largely constant autocorrelation times for all systems. It is reasonable to assume that by explicitly making use of information about the gradient of the wave function, the MALA update proposal retains better decorrelation efficiency than Random Walk MCMC, when considering more and more complicated wave functions. Finally, the showcased autocorrelation times are in reasonably good agreement with the fact that the default number of decorrelating steps performed between parameter updates is chosen between 10–30 in the currently used neural wave function program packages.

The experiments depicted in Figure 4 also demonstrate a slightly smaller correlation between subsequent samples of the Langevin sampler in comparison to those of the Metropolis sampler, for all but the smallest of systems. On the bottom pane of Figure 4, the wall clock run time of performing $\tau$ sampling steps are shown for each system, to account for the slightly increased com-

Accepted to J. Chem. Phys. 10.1063/5.0157512



putational cost of the MALA update proposal. Considering wall clock run times, the Metropolis sampler is more efficient on atoms up to carbon, while the Langevin sampler performs slightly to considerably better on the larger atoms and cyclobutadiene. While we find MALA to be more efficient than Random Walk Metropolis, we observe that for larger systems with heavier nuclei it could result in less stable optimization. To improve the black-boxed nature of the method, we applied Random Walk MCMC in all subsequent experiments.

## VI. SCALING

Understanding the scaling of a method's computational cost with the considered system size is of utmost importance in the field of quantum chemistry, where a pervasive caveat of the most accurate approaches is their unfavorable scaling behavior. Given its high accuracy, the asymptotic scaling of VMC (typically listed with $N^4$)[32] is considered favorable. Although this general scaling is indeed much better than e.g. the $N^7$ scaling of the gold-standard CCSD(T) method, and on par with the scaling of hybrid density functionals (such as DM21[3]), deep QMC calculations incur a larger prefactor, resulting in much higher practical costs on systems of intermediate size. While reducing this prefactor is an important long term goal of method developers in the field, investigating the method's scaling is also of interest, to estimate the prospect of system sizes feasible with further improvement and serve as a baseline for future developments. In this section, the scaling of the computational cost of the variational training of deep-learning ansatzes is investigated using the DEEPQMC program package. Further scaling aspects of the pseudopotential implementation, and design choices regarding the major computational bottlenecks of the algorithm are discussed in Appendix A.

The theoretical scaling of VMC ($N^4$) is obtained when combining the $N^3$ cost of the determinant evaluation with an additional factor of $N$ from the Laplacian required in the computation of the kinetic energy. In practice, however, for simulations with the currently feasible system sizes, the determinant evaluation makes up only a fraction of the computational cost, which is instead dominated by the evaluation of the neural networks of the ansatz. To investigate the practical scaling of a variational training step in DEEPQMC, single iteration run times are compared across atoms with increasing atomic numbers, as well as across chains containing an increasing number of hydrogen atoms (Figure 5). Although the systems contain different numbers of particles, due to the parametrization with GNNs the total parameter count of the wave function ansatz changes only

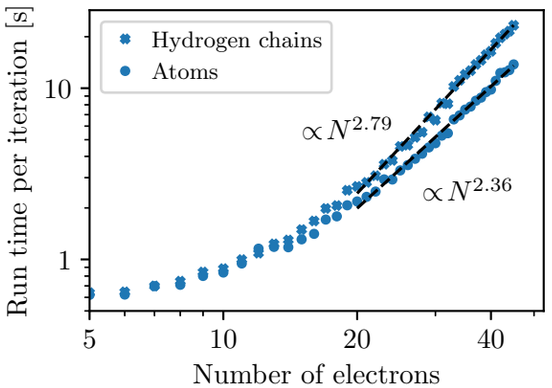


FIG. 5. **Scaling of the computational cost with system size.** The figure depicts the time in seconds per variational optimization step for systems with up to 45 electrons. The timings were obtained for training steps with a batch size of 2000 run, on a single A100 GPU. An exponential fit gives the scaling exponent of 2.79 for the hydrogen chains and 2.37 for the atoms respectively.

marginally between systems. On the other hand, owing to the varying numbers of nuclei, isoelectronic species can have slightly different computational requirements. The system classes of atoms and hydrogen chains were chosen, as they represent the lower and upper bounds respectively, on the number of nuclei a neutral system with a fixed number of electrons can contain. Consequently, the scaling of the run time with the number of electrons is also expected to be bounded by these system classes. With the tight empirical bounds of $N^{2.36-2.79}$ depicted in Figure 5, the observed scaling of DEEPQMC is still far below the theoretical estimate of $N^4$, highlighting the potential for extension to larger systems.

## VII. ANSATZ VALIDATION

Relying on the general framework introduced above, the DEEPQMC software suite enables the use of many of the previously published deep-learning ansatzes by providing configuration files to reproduce PauliNet[7], FermiNet[8] and DeepErwin[11]. To validate our implementation of these ansatzes, the hyperparameters of sampling, optimization, and GNN architecture are compared in depth to those of the respective reference implementations. Additionally, it is verified that when using the same parameters, the DEEPQMC implementations predict the same wave function value and local energy as their reference counterparts for a given configuration of electrons and nuclei.

Accepted to J. Chem. Phys. 10.1063/5.0157512



Note that we have refrained from exactly matching the cusp-corrected GTOs of PauliNet, because subsequent work has demonstrated that explicitly including a reference solution is limiting the accuracy of the ansatz. However, by combining Gaussian envelopes initialized from the coefficients of a reference calculation with a nuclear cusp correction in the Jastrow factor (26) it is possible to obtain a variant of PauliNet within DEEPQMC, that matches the characteristics of the original ansatz.

In Figure 6 the empirical performance of the various ansatzes is checked against results published in the literature for a small set of molecules. It can be seen that our DEEPQMC implementation of PauliNet, FermiNet, and DeepErwin matches the reference energies well. The remaining discrepancies of FermiNet result from slightly different experimental setups, such as an increased number of reference optimization steps (200 000 compared to 50 000 used here) and batch size (4096 compared to 2048 used here), or an older TensorFlow-based implementation being used in case of $N_2$. The impact of these changes on the deviations of the model accuracy highlights the importance and difficulty of comparing ansatzes implemented in different codebases under the same experimental conditions.

As a further contribution, we introduce and analyze the performance of a new default ansatz for the DEEPQMC program package, which we refer to as PauliNet2. This exemplary ansatz was optimized to have a good trade-off between accuracy and trainable model parameters. Despite achieving a similar accuracy as FermiNet and DeepErwin for the small systems under investigation (see Figures 6 and 7), the PauliNet2 ansatz has only about a third of the model parameters of FermiNet and a quarter of DeepErwin (i.e. for the CO molecule 239 829, 766 944, and 998 816 parameters respectively). The ansatz combines the SchNet-like graph convolutions of PauliNet (17) with the iterative update of the edge embeddings of FermiNet (18). Edge features are constructed from difference vectors between the electrons and isotropic exponentials are used as envelopes. Furthermore, the ansatz comprises a trainable Jastrow factor (24) as well as the fixed electronic cusp correction (25). While these hyperparameters are found to be suitable for the presented experiments, it is conceivable that an extended hyperparameter search targeting specific applications could further improve its performance. The detailed settings of the discussed ansatzes can be found in the respective configuration files shipped with the DEEPQMC package.

Accepted to J. Chem. Phys. 10.1063/5.0157512



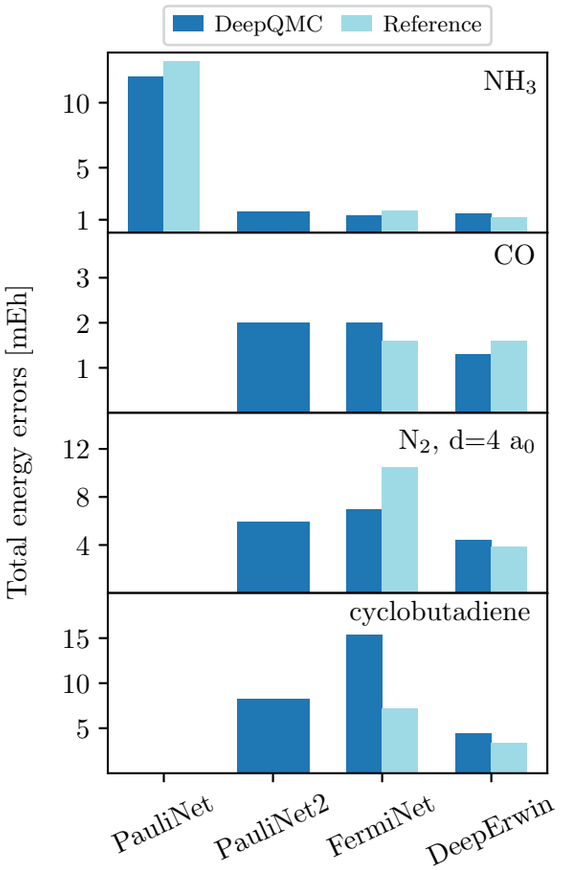


FIG. 6. **Validating the DEEPQMC implementation of various ansatzes by comparing their accuracy to published results obtained with their respective reference implementations.** Note that results obtained with the DEEPQMC or DeepErwin codebases were computed using 50 000 variational optimization steps and a batch size of 2048, while the FermiNet reference results used 200 000 training steps and 4096 samples in a batch. Results computed with the reference implementations are taken from the works of Pfau[8], Spencer[44], and Gerard[11].

## VIII. APPLICATION EXAMPLES

In this section, the ease of applying the DEEPQMC program package as a black-box method to obtain electronic energies is demonstrated on benchmark datasets. Two widely different example problems are chosen in order to showcase the general applicability of the presented method. The experiments are performed using DEEPQMC command line interface, which exposes all configuration options of the software suite while also allowing for effortless submission of simple calculations. Short usage examples of the DEEPQMC command line interface are provided in

Accepted to J. Chem. Phys. 10.1063/5.0157512



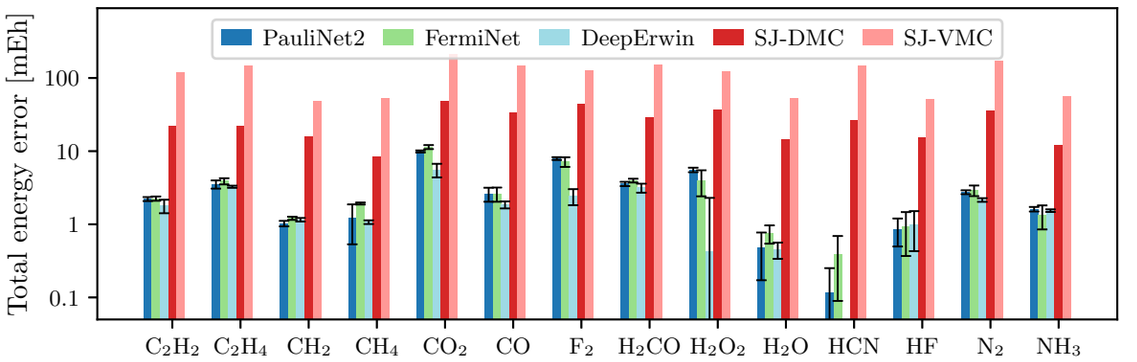


FIG. 7. **Total energy deviations for small molecules of H, C, O, N, and F atoms**. The presented molecules participate in the reactions investigated by Nemec[57]. Reference total and HF energies are taken from the review of O'Neill[60], while SJ VMC and SJ DMC results are taken from the work of Nemec[57]. Results for the hydrogen molecule are omitted, as it is described nearly exactly by all depicted methods. Error bars denote the sampling error as estimated according to (35).

Appendix B.

## A. Small molecule reactions

The electronic contributions to the reaction energies of 12 reactions involving small inorganic molecules and hydrocarbons are investigated. These reactions were used by Nemec[57] to benchmark the accuracy of Slater–Jastrow (SJ) type trial wave functions, constructed following Drummond et al.[58] using electron-electron, electron-nucleus, and electron-electron-nucleus terms in the Jastrow factor. The 14 participating molecules are built from H, C, O, N, and F atoms, containing at least 2 and at most 22 electrons. To facilitate the comparison with the DMC results of Nemec[57], the same molecular geometries are considered, obtained from the work of Feller[59]. Reference energies are taken from the review of O'Neill[60]. All electron, complete basis set extrapolated CCSD(T) energies are computed in house, using the PySCF program package[36].

First, single-point electronic energies obtained for the participating molecules are compared in Figure 7. On the vertical axis, the error of the recovered total energy is plotted, for VMC and DMC calculations utilizing SJ type trial wave functions, and for VMC with deep-learning ansatzes. Looking at Figure 7, one can observe that the total energy errors of SJ-VMC ansatzes are consistently above 47 mEh (with a mean of 114 mEh), while the associated DMC errors are

Accepted to J. Chem. Phys. 10.1063/5.0157512



in the range of 8 - 50 mEh (26 mEh on average). In comparison, deep-learning ansatzes exhibit at maximum only 11 mEh total energy error, with a mean deviation of 2.6 mEh. While the main goal of quantum chemistry methods is to accurately model energy differences, rather than recover exact total energies, it is encouraging to see that DEEPQMC and deep-learning ansatzes in general are very competitive in this area.

The accuracy of the energy differences obtained with SJ-DMC, CCSD(T), and deep-learning QMC methods are compared in Figure 8. Note that energy differences obtained with SJ-VMC are not shown in this figure, as they are an order of magnitude less accurate than the depicted approaches. Comparing SJ-DMC results with those obtained from DEEPQMC, one concludes that combining the VMC method with expressive deep-learning ansatzes greatly increases its accuracy, surpassing SJ-DMC on eleven out of twelve reactions. The accuracy advantage of DEEPQMC's PauliNet2, FermiNet, and DeepErwin ansatzes is similarly clear when comparing their respective reaction energy mean absolute deviations (MADs) of 2.4 mEh and 2.3 mEh, and 1.5 mEh to the 7.6 mEh of SJ-DMC. As a final comparison, Figure 8 also shows the reaction energy differences obtained from a complete basis set extrapolated, all-electron CCSD(T). Not surprisingly, CCSD(T) performs outstandingly on these small, single reference systems in equilibrium geometry, achieving a MAD of 3.4 mEh, and chemical accuracy (less than 1 kcal/mol or 1.6 mEh error) on four reactions. In comparison, the MAD value of PauliNet2 for this exemplary study with DEEPQMC is found to be below that of CCSD(T), and chemical accuracy is achieved on seven out of twelve reactions.

### B. Transition metal oxides

The effects of utilizing pseudopotentials in variational optimization of deep-learning molecular wave function are evaluated on a series of four first-row transition metal oxides. The bond lengths of the ScO, TiO, VO, and CrO molecules are taken from the experimental results of Annaberdieyev et al.[61]. The latest ccECP pseudopotentials[61] are applied to the transition metal atoms only, replacing neon-like cores of 10 electrons. Although replacing argon cores (18 electrons) with pseudopotentials would result in even larger computational savings, this is avoided as the third shell electrons are known to play a non-negligible role in the bond formation of transition metal atoms[62]. Apart from the introduction of pseudopotentials, the ansatz employed on small molecule reactions (Section VIII A) is utilized here without further modifications.

Accepted to J. Chem. Phys. 10.1063/5.0157512



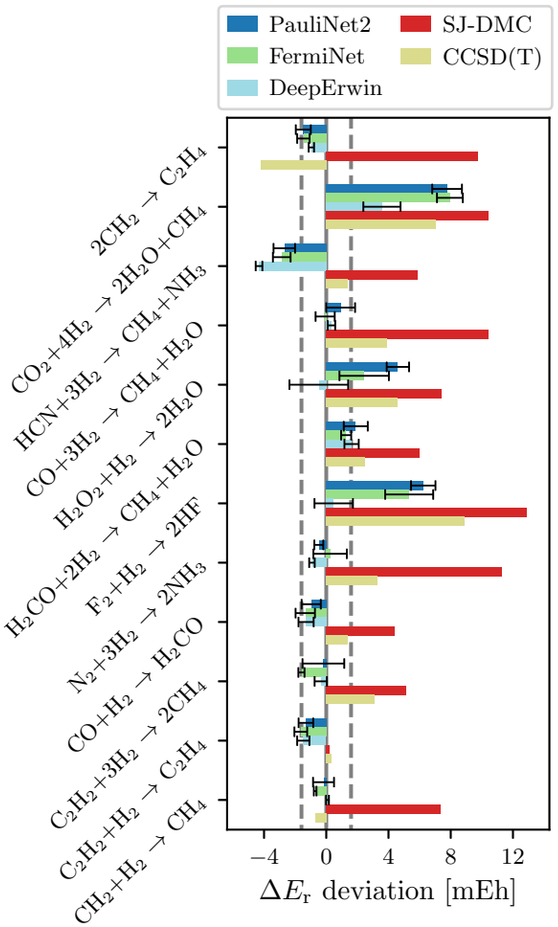


FIG. 8. **Reaction energy deviations for the reactions involving small molecules of H, C, O, N, and F atoms.** Reference reaction energies are computed from the electronic energies reviewed by O'Neill[60], SJ-DMC results are obtained from Nemec[57], while complete basis set extrapolated CCSD(T) values are calculated in house using PySCF[36]. The energies for PauliNet2, FermiNet and DeepErwin are obtained with the DEEPQMC program package. The range of $\pm 1$ kcal/mol deviation (often referred to as chemical accuracy) is highlighted with dashed lines.

Comparing the technical details of pseudopotential calculations to all-electron ones, the advantage of the former is clear. Due to the exclusion of the fastest moving core electrons, the length of the electron position updates is sixfold increased, and higher accuracy is achieved in a given number of training steps, at about half of the computational cost. Next, the dissociation energies of the four transition metal oxides are estimated. The dissociation energy for transition metal X is

Accepted to J. Chem. Phys. 10.1063/5.0157512



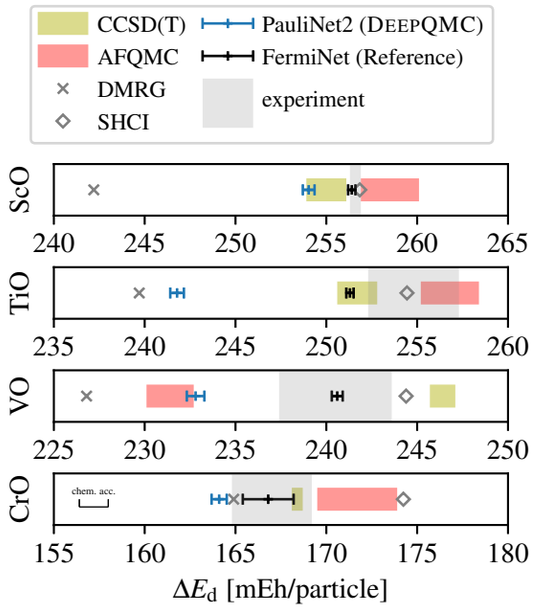


FIG. 9. **Dissociation energy of transition metal oxides calculated using different methods.** DEEPQMC+ECP result was obtained using 55 000 training steps and 5 000 evaluation steps. The results for FermiNet+ECP are taken from Li et al.[16], where they used 10 times more steps and consequently achieve higher accuracy. Other results are from the references[62,64].

defined as

$$\Delta E_\mathrm{d}^\mathrm{XO} = E^\mathrm{X} + E^\mathrm{O} - E^\mathrm{XO}\,, \tag{37}$$

where $E^\mathrm{O} = -75.0631(1)\,\mathrm{Ha}$ is the result of an all-electron calculation with the same hyperparameters. Figure 9 compares the obtained dissociation energies to experimental values[63], and some other accurate computational methods like CCSD(T)[61], FermiNet[16], auxiliary field quantum Monte Carlo (AFQMC), semi-stochastic heat batch configuration interaction (SHCI), and density matrix renormalization group (DMRG)[64]. Apart from the TiO case, the accuracy of DEEPQMC with pseudopotentials is comparable to other theoretical methods, such as CCSD(T) or AFQMC. The fact that the dissociation energy estimates with DEEPQMC are systematically lower than the experimental results, indicates that the single atoms are described more accurately than the oxide molecules. This can be counteracted by increasing the expressiveness of the ansatz and investing more compute. Note that results obtained with FermiNet[16] utilized a larger ansatz and trained for about ten times more training iterations than done in this study.

## IX. SUMMARY AND CONCLUSIONS

We have presented the DEEPQMC program package – a general variational quantum Monte Carlo framework for optimizing deep-learning molecular wave functions. The implementation focuses on modularity, facilitating rapid development of new ansatzes, and provides maximal freedom in choosing the specifics of the variational training setup. The ansatz shipped with DEEPQMC attempts to unify most of the currently existing deep-learning molecular wave functions, while remaining easy to extend as new models emerge. To reduce the computational complexity associated with heavy nuclei, some popular precomputed pseudopotentials are also implemented.

Using the framework provided by DEEPQMC, the most important practical aspects of variational optimization of deep-learning molecular wave functions are discussed. The importance of a proper gradient estimator along with robust gradient clipping is highlighted. For consistent ansatz initialization, supervised pretraining to HF wave functions is suggested. The advantages of using the second-order KFAC optimizer are demonstrated, along with a rationalization of its effectiveness. The theoretical convergence of the Markov Chain Monte Carlo sampling error is verified, and MALA is shown to be more effective in obtaining decorrelated samples than the widely utilized Random Walk MCMC algorithm. The empirical scaling of the method's computational cost is found to be more favorable than the most popular post-HF approaches, while its large prefactor is identified as an obstacle on the path to wider adoption.

The black-box application of the program package is demonstrated in two significantly different settings. The electronic reaction energies of 12 small molecule reactions are computed with a mean absolute deviation of 1.5 mEh, and compared to the 3.4 mEh achieved by CCSD(T) and 7.6 mEh achieved by DMC with SJ ansatzes. Using the same ansatz hyperparameters, dissociation energies are computed for a series of transition metal monoxides, utilizing the latest ccECP[61] pseudopotential. Improved training characteristics compared to all-electron calculations highlight the benefit of employing pseudopotentials. The accuracies of the predicted dissociation energies are on par with or exceed those of some other recently popularized methods such as auxiliary field quantum Monte Carlo, or density matrix renormalization group.

To conclude, the presented method shows great promise to become an easy-to-use, general, black-box method accurately describing the molecular electronic structure. Especially encouraging is the favorable scaling of computational requirements with increasing system size. It is easy to envision that after further development reducing the large prefactor of the computational costs,

the DEEPQMC package will prove valuable to the wider quantum chemistry community.

## DATA AVAILABILITY

The data that support the findings of this study will be made openly available upon publication of the manuscript.

## ACKNOWLEDGMENTS

We would like to thank Jannes Nys for helpful discussions regarding the nuclear cusp correction, and Leon Gerard for providing reference data for the ansatz comparison. The work of ZS and PBS is funded and supported by the Berlin Mathematics Research Center MATH+ (Project AA2-22). The work of MM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 772230). Computing time granted by the HLRN and provided on the supercomputer Lise compute at NHR@ZIB is gratefully acknowledged.

## REFERENCES

[1]F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, Annual review of physical chemistry **71**, 361 (2020).

[2]H. Kulik, T. Hammerschmidt, J. Schmidt, S. Botti, M. Marques, M. Boley, M. Scheffler, M. Todorović, P. Rinke, C. Oses, *et al.*, Electronic Structure **4**, 023004 (2022).

[3]J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez, D. Hassabis, and A. J. Cohen, Science **374**, 1385 (2021).

[4]J. Hermann, J. Spencer, K. Choo, A. Mezzacapo, W. M. C. Foulkes, D. Pfau, G. Carleo, and F. Noé, "Ab-initio quantum chemistry with neural-network wave functions," (2022), arXiv:2208.12590 [physics, stat].

[5]G. Carleo and M. Troyer, Science **355**, 602 (2017).

[6]J. Han, L. Zhang, and W. E, Journal of Computational Physics **399**, 108929 (2019).

[7]J. Hermann, Z. Schätzle, and F. Noé, Nature Chemistry **12**, 891 (2020), arXiv:1909.08423 [physics, stat].

[8]D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, Physical Review Research **2**, 033429 (2020), publisher: American Physical Society.

[9]Z. Schätzle, J. Hermann, and F. Noé, The Journal of Chemical Physics **154**, 124108 (2021).

[10]J. Lin, G. Goldshlager, and L. Lin, Journal of Computational Physics **474**, 111765 (2023).

[11]L. Gerard, M. Scherbela, P. Marquetand, and P. Grohs, "Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?" (2022), arXiv:2205.09438 [physics].

[12]I. von Glehn, J. S. Spencer, and D. Pfau, "A Self-Attention Ansatz for Ab-initio Quantum Chemistry," (2022), arXiv:2211.13672 [physics].

[13]M. T. Entwistle, Z. Schätzle, P. A. Erdman, J. Hermann, and F. Noé, Nature Communications **14**, 274 (2023), number: 1 Publisher: Nature Publishing Group.

[14]G. Pescia, J. Han, A. Lovato, J. Lu, and G. Carleo, Physical Review Research **4**, 023138 (2022), publisher: American Physical Society.

[15]M. Wilson, S. Moroni, M. Holzmann, N. Gao, F. Wudarski, T. Vegge, and A. Bhowmik, Physical Review B **107**, 235139 (2023).

[16]X. Li, C. Fan, W. Ren, and J. Chen, Physical Review Research **4**, 013021 (2022), arXiv:2108.11661 [physics.chem-ph].

[17]Y. Qian, W. Fu, W. Ren, and J. Chen, The Journal of Chemical Physics **157**, 164104 (2022).

[18]M. Wilson, N. Gao, F. Wudarski, E. Rieffel, and N. M. Tubman, arXiv preprint arXiv:2103.12570 (2021).

[19]W. Ren, W. Fu, X. Wu, and J. Chen, Nature Communications **14**, 1860 (2023).

[20]M. Scherbela, R. Reisenhofer, L. Gerard, P. Marquetand, and P. Grohs, "Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks," (2021), arXiv:2105.08351 [physics].

[21]N. Gao and S. Günnemann, "Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions," (2022), arXiv:2110.05064 [physics].

[22]N. Gao and S. Günnemann, "Sampling-free Inference for Ab-Initio Potential Energy Surface Networks," (2023), arXiv:2205.14962 [physics].

[23]M. Scherbela, L. Gerard, and P. Grohs, "Towards a Foundation Model for Neural Network Wavefunctions," (2023), arXiv:2303.09949 [physics].

[24]N. Gao and S. Günnemann, "Generalizing Neural Wave Functions," (2023), arXiv:2302.04168 [physics, physics:quant-ph].

[25]F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, SciPost Phys. Codebases , 7 (2022).

[26]J. Hermann, Z. Schätzle, P. B. Szabó, M. Mezera, and DeepQMC Contributers, "DEEPQMC," (2023).

[27]J. S. Spencer, D. Pfau, and FermiNet Contributors, "FERMINET," (2020).

[28]M. Scherbela, L. Gerard, and R. Reisenhofer, "DEEPERWIN," (2020).

[29]J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," (2018).

[30]G. Van Rossum and F. L. Drake, *Python 3 Reference Manual* (CreateSpace, Scotts Valley, CA, 2009).

[31]T. Hennigan, T. Cai, T. Norman, and I. Babuschkin, "Haiku: Sonnet for JAX," (2020).

[32]W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Reviews of Modern Physics **73**, 33 (2001).

[33]R. P. Feynman and M. Cohen, Physical Review **102**, 1189 (1956), publisher: American Physical Society.

[34]M. Burkatzki, C. Filippi, and M. Dolg, Journal of Chemical Physics **126**, 234105 (2007).

[35]M. C. Bennett, C. A. Melton, A. Annaberdiyev, G. Wang, L. Shulenburger, and L. Mitas, Journal of Chemical Physics **147**, 224106 (2017).

[36]Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E. Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov, and G. K.-L. Chan, The Journal of Chemical Physics **153**, 024109 (2020), publisher: American Institute of Physics.

[37]T. Kato, Communications on Pure and Applied Mathematics **10**, 151 (1957).

[38]K. Inui, Y. Kato, and Y. Motome, Physical Review Research **3**, 043126 (2021), publisher: American Physical Society.

[39]S. Sæther, T. Kjærgaard, H. Koch, and I.-M. Høyvik, Journal of Chemical Theory and Computation **13**, 5282 (2017), publisher: American Chemical Society.

[40]X. He and K. M. J. Merz, Journal of Chemical Theory and Computation **6**, 405 (2010), publisher: American Chemical Society.

[41]D. L. Yeager and P. Jo/rgensen, The Journal of Chemical Physics **71**, 755 (1979), publisher: American Institute of Physics.

[42]S.-i. Amari, in *Advances in Neural Information Processing Systems*, Vol. 9, edited by M. C. Mozer, M. Jordan, and T. Petsche (MIT Press, 1996).

[43]J. Martens and R. Grosse, in *Proceedings of the 32nd International Conference on Machine Learning* (PMLR, 2015) pp. 2408–2417, iSSN: 1938-7228.

[44]J. S. Spencer, D. Pfau, A. Botev, and W. M. C. Foulkes, "Better, Faster Fermionic Neural Networks," (2020), arXiv:2011.07125 [physics].

[45]G. Pescia, J. Nys, J. Kim, A. Lovato, and G. Carleo, "Message-Passing Neural Quantum States for the Homogeneous Electron Gas," (2023).

[46]A. Botev and J. Martens, "KFAC-JAX," (2022).

[47]I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," (2019), arXiv:1711.05101 [cs, math].

[48]Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Physical Review B **96**, 205152 (2017), publisher: American Physical Society.

[49]S. Sorella, Physical Review Letters **80**, 4558 (1998).

[50]S. Sorella, Physical Review B **64**, 024512 (2001).

[51]J. Martens, The Journal of Machine Learning Research **21**, 146:5776 (2020).

[52]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, The Journal of Chemical Physics **21**, 1087 (1953), publisher: American Institute of Physics.

[53]W. K. Hastings, Biometrika **57**, 97 (1970).

[54]J. Besag, Journal of the Royal Statistical Society: Series B (Methodological) **56**, 581 (1994).

[55]J. M. Flegal, M. Haran, and G. L. Jones, Statistical Science **23**, 250 (2008), publisher: Institute of Mathematical Statistics.

[56]A. Sokal (Springer, 1996).

[57]N. Nemec, M. D. Towler, and R. J. Needs, The Journal of Chemical Physics **132**, 034111 (2010), publisher: American Institute of Physics.

[58]N. D. Drummond, M. D. Towler, and R. J. Needs, Physical Review B **70**, 235119 (2004).

[59]D. Feller, K. A. Peterson, and D. A. Dixon, The Journal of Chemical Physics **129**, 204105 (2008), publisher: American Institute of Physics.

[60]D. P. O'Neill and P. M. W. Gill, Molecular Physics **103**, 763 (2005).

[61]A. Annaberdiyev, G. Wang, C. A. Melton, M. C. Bennett, L. Shulenburger, and L. Mitas, The Journal of Chemical Physics **149**, 134108 (2018), arXiv:1805.06436 [cond-mat.mtrl-sci].

[62]M. Dolg, U. Wedig, H. Stoll, and H. Preuss, Journal of Chemical Physics **86**, 866 (1987).

[63]Y. A. Aoto, A. P. de Lima Batista, A. Köhn, and A. G. S. de Oliveira-Filho, Journal of Chemical Theory and Computation **13**, 5291 (2017).

[64]K. T. Williams, Y. Yao, J. Li, L. Chen, H. Shi, M. Motta, C. Niu, U. Ray, S. Guo, R. J. Anderson, J. Li, L. N. Tran, C.-N. Yeh, B. Mussard, S. Sharma, F. Bruneval, M. van Schilfgaarde, G. H. Booth, G. K.-L. Chan, S. Zhang, E. Gull, D. Zgid, A. Millis, C. J. Umrigar, and L. K. Wagner (Simons Collaboration on the Many-Electron Problem), Phys. Rev. X **10**, 011041 (2020).

## Appendix A: Additional scaling experiments

### 1. Pseudopotentials

Figure 10 shows the scaling of the run time of the non-local pseudopotential evaluation (second term in (12)) on the third and fourth-row atoms. This term dominates the total computational overhead of using pseudopotentials overwhelmingly. From the 5 nested summations of (12), only the sum over the valence electrons scales with the number of electrons, hinting at an approximate linear scaling with system size. Looking at Figure 10, the obtained empirical scaling of $N^{1.19}$ is in good agreement with expectations. The sudden jump in run time from 20 to 21 electrons is caused by the reduction of valence electrons, as the utilized ccECP pseudopotentials use a larger core for 4p elements than for 3d ones.

### 2. Memory requirement

For all investigated applications, the memory requirement bottleneck is presented by the computation of the Laplacian of the wave function, $\sum_i^{3N} \frac{\partial^2}{\partial^2 r_i} \psi_{\boldsymbol{\theta}}(\mathbf{r})$. In this step, one obtains second derivatives of the wave function with respect to the electron coordinates. We obtain the derivatives of the wave function by applying automatic differentiation in backward-forward mode. While the gradient $\boldsymbol{\nabla}_{\mathbf{r}} \psi_{\boldsymbol{\theta}}$ is obtained in one backward pass for all coordinates, the diagonal of the Hessian $(\frac{\partial^2 \psi_{\boldsymbol{\theta}}}{\partial^2 r_i})$ requires an additional $3N$ forward mode differentiations to compute, one for each electron coordinate. Due to the flexible function transformations of JAX, both the serial and parallel exe-

Accepted to J. Chem. Phys. 10.1063/5.0157512



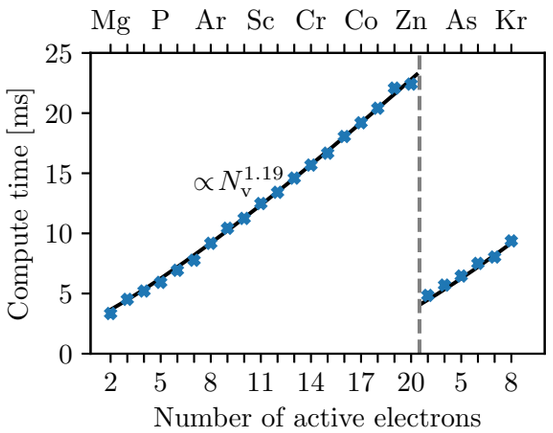


FIG. 10. **Scaling of the compute time of the non-local part of the ccECP pseudopotential.** The evaluation of the second term of Eq. (11) on a single NVIDIA GeForce RTX 3090 GPU. An exponential fit gives a scaling exponent of $1.19 \pm 0.04$ with the number of valence (explicitly treated) electrons. The pseudopotential uses a Neon core for elements up to Zn and a larger core for heavier atoms.

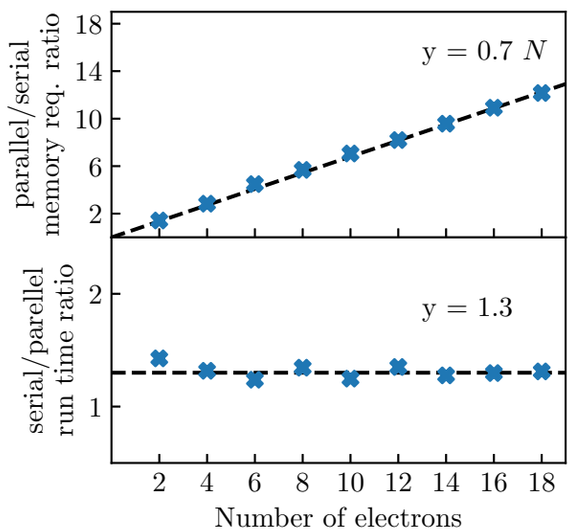


FIG. 11. **Comparing the memory requirements and run times of the serial and parallel Laplacian computations.** Data is obtained by evaluating the Laplacian of an untrained DEEPQMC ansatz with respect to all electron coordinates. The parallel implementation computes all diagonal elements of the Hessian at once, while the serial version computes one entry at a time. The number of diagonal Hessian entries scales linearly with the number of electrons. Run times measured on a single Nvidia GTX 1080 Ti GPU.

cution of the $3N$ forward mode differentiation passes can be implemented in a few lines of code, with the two implementations presumably differing in how they trade computational efficiency for memory requirement.

To decide between the serial and parallel approach to the Laplacian computation, benchmark calculations on a series of atoms with increasing nuclear charges are performed. The obtained relative memory requirements of the parallel and serial computations are presented on the left vertical axis of Figure 11. The observed linear scaling of the relative memory requirement between parallel and serial evaluations can be understood by considering that the parallel implementation holds data for all $3N$ backward passes in memory, while the serial approach stores data for a single pass at a time. However, the prefactor of the scaling curve is significantly less than three, which indicates that JAX performs some optimizations on the parallel code that reduce the naive $3N$ memory requirement multiplier. Considering run times of the two versions (lower panel of 11) it is found that the relative timings of the serial implementation over the parallel implementation do not scale with the system size. In fact, the ratio of run times between the serial and parallel implementations appears to converge around 1.5. Taking the above observations into account, the serial implementation of the Laplacian evaluation is chosen, due to its favorable scaling memory requirements which outweigh the slight, non-scaling run time edge of the parallel implementation.

## Appendix B: Usage of DEEPQMC

In this section, we provide a few minimal examples of the usage of the DEEPQMC command line interface. The interface is based on HYDRA, which provides a modular way to configure and execute complex jobs. DEEPQMC implements a wide variety of configuration options for the wave function ansatz as well as the hyperparameters of training and evaluation. For ease of use, the package includes predefined configuration files, which can be augmented using the command line or extended with custom configuration files. For a thorough tutorial and API documentation the reader is referred to the DEEPQMC documentation. For examples of typical DEEPQMC commands see Figure 12.

Accepted to J. Chem. Phys. 10.1063/5.0157512



```
# Optimize the default ansatz for H2O using decorrelated Langevin Sampling and
# a reduced KFAC norm constraint
deepqmc hamil/mol=H2O task/sampler=decorr_langevin task.opt.norm_constraint=0.0005

# Now use a FermiNet ansatz along with its default hyperparameters
deepqmc ansatz=ferminet task=train_ferminet hamil/mol=H2O hydra.run.dir=fn

# Evaluate the previously trained ansatz using 10k inference steps
deepqmc task=evaluate task.restdir=fn +task.steps=10000
```

FIG. 12. **Example usage of the DEEPQMC program package through its command line interface.**