



universität
wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

“Solving the electronic Schrödinger equation
using Deep Neural Networks and Transfer Learning”

verfasst von / submitted by

Leon Gerard, B.Sc. M.Sc.

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr. rer. nat.)

Wien, 2024 / Vienna, 2024

Studienkennzahl lt. Studienblatt /
degree programme code as it appears
on the student record sheet:

UA 796 605 405

Dissertationsgebiet lt. Studienblatt /
field of study as it appears on the student record sheet:

Mathematik

Betreut von / Supervisor:

Univ.-Prof. Dr. Philipp Grohs

Abstract

This thesis comprises a series of publications in the emerging field of deep learning-based Variational Monte Carlo (DL-VMC), which offers a highly accurate ab-initio method for computing electronic properties of molecules by solving the Schrödinger equation. The time-independent, non-relativistic Schrödinger equation is an eigenvalue equation, whereas the thesis places particular emphasis on the smallest eigenvalue and its corresponding eigenfunction. In DL-VMC, in an unsupervised manner, a neural network is optimized to represent the solution of the Schrödinger equation, also called the wave function. The method has recently demonstrated superior accuracy compared to other computational methodologies, albeit at a substantial computational cost due to the large number of trainable parameters in the neural network and the need for Monte Carlo integration to obtain a precise estimate of the eigenvalue.

The following thesis contributes to the recent improvements to enhance the method’s efficiency by introducing a novel technique for effectively learning wave functions for various molecules and their geometrical conformations in parallel. For the first time, we are able to optimize a wave function model on a diverse range of molecules and transfer this pretrained neural network to previously unencountered molecules, yielding competitive results with only a few fine-tuning steps. Furthermore, this work aims to improve the understanding of specific design choices involved in constructing neural network-based wave functions. In particular, we explore the effect of physical prior knowledge in the form of inexpensive, lower accuracy conventional methods, as initially proposed by Hermann, Schätzle, and Noé (2020), on the performance of the model. We demonstrate that this can lead to a deterioration of the final accuracy.

Zusammenfassung

Diese Arbeit fasst eine Reihe von Veröffentlichungen auf dem aufstrebenden Gebiet des Deep Learning basierten Variational Monte Carlo (DL-VMC) Ansatzes zusammen. DL-VMC ist eine hochgenaue ab-initio Methode zur Berechnung elektronischer Eigenschaften von Molekülen, welche man durch die Lösungen der Schrödinger Gleichung erhält. Die zeitunabhängige, nichtrelativistische Schrödinger-Gleichung ist eine Eigenwertgleichung, wobei der Schwerpunkt der folgenden Doktorarbeit auf dem kleinsten Eigenwert und der zugehörigen Eigenfunktion liegt. In DL-VMC wird auf unüberwachte Weise ein neuronales Netzwerk optimiert, um die Lösungen der Schrödinger-Gleichung, die Wellenfunktionen, zu repräsentieren. DL-VMC erreicht eine zum Teil höhere Genauigkeit im Vergleich zu anderen Berechnungsmethoden, wenn auch mit einem erheblichen Kostenaufwand aufgrund der großen Anzahl von trainierbaren Parametern im neuronalen Netzwerk und der Notwendigkeit der Monte-Carlo-Integration, um eine genaue Schätzung des Eigenwerts zu erhalten.

Die folgende Arbeit leistet einen Beitrag zu den jüngsten Fortschritten bei der Verbesserung der Effizienz der Methode. Dies wird erreicht durch die Einführung einer neuartigen Technik zum effektiven Lernen von Wellenfunktionen über verschiedene Moleküle und ihren geometrischen Konformationen. Wir führen eine verallgemeinerte Wellenfunktion ein, die auf einer Vielzahl von Molekülen und deren geometrischen Konformationen vortrainiert werden kann. Dies ermöglicht die effiziente Übertragung des neuronalen Netzes auf unbekannte Moleküle, was zu genauen Ergebnissen mit nur wenigen Optimierungsschritten führt. Darüber hinaus zielt diese Arbeit darauf ab, das Verständnis für spezifische Design-Entscheidungen bei der Konstruktion von Wellenfunktionen auf der Basis neuronaler Netzwerke zu verbessern. Ein besonderes Augenmerk liegt auf der Einbeziehung eines physikalischen Priors in Form von kostengünstigeren und konventionellen Methoden mit geringerer Genauigkeit, wie ursprünglich von Hermann, Schätzle, und Noé (2020) vorgeschlagen wurde. Wir zeigen, dass dieser Ansatz zu einer signifikanten Verzerrung während der Optimierung führen kann, was eine Abnahme der Genauigkeit zur Folge hat.

Acknowledgment

First of all, I want to express my gratitude to Philipp Grohs for his invaluable support throughout my PhD, especially for motivating me to pursue my own research interest. His guidance and the research environment he provided have been instrumental in my academic journey.

I also want to acknowledge the collaborations with Michael Scherbela, Rafael Reisenhofer, Philipp Marquetand, and Halvard Sutterud, which have led to the research presented in this thesis. The insightful discussions and shared knowledge during this time were invaluable. Special thanks goes to Michael Scherbela. I have learned immensely from his expertise in the field of physics.

Furthermore, I extend my heartfelt thanks to Sofia and my parents. Their support has been the cornerstone of my time at the University of Vienna.

Contents

Preamble

1	Introduction	13
2	Preliminaries	16
3	Synopses of the Publications.	27
4	Discussion	41
	References	43

Publications

I	Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?	51
II	Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks	71
III	Towards a transferable fermionic neural wavefunction for molecules	91
IV	Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions	109

Appendix

	Author contributions	133
--	--------------------------------	-----

Preamble

1 Introduction

Accurate predictions of electronic properties hold paramount significance in the field of material design and drug discovery. Solving the Schrödinger equation offers a pathway to accessing these properties through the solutions of the partial differential equation, the so-called wave functions. The time-independent, non-relativistic Schrödinger equation, considered in the following thesis, takes the form of an eigenvalue equation, with the smallest eigenvalue corresponding to the ground-state energy. The *potential energy surface* characterizing the energy landscape of a molecule with respect to its geometrical conformations is of particular significance. An efficient evaluation of the energy surface allows, for example, to perform a computational structure search or to compute chemical reaction rates. For these types of tasks, highly accurate estimates of the ground-state energy become essential. For example, a study by Barone et al. (2013) demonstrated that the geometrical conformations of the smallest amino acid, Glycine, are partitioned by transition barriers of only ~ 20 millihartree (mHa). To put the scale of the transition barrier into perspective, the ground-state energy is typically several Hartree (refer to Fig. 1.1 for a visual representation). Unfortunately, an analytical

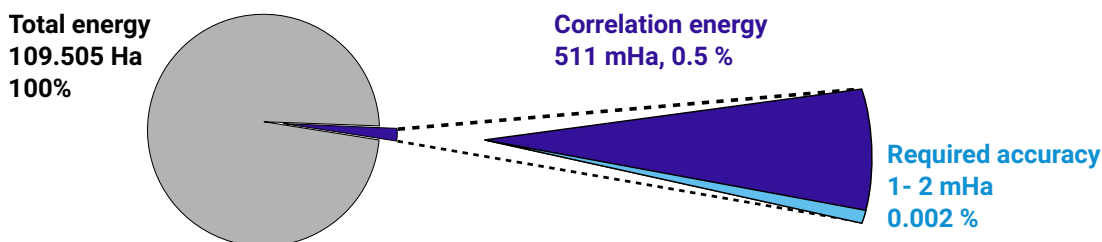


Figure 1.1: Conceptual visualization of the ground-state energy of a Nitrogen dimer. Figure is taken from Gerard et al. (2022).

solution to the Schrödinger equation exists only for the single-particle case. This limitation has led to the development of a diverse range of computational methods, including renowned techniques like Density Functional Theory (DFT), which was awarded the Nobel Prize in 1998. On the one hand, there exist methods like DFT or Hartree Fock that can handle systems comprising hundreds of atoms with relatively crude but efficient approximations. On the other hand, methods such as CCSD(T), often named the gold standard when it comes to precise approximations, tend to exhibit scaling behavior typically in the order of $O(n_{el}^7)$, where n_{el} represents

the number of electrons considered. In the case of the configuration interaction singles, doubles, triples, quadruples (CISDTQ) method the scaling can even go up to $O(n_{el}^{10})$ (Scherbela et al., 2022), limiting the method to rather small system sizes. Another way to categorize the computational landscape is by classifying methods into variational and non-variational. Variational methods, such as Hartree Fock or the more accurate Full Configuration interaction approach, are lower bounded by the ground-state energy, ensuring a direct assessment of the accuracy. On the other hand, methods like DFT or CCSD(T) are non-variational and can potentially underestimate the ground-state energy.

In 2020, Pfau et al. (2020) and Hermann, Schätzle, and Noé (2020) integrated Deep Learning into Variational Monte Carlo (DL-VMC) within first quantization to efficiently tackle the Schrödinger equation for gas-phase molecules. They introduced a variational approach that offers accuracy comparable to CCSD(T) while exhibiting a theoretical scaling behavior of $O(n_{el}^4)$. Rather than depending on basis functions, they introduced a neural network architecture as an ansatz for the wave function, merging the Slater-Jastrow type ansatz (Foulkes et al., 2001) with Deep Neural Networks. However, this promising approach faces limitations due to the computational overhead required for optimizing a neural network. In particular the computational cost arises from the substantial number of trainable parameters and the necessity of utilizing Monte Carlo integration as an estimator for the ground-state energy. In addition, a new network must be optimized when considering a different geometrical conformation or molecule.

Much of this thesis aims at strategies and techniques to reduce the computational cost to enhance the method’s applicability. Among other ideas, we therefore leverage techniques such as *deep transfer learning*, wherein one trains a neural network on a corpus of tasks, often unsupervised. The pretrained network can then be applied to entirely new problems, requiring only a brief and computationally efficient fine-tuning phase. For instance, this practice is prevalent in the field of Natural Language Processing (NLP) (Devlin et al., 2018) as well as in Computer Vision (Alayrac et al., 2022) where a large model is trained on a corpus of text or images and transferred to entirely new problems with the ability to generalize. In a series of papers, we discuss the idea of a generalized wave function architecture. This architecture enables us to pretrain a single neural network on a set of molecules with varying number of particles, including a diverse range of geometrical conformations. The pretrained model can then be used for an efficient transfer to new geometrical conformations and even entirely new molecules with a substantial reduction in computational cost required for accurate predictions of the ground-state energy, especially for the computation of a complete potential energy surface. Additionally, we closely examine the originally proposed architectures in Pfau et al.

(2020) and Hermann, Schätzle, and Noé (2020), while also addressing challenges when incorporating physical prior knowledge into the neural network architecture.

2 Preliminaries

To introduce the topic of DL-VMC, we briefly discuss the general problem statement, including constraints such as the Pauli exclusion principle. Afterwards, we describe a general framework of neural network architectures in first quantization and related design choices to the architecture, largely based on the original work of Pfau et al. (2020) and Hermann, Schätzle, and Noé (2020).

2.1 General problem statement

The time-independent, non-relativistic Schrödinger equation amounts to an eigenvalue equation

$$H\psi(\mathbf{x}) = E\psi(\mathbf{x}), \quad (2.1)$$

whereas the molecular system under the *Born-Oppenheimer approximation* is described by the following Hamiltonian

$$\begin{aligned} H = & -\frac{1}{2} \sum_{i=1}^{n_{\text{el}}} \nabla_{\mathbf{r}_i}^2 + \sum_{i=1}^{n_{\text{el}}-1} \sum_{j=i+1}^{n_{\text{el}}} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \\ & + \sum_{I=1}^{n_{\text{nuc}}-1} \sum_{J=I}^{n_{\text{nuc}}} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{i=1}^{n_{\text{el}}} \sum_{I=1}^{n_{\text{nuc}}} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|. \end{aligned} \quad (2.2)$$

with $\mathbf{r}_i \in \mathbb{R}^3$, $i \in \{1, \dots, n_{\text{el}}\}$ denoting the electron coordinates and $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_{n_{\text{nuc}}}) \in \mathbb{R}^{n_{\text{nuc}} \times 3}$, $\mathbf{Z} = (Z_1, \dots, Z_{n_{\text{nuc}}}) \in \mathbb{N}^{n_{\text{nuc}}}$ refer to the nuclear coordinates and their respective charges. The geometrical conformation of a molecule is summarized by $\{(\mathbf{R}, \mathbf{Z})\} = \{(\mathbf{R}_1, Z_1), \dots, (\mathbf{R}_{n_{\text{nuc}}}, Z_{n_{\text{nuc}}})\}$. The Hamiltonian encompasses the kinetic energy through the Laplace operator and the Coulomb attraction forces between electrons and nuclei, represented by the terms 2, 3, and 4 in eq. (2.2). The eigenfunctions in the eigenvalue equation are denoted as wave functions, each associated with its corresponding eigenvalue, also referred to as the energy $E \in \mathbb{R}$. In the following thesis, we are purely interested in the ground-state energy, denoted as the smallest eigenvalue E_0 from equation (2.1), and its corresponding eigenfunction ψ_0 . For the sake of simplicity, the dependence of the ground-state energy and the eigenfunction on the nuclear coordinates are omitted. However, in general, there is a strong interest in determining the potential energy surface, in other words, the ground-state energy with respect to the nuclear coordinates $\mathbf{R}$.

Increasing complexity for solving the eigenvalue equation arises from the Pauli exclusion principle, which the wave function must adhere to for electronic systems such as gas-phase molecules (Szabo and Ostlund, 1996). Consequently, the wave function’s dependence extends beyond just the electron coordinates to the respective spin states of each electron, i.e. $\mathbf{x} = ((\mathbf{r}_1, \sigma_1), \dots, (\mathbf{r}_{n_{\text{el}}}, \sigma_{n_{\text{el}}}))$, $\sigma_i \in \{\uparrow, \downarrow\}$, $i \in \{1, \dots, n_{\text{el}}\}$. Although the Hamiltonian is spin-independent, the wave function must exhibit antisymmetry with respect to the permutation of equal spin electrons. In other words, suppose electron i and j possess the same spin, then the wave function must satisfy

$$P_{i,j}\psi(\dots, (\mathbf{r}_i, \sigma_i), \dots, (\mathbf{r}_j, \sigma_j), \dots) = -\psi(\dots, (\mathbf{r}_j, \sigma_j), \dots, (\mathbf{r}_i, \sigma_i), \dots).$$

with the operator $P_{i,j}$ permuting the input, i.e.

$$P_{i,j}\psi(\dots, (\mathbf{r}_i, \sigma_i), \dots, (\mathbf{r}_j, \sigma_j), \dots) = \psi(\dots, (\mathbf{r}_j, \sigma_j), \dots, (\mathbf{r}_i, \sigma_i), \dots).$$

Without loss of generality, we drop in the following the explicit spin dependence of the electrons as input to the wave function. Instead, we ensure the condition by reordering the electron coordinates $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_{\uparrow}}, \dots, \mathbf{r}_{n_{\uparrow}+n_{\downarrow}})$ such that the first $n_{\uparrow}$ electrons have spin-up ($\uparrow$) and the remaining $n_{\downarrow} = n_{\text{el}} - n_{\uparrow}$ have the opposite spin (spin-down $\downarrow$).

2.2 Variational principle

Having access to the precise wave function, specifically for the smallest eigenvalue, enables the computation of various electronic properties and facilitates a comprehensive description of the physical system under study. Unfortunately, as indicated in the introduction, analytical solutions are only available for single-particle systems, necessitating the dependence on numerical approximations for higher-dimensional problems. One category of these methods relies on a variational principle called the Rayleigh-Ritz principle (Szabo and Ostlund, 1996).

Assume ψ_{θ} to be an arbitrary antisymmetric function (square-integrable with sufficient regularity) parametrized by some parameters θ . In the subsequent sections, these will represent the trainable neural network weights. However, in general, those parameters can also define a different function class. Considering the Rayleigh-Ritz quotient

$$E_{\theta} = \int_{\mathbb{R}^{3 \times n_{\text{el}}}} \frac{\psi_{\theta}(\mathbf{r}) H \psi_{\theta}(\mathbf{r})}{\Omega_{\theta}} d\mathbf{r} = \mathbb{E}_{\mathbf{r} \sim p_{\theta}(\mathbf{r})} \left[\frac{H \psi_{\theta}(\mathbf{r})}{\psi_{\theta}(\mathbf{r})} \right] \quad (2.3)$$

with normalization factor $\Omega_{\theta} = \int_{\mathbb{R}^{3 \times n_{\text{el}}}} \psi_{\theta}(\mathbf{r})^2 d\mathbf{r}$ and density $p_{\theta}(\mathbf{r}) = \frac{|\psi_{\theta}(\mathbf{r})|^2}{\Omega_{\theta}}$, the task of discovering the smallest eigenvalue and its corresponding eigenfunction can

be redefined as the minimization of eq. (2.3). In other words, due to the variational principle, it holds

$$\mathbb{E}_{\mathbf{r} \sim p_\theta(\mathbf{r})} \left[\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})} \right] \geq E_0. \quad (2.4)$$

Equality is reached in the case of ψ_θ representing the ground-state eigenfunction ψ_0 . This lower bound gives us an objective for optimizing the parameters of ψ_θ and estimating the ground-state energy by E_θ . A key focus of this thesis revolves around developing a highly expressive neural network-based ansatz to construct such an antisymmetric function ψ_θ . Subsequently, ways to efficiently optimize the trainable parameters θ using the variational principle (2.4) are discussed. However, we first introduce an approach to efficiently compute the high-dimensional integral.

2.3 Metropolis-Hastings Algorithm

During optimization of the parameters θ and to get a final prediction of the ground-state energy, it is essential to have an efficient method for calculating the high-dimensional integral in eq. (2.3). In Variational Monte Carlo, one commonly employs *Markov chain Monte Carlo* (MCMC), utilizing algorithms like *Metropolis-Hastings*. This allows to sample from p_θ and perform Monte Carlo integration to approximate the Rayleigh-Ritz quotient, providing an estimator for the ground-state energy

$$E_\theta = \mathbb{E}_{\mathbf{r} \sim p_\theta(\mathbf{r})} \left[\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})} \right] \approx \frac{1}{n_s} \sum_{a=1}^{n_s} \frac{H\psi_\theta(\mathbf{r}^a)}{\psi_\theta(\mathbf{r}^a)}$$

with $\mathbf{r}^a \sim p_\theta(\mathbf{r})$ and n_s Monte Carlo samples. Fortunately, we can completely disregard the normalization constant Ω_θ , simplifying significantly the task of estimating E_θ . This is for two reasons, first of all, the Metropolis-Hastings algorithm allows us to sample from an unnormalized density. Furthermore, due to the equivalent reformulation of the Rayleigh-Ritz quotient (2.3), the normalization constant factors out in the so-called local energy $\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})}$.

The density p_θ we sample from with MCMC gets updated during optimization due to the dependence on θ (c.f. Sec. 2.5). To ensure the sampled electron positions follow the correct distribution, we have to perform multiple consecutive steps of MCMC, whereas a single step with the Metropolis-Hastings algorithm can be broken down into the following stages, as outlined in algorithm 1. Firstly, starting with an initial set of electron positions, either from a previous step or randomly initialized, represented as $\mathbf{r} \in \mathbb{R}^{n_{\text{el}} \times 3}$ a proposed state $\mathbf{r}^{\text{prop}} \in \mathbb{R}^{n_{\text{el}} \times 3}$ is generated. It is obtained by sampling from a normal distribution with a step size

$\gamma > 0$. Subsequently, the acceptance probability is computed, and the proposed state is accepted if this probability exceeds a uniformly distributed random value. Throughout the sampling process, the step size is adjusted dynamically to maintain an acceptance rate of approximately 50% over a batch of electron positions.

Algorithm 1 A single Markov chain Monte Carlo step

Input: Electron positions $\mathbf{r} \in \mathbb{R}^{n_{\text{el}} \times 3}$, stepsize $\gamma \in \mathbb{R}_{\geq 0}$, wave function ψ_θ
Output: Electron positions $\mathbf{r}^{\text{new}} \in \mathbb{R}^{n_{\text{el}} \times 3}$

▷ Propose new step
 $\epsilon_i \sim \mathcal{N}(0, \gamma^2)$ ▷ $\epsilon_i \in \mathbb{R}^3, \forall i \in \{1, \dots, n_{\text{el}}\}$
 $\mathbf{r}_i^{\text{prop}} \leftarrow \mathbf{r}_i + \epsilon_i$ ▷ $\forall i \in \{1, \dots, n_{\text{el}}\}$

▷ Calculate acceptance probability
 $\alpha_{\mathbf{r}^{\text{prop}}, \mathbf{r}} \leftarrow \min \left\{ 1, \frac{|\psi_\theta(\mathbf{r}^{\text{prop}})|^2}{|\psi_\theta(\mathbf{r})|^2} \right\}$
 $u \sim \mathcal{U}([0, 1])$
if $\alpha_{\mathbf{r}^{\text{prop}}, \mathbf{r}} > u$ **then**
 $\mathbf{r}^{\text{new}} \leftarrow \mathbf{r}^{\text{prop}}$
else
 $\mathbf{r}^{\text{new}} \leftarrow \mathbf{r}$
end if

Currently, the Metropolis-Hastings algorithm remains the prevailing sampling approach for DL-VMC methods. A typical challenge encountered while sampling electron positions arises from the varying length scales near the nuclei compared to those farther away, especially in the case of heavier atoms. As a consequence, to maintain a given acceptance rate of 50%, the stepsize γ gets reduced. The different length scales arise because of increasing charges the probability distribution is stronger peaked around the nuclei (Schätzle et al., 2023). Hence, Schätzle et al. (2023) compared the Metropolis-Hastings algorithm to a Langevin Monte Carlo approach (Umrigar, Nightingale, and Runge, 1993) with slightly less decorrelated samples. On the other hand, von Glehn, Spencer, and Pfau (2023) proposed a slight modification to algorithm 1 by dividing the number of electrons into multiple blocks of electrons and separately accepting / rejecting a subset of electron positions with a similar effect as Langevin Monte Carlo.

2.4 Neural network-based wave function

By combining Slater determinants (Slater, 1929) with many-body correlation orbitals Pfau et al. (2020) as well as Hermann, Schätzle, and Noé (2020) established expressive ansätze for the ground-state wave function, called FermiNet and

PauliNet respectively. In principle, these neural network architectures, representing the parametrized wave function ψ_θ , can be divided into three parts.

First, the computation of a high-dimensional electron representation capturing many-body correlations using a neural network κ

$$\mathbf{h}_i = \kappa(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}, \{(\mathbf{R}, \mathbf{Z})\}) \in \mathbb{R}^{d_{\text{emb}}}$$

whereas the set of electrons is split into the set of spin-up electrons $\{\mathbf{r}_\uparrow\} = \{\mathbf{r}_1, \dots, \mathbf{r}_{n_\uparrow}\}$ and the set of spin-down electrons $\{\mathbf{r}_\downarrow\} = \{\mathbf{r}_{n_\uparrow+1}, \dots, \mathbf{r}_{n_{\text{el}}}\}$ with $i \in \{1, \dots, n_{\text{el}}\}$.

In the next step, every electron embedding $\mathbf{h}_i$ is evaluated for n_{el} many orbitals. For example, one could use to generate orbitals a linear mapping $\mathbf{F} = [\mathbf{F}_1, \dots, \mathbf{F}_{n_{\text{el}}}] \in \mathbb{R}^{n_{\text{el}} \times d_{\text{emb}}}$ and exponentially decaying envelope functions with respect to the nuclei positions $\varphi_k : \mathbb{R}^3 \times \mathbb{R}^{n_{\text{nuc}} \times 3} \rightarrow \mathbb{R}$

$$\Phi_{ik} = \varphi_k(\mathbf{r}_i, \mathbf{R}) \phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}, \{(\mathbf{R}, \mathbf{Z})\}) \quad (2.5)$$

$$\phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}, \{(\mathbf{R}, \mathbf{Z})\}) = \langle \mathbf{F}_k, \kappa(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}, \{(\mathbf{R}, \mathbf{Z})\}) \rangle \in \mathbb{R} \quad (2.6)$$

with $\langle \cdot, \cdot \rangle$ denoting the dot product and $k \in \{1, \dots, n_{\text{el}}\}$. The specific choice of φ_k will be discussed in more detail in the following paragraphs but essentially assures the boundary condition of the wave function far from the nuclei (Pfau et al., 2020).

Finally, to ensure the antisymmetry of the ansatz, one evaluates the (Slater) determinant of the orbital matrix

$$\psi_\theta(\mathbf{r}, \{(\mathbf{R}, \mathbf{Z})\}) = \det \left([\Phi_{ik}]_{i,k=1}^{n_{\text{el}}} \right) \in \mathbb{R}. \quad (2.7)$$

We want to emphasize that $\theta \in \mathbb{R}^{n_{\text{params}}}$ includes all trainable parameters of the neural network. In particular, it also contains parameters from the embedding and the orbital construction such as $\mathbf{F}$. In order to enhance the clarity of our notation, we opt to eliminate the inclusion of θ as a variable during the intermediate construction stages of the neural network-based wave function. To further simplify the notation, we also drop the dependence on the geometry of the molecule $\{(\mathbf{R}, \mathbf{Z})\}$. In the subsequent paragraphs, we will provide a comprehensive analysis of each of the three parts of the architecture.

Embedding In DL-VMC, the primary source of expressive power in the neural network-based wave function can be found within the embedding structure. Architectures such as FermiNet or PauliNet allocate most of their trainable weights to the function κ . Hence, substantial efforts have been dedicated to enhancing this part of the architecture using attention mechanisms (Pescia et al., 2023; von

Glehn, Spencer, and Pfau, 2023) and graph convolution (Gao and Günnemann, 2023a; Gerard et al., 2022; Hermann, Schätzle, and Noé, 2020).

In the following, we want to give an exemplary choice of the embedding architecture, recently proposed in Scherbela, Gerard, and Grohs (2023). The idea is to operate on the graph of particles connected via their edges containing information about their relative positions. The edges are computed using multilayer perceptrons (MLP)

$$\begin{aligned}\mathbf{e}_{ij}^{\text{el-el}} &= \text{MLP}^{\sigma_{ij}}([\mathbf{r}_{ij}, r_{ij}]) \\ \mathbf{e}_{iI}^{\text{el-nuc}} &= \text{MLP}^{\text{e-n}}([\mathbf{r}_{iI}, r_{iI}]) \\ \mathbf{e}_{IJ}^{\text{nuc-nuc}} &= \text{MLP}^{\text{n-n}}([\mathbf{R}_{IJ}, R_{IJ}])\end{aligned}\tag{2.8}$$

embedding the electron-electron $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = |\mathbf{r}_{ij}|$, the electron-nuclear $\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$, $r_{iI} = |\mathbf{r}_{iI}|$ and the nuclear-nuclear $\mathbf{R}_{IJ} = \mathbf{R}_I - \mathbf{R}_J$, $R_{IJ} = |\mathbf{R}_{IJ}|$, differences and distances into a potentially higher dimensional feature space with $[\cdot]$ resembling the concatenation operation and $i, j \in \{1, \dots, n_{\text{el}}\}$, $I, J \in \{1, \dots, n_{\text{nuc}}\}$. Driven by the antisymmetry exhibited by electrons with the same spin, we distinguish between two types of electron-electron edges in our computation. These are edges between electrons of the same spin, meaning both electrons i and j are either spin-up or spin-down ($\sigma_{ij} = \text{"same"}$), and edges between electrons of different spins, where electrons i and j are in opposite spin states ($\sigma_{ij} = \text{"diff"}$) and therefore, different MLPs and trainable parameters are applied.

Message passing updates are employed to transfer information among particles. A single message passing step is broken down into the following operations

$$\begin{aligned}\tilde{\mathbf{a}}_i^{\text{rec}} &= \text{MessagePassing}(\mathbf{a}_i^{\text{rec}}, \{\mathbf{a}_j^{\text{send}}\}, \{\mathbf{e}_{ij}\}) \\ &= \gamma \left(\text{Linear}(\mathbf{a}_i^{\text{rec}}) + \text{Linear}(\mathbf{m}_i) \right)\end{aligned}$$

with messages

$$\mathbf{m}_i = \sum_{j=1}^{n_p} \text{Linear}(\mathbf{a}_j^{\text{send}}) \odot \text{Linear}(\mathbf{e}_{ij}).$$

Here, $\tilde{\mathbf{a}}_i^{\text{rec}}$ represents the updated state of a receiving particle, influenced by the information from a set of sending particles $\{\mathbf{a}_j^{\text{send}}\} = \{\mathbf{a}_1^{\text{send}}, \dots, \mathbf{a}_{n_p}^{\text{send}}\}$, which are connected through their edges $\{\mathbf{e}_{ij}\} = \{\mathbf{e}_{i1}, \dots, \mathbf{e}_{in_p}\}$, where n_p is the number of sending particles. A common technique to generate the messages $\mathbf{m}_i$ for every node i is to use graph convolution with an element-wise multiplication $\odot$ along the feature dimension. The linear layers (Linear) map the edge and particle representations

to the same feature dimension. The non-linear activation function $\gamma(\cdot)$ is applied element-wise. No weights are shared across the different linear layers, increasing the potential expressivity of the network.

With the construction of the edges and message passing step we can now build the complete embedding architecture. First, we initialize nuclear embeddings

$$\hat{\mathbf{h}}_I^{\text{nuc}} = \text{OneHot}(Z_I) \quad \tilde{\mathbf{h}}_I^{\text{nuc}} = \text{MessagePassing}(\hat{\mathbf{h}}_I^{\text{nuc}}, \{\hat{\mathbf{h}}_J^{\text{nuc}}\}, \{\mathbf{e}_{IJ}^{\text{nuc-nuc}}\}) \quad (2.9)$$

using one-hot encoding of the nuclear charges with $I, J \in \{1, \dots, n_{\text{nuc}}\}$. This allows us to initialize the electron embeddings $\mathbf{h}_i^0$

$$\mathbf{h}_i^0 = \text{MessagePassing}(\mathbf{0}, \{\tilde{\mathbf{h}}_J^{\text{nuc}}\}, \{\mathbf{e}_{iJ}^{\text{el-nuc}}\}),$$

with a zero vector $\mathbf{0}$ as an initial receiving electron embedding for every electron i . Finally, to incorporate electron-electron interactions, we perform multiple rounds of message passing steps between electrons

$$\mathbf{h}_i^{l+1} = \text{MessagePassing}(\mathbf{h}_i^l, \{\mathbf{h}_i^l\}, \{\mathbf{e}_{ij}^{\text{el-el}}\})$$

for $l \in \{0, \dots, L-1\}$. After L iterations we obtain the final embedding vector $\mathbf{h}_i = \mathbf{h}_i^L \in \mathbb{R}^{d_{\text{emb}}}$ for every electron.

Orbitals The electron embeddings are subsequently employed to construct the many-body orbital matrix $[\Phi_{ik}]_{i,k=1}^{n_{\text{el}}}$. The matrix can be further broken down into the product of orbitals $\phi_k(\mathbf{r}_i, \{\mathbf{r}_{\uparrow}\}, \{\mathbf{r}_{\downarrow}\})$, dependent on all particles, and exponentially decaying envelope functions $\varphi_k(\mathbf{r}_i, \mathbf{R})$ (c.f. eq. (2.6)). The incorporation of the functions φ_k is motivated by the need to guarantee the proper asymptotic behavior of the wave function ansatz ψ_{θ} for electrons located far from the nucleus (Pfau et al., 2020). A generalized form of the initial FermiNet and PauliNet implementation, recently implemented in the DeepQMC framework (Schätzle et al., 2023), can be written as

$$\varphi_k(\mathbf{r}_i, \mathbf{R}) = \sum_{I=1}^{n_{\text{nuc}}} \sum_{b=1}^{n_{\text{basis}}} c_{kIb} \exp(-\alpha_{kIb} r_{iI}^{\beta}) \quad (2.10)$$

with (potentially) trainable parameters $c_{kIb} \in \mathbb{R}$ and $\alpha_{kIb} \in \mathbb{R}$. One major difference between FermiNet and PauliNet is their specific choice of envelope functions. FermiNet uses $\beta = 1$, i.e. Slater-type orbitals, with $n_{\text{basis}} = 1$. On the other hand, PauliNet relies on Gaussian-type orbitals with $\beta = 2$ and initializes the coefficients c_{kIb} and α_{kIb} using a reference calculation such as Hartree Fock in a minimal basis set with $n_{\text{basis}} > 1$ basis functions (Hermann, Schätzle, and Noé,

2020). This is a slight simplification of the original PauliNet architecture. For example, the original work used cusp-corrected Gaussian-type orbitals. However, it was demonstrated that the envelope functions from eq. (2.10) exhibit similar behavior to the originally employed ones (Schätzle et al., 2023).

An important aspect to bear in mind when constructing the orbitals is the anti-symmetry property of the wave function. As we will explore in the next paragraph, antisymmetrization is usually achieved by computing the determinant of the orbital matrix. This requirement imposes that the orbital functions must be equivariant with respect to the exchange of same spin electrons. In other words let i and j be two electrons of same spin then it should hold

$$P_{i,j}\phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}) = \phi_k(\mathbf{r}_j, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\})$$

for all orbitals $k \in \{1, \dots, n_{\text{el}}\}$ with $P_{i,j}$ being the operator exchanging the positions of two electrons. The property ensures that when two electrons of the same spin exchange positions, two rows of the matrix $[\Phi_{ik}]_{i,k=1}^{n_{\text{el}}}$ are also exchanged. Consequently, the determinant of the matrix changes its sign. This restriction also impacts the design of the embedding function κ . It can be inferred from the message passing steps, that the proposed architecture in this section enforces the desired symmetries (see also Schätzle et al. (2023) for a more general type of graph neural network). Furthermore, by employing distinct network weights for edges of same and different spin electron pairs (c.f. eq. (2.8)), we ensure that the neural network-based wave function is not necessarily antisymmetric regarding the exchange of electrons of different spin, a symmetry that is not required for determining the ground-state wave function.

Finally, we want to discuss the dependency on the system size when constructing the orbital matrix. A significant portion of this thesis explores the transferability of pretrained neural network-based wave functions across diverse physical systems. Therefore, we aim to develop a generalized wave function ansatz that can be optimized across various geometries of the same as well as across different molecules. In eq. (2.6) the linear mapping $\mathbf{F} \in \mathbb{R}^{n_{\text{el}} \times d_{\text{emb}}}$ together with the trainable coefficients c_{kIb} and α_{kIb} in eq. (2.10), depend on the number of electrons and nuclei considered. All these parameters are optimized during the training of the neural network. When dealing with independent calculations for different systems, this presents no restriction. However, challenges arise when one wants to train a single neural network across molecules with varying number of particles. In such cases, it is not clear how to efficiently share those parameters. In Sec. 3.2, 3.3 and 3.4, this problem and a possible solution will be discussed further.

Antisymmetrization As previously mentioned, one method to ensure antisymmetry is by employing Slater determinants (Slater, 1929), a standard technique in

classical Variational Monte Carlo and widely utilized in recent deep learning-based methods (Gerard et al., 2022; Hermann, Schätzle, and Noé, 2020; Pfau et al., 2020). In contrast to the description in eq. (2.7) one commonly applies not a single determinant but generates orbitals for multiple determinants, normally in the range of 10-40. This helps to increase the accuracy as discussed in Pfau et al. (2020). From a scaling perspective, the determinant presents a significant challenge, with a complexity that scales as $O(n_{\text{el}}^3)$ in asymptotic terms. This has led to the development of alternative approaches with a more favorable scaling. For instance, Han, Zhang, and E (2019) used the product of scalar-valued functions ϕ represented by a deep neural network:

$$\prod_{i < j} (\phi(\mathbf{r}_i, \mathbf{r}_j, r_{ij}) - \phi(\mathbf{r}_j, \mathbf{r}_i, r_{ij})).$$

A more recent approach achieved, based on a sorting algorithm, an asymptotic scaling of $O(n_{\text{el}} \log(n_{\text{el}}))$ (Richter-Powell et al., 2023). Despite ongoing research aimed at improving the computational scaling of the antisymmetrization, current results in terms of accuracy are not yet comparable to those achieved with Slater determinants which remains the method of choice.

2.5 Optimization

After discussing the variational principle, the sampling technique for the high-dimensional integral and introducing key parts of the neural network architecture, we can now delve into the problem of optimizing the neural network parameters. Applying the variational principle, we can utilize the formulation provided in eq. (2.4) as our loss function:

$$L(\theta) = \mathbb{E}_{\mathbf{r} \sim p_{\theta}(\mathbf{r})} \left[\frac{H\psi_{\theta}(\mathbf{r})}{\psi_{\theta}(\mathbf{r})} \right].$$

To perform gradient descent, the computation of the gradient with respect to the parameters would in general require third derivatives due to the kinetic energy in the Hamiltonian. Furthermore, stochastic gradient descent potentially leads to a biased estimator because the sampling process depends on the parametrized wave function ψ_{θ} . Fortunately, one can exploit the hermiticity of the Hamiltonian to rewrite the gradient with respect to the parameters of the loss function as:

$$\nabla_{\theta} L(\theta) = 2\mathbb{E}_{\mathbf{r} \sim p_{\theta}(\mathbf{r})} \left[\left(\frac{H\psi_{\theta}(\mathbf{r})}{\psi_{\theta}(\mathbf{r})} - L(\theta) \right) \nabla_{\theta} \log |\psi_{\theta}| \right]. \quad (2.11)$$

Eq. (2.11) allows the computation of an unbiased estimator for stochastic gradient descent with, at most, second derivatives. Additionally, in the case that ψ_{θ} represents the true wave function, the Monte Carlo estimator has zero variance due to

the local energy $\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})}$ being spatially constant. A full derivation of the gradient can be found for example in Inui, Kato, and Motome (2021).

A commonly used optimizer in deep learning is Adam (Kingma and Ba, 2017), which performs parameter update steps in the steepest descent direction in Euclidean space, as discussed by Martens and Grosse (2015) and Zhang et al. (2023). Although it is the standard optimizer in many deep learning applications, it was demonstrated that second-order methods relying on the Fisher information matrix yield better performance in terms of accuracy and computational cost when applied to the wave function optimization for the Schrödinger equation (Pfau et al., 2020; Schätzle et al., 2023). This is driven by the fact that a small change in the parameter space can result in completely different wave functions and physical quantities (Becca and Sorella, 2017). As a result, each gradient step is preconditioned using the inverse of the Fisher matrix $\mathbf{M} \in \mathbb{R}^{n_{\text{params}} \times n_{\text{params}}}$

$$M_{ij} = \mathbb{E}_{\mathbf{r} \sim p_\theta(\mathbf{r})} \left[\frac{\partial \log p_\theta(\mathbf{r})}{\partial \theta_i} \frac{\partial \log p_\theta(\mathbf{r})}{\partial \theta_j} \right], \quad i, j = 1, \dots, n_{\text{params}}$$

which in the case of unnormalized distributions is equivalent to the stochastic reconfiguration method (Pfau et al., 2020), an optimization technique used in many Quantum Chemistry applications (Casula, Attaccalite, and Sorella, 2004; Pescia et al., 2023). Resulting in the following natural gradient update step

$$\theta^{\text{new}} = \theta - \eta \mathbf{M}^{-1} \nabla_\theta L(\theta)$$

with learning rate η . This corresponds to an update step in the steepest descent direction in terms of the probability distribution defined by the wave function ψ_θ , measured by the Kullback-Leibler divergence (Amari, 1998; Martens and Grosse, 2015; Zhang et al., 2023). Since inverting the Fisher matrix $\mathbf{M}$ becomes infeasible for large neural networks, it is necessary to approximate it through methods such as Conjugated Gradient (Gao and Günnemann, 2023b) or Kronecker-factored approximate curvature (KFAC) (Martens and Grosse, 2015). Although additional computational overhead is incurred in each optimization step due to the computation and inversion of the approximated Fisher matrix, the reduced number of steps required to attain sufficient accuracy makes the use of second-order methods advantageous (c.f. Fig. 2.1).

In addition to the variational optimization, Pfau et al. (2020) proposed to perform a supervised optimization phase with respect to a reference method. This step, referred to as *supervised pretraining* in the following, minimizes the difference between the neural network orbitals and reference orbitals

$$L^{\text{pre}}(\theta) = \mathbb{E}_{\mathbf{r} \sim p_\theta(\mathbf{r})} \left[\sum_{k=1}^{n_{\text{el}}} \sum_{i=1}^{n_{\text{el}}} (\phi_k^{\text{ref}}(\mathbf{r}_i, \mathbf{R}) - \varphi_k(\mathbf{r}_i, \mathbf{R}) \phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}))^2 \right],$$

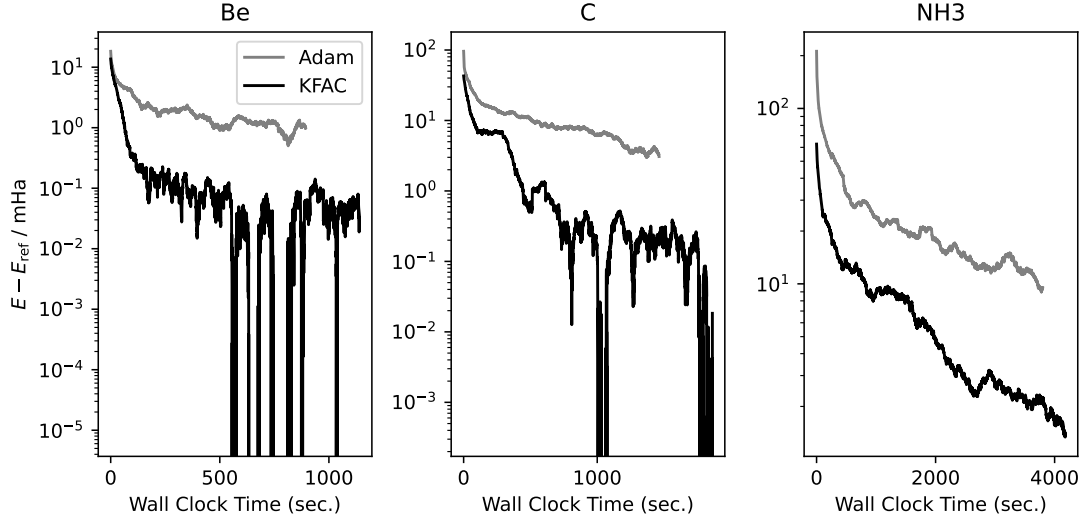


Figure 2.1: Comparing the optimizers Adam (grey) and KFAC (black) for three molecules (Beryllium, Carbon, and Ammonia). The x-axis represents the wall clock time in seconds, and the y-axis (log-scale) the energy error with respect to a highly accurate reference calculation in millihartree (mHa).

whereas the neural network-based orbitals are calculated as described in eq. (2.5). As reference calculation, a mean field solution such as Hartree Fock, is usually used, where each orbital depends on a single electron. Pfau et al. (2020) argued that supervised pretraining can improve convergence and numerical stability of the subsequent variational optimization. In Ch. I, potential negative side effects are discussed when one overfits to the provided reference solution.

3 Synopses of the Publications

This chapter presents the main concepts and findings of the publications that are contained in this thesis, see Table 3.1 for an overview.

	Title	Reference
Ch. I Sec. 3.1	<i>Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?</i>	Gerard, Scherbela, Marquetand, and Grohs (2022), <i>NeurIPS</i> .
Ch. II Sec. 3.2	<i>Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks</i>	Scherbela, Reisenhofer, Gerard, Marquetand, and Grohs (2022), <i>Nature Computational Science</i> .
Ch. III Sec. 3.3	<i>Towards a transferable fermionic neural wavefunction for molecules</i>	Scherbela, Gerard, and Grohs (2024), <i>Nature Communications</i> .
Ch. IV Sec. 3.4	<i>Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions</i>	Scherbela, Gerard, and Grohs (2023), <i>NeurIPS</i> .

Table 3.1: Overview of the publications included in this thesis.

3.1 Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?

The paper is structured into two main parts. In the first part, we enhance the embedding architecture originally proposed by FermiNet, incorporating concepts from PauliNet. In the second part, we empirically analyze the bias introduced through physical prior knowledge, specifically in the envelope functions, as referenced in eq. (2.10).

A principal distinction between the two architectures, PauliNet and FermiNet, can be made through the construction of messages between particles. Let us denote

again the i -th electron embedding after l iterations by $\mathbf{h}_i^l$, which contains information about the molecule’s geometry $\{(\mathbf{R}, \mathbf{Z})\}$, and the other electrons. Both FermiNet and PauliNet update the electron embedding through a residual connection as follows:

$$\mathbf{h}_i^{l+1} = \mathbf{f}_i^l + \mathbf{h}_i^l.$$

Here, the messages are denoted by $\mathbf{f}_i^l$, whereas PauliNet incorporates graph convolution

$$\begin{aligned}\mathbf{m}_i^{\text{el-el}} &= \sum_j \text{Linear}(\mathbf{h}_j^l) \odot \text{Linear}(\mathbf{e}_{ij}^{\text{el-el}}) \\ \mathbf{m}_i^{\text{el-nuc}} &= \sum_I \text{Linear}(\mathbf{h}_I^{\text{nuc}}) \odot \text{Linear}(\mathbf{e}_{iI}^{\text{el-nuc}})\end{aligned}\tag{3.1}$$

and uses as message a sum of electron and nuclei features

$$\mathbf{f}_i^l = \text{MLP}(\mathbf{m}_i^{\text{el-el}}) + \text{MLP}(\mathbf{m}_i^{\text{el-nuc}})$$

with $\mathbf{h}_I^{\text{nuc}}$ being a one-hot encoding of the charge Z_I similar to (2.9). On the other hand, FermiNet emphasizes the direct use of electron embeddings as inputs

$$\mathbf{f}_i^l = \text{MLP}\left(\left[\mathbf{h}_i^l, \frac{1}{n_\uparrow} \sum_{j=1}^{n_\uparrow} \mathbf{h}_j^l, \frac{1}{n_\downarrow} \sum_{j=1}^{n_\downarrow} \mathbf{h}_j^l, \frac{1}{n_\uparrow} \sum_{j=1}^{n_\uparrow} \mathbf{e}_{ij}^{\text{el-el}}, \frac{1}{n_\downarrow} \sum_{j=1}^{n_\downarrow} \mathbf{e}_{ij}^{\text{el-el}}\right]\right)\tag{3.2}$$

with $\mathbf{e}_{ij}^{\text{el-el}}$, $\mathbf{e}_{iI}^{\text{el-nuc}}$ defining edges between different particles similar to (2.8). We want to emphasize that this is a simplified summary of PauliNet and FermiNet. For better readability, details like spin dependence and the precise construction of the edges are omitted. There are minor differences in the construction of those edges and the initialization of the electron embedding $\mathbf{h}_i^0$. Particularly, FermiNet initializes $\mathbf{h}_i^0$ as the concatenation of the electron-nuclei differences and distances for each electron to all nuclei

$$\mathbf{h}_i^0 = [\mathbf{r}_{iJ}, r_{iJ}]_{J=1}^{n_{\text{nuc}}} \in \mathbb{R}^{4n_{\text{nuc}}}.$$

This introduces the geometrical information of the molecule only at the beginning of the embedding in contrast to PauliNet at every iteration l through the use of electron-nuclei edges. While FermiNet outperforms PauliNet in terms of accuracy, this can largely be attributed to the significantly higher number of trainable parameters employed in its neural network architecture (700k vs. 150k) (Hermann, Schätzle, and Noé, 2020; Pfau et al., 2020) and as we see in the second part of the paper, due to a different choice of envelope function. Comparing the graph-

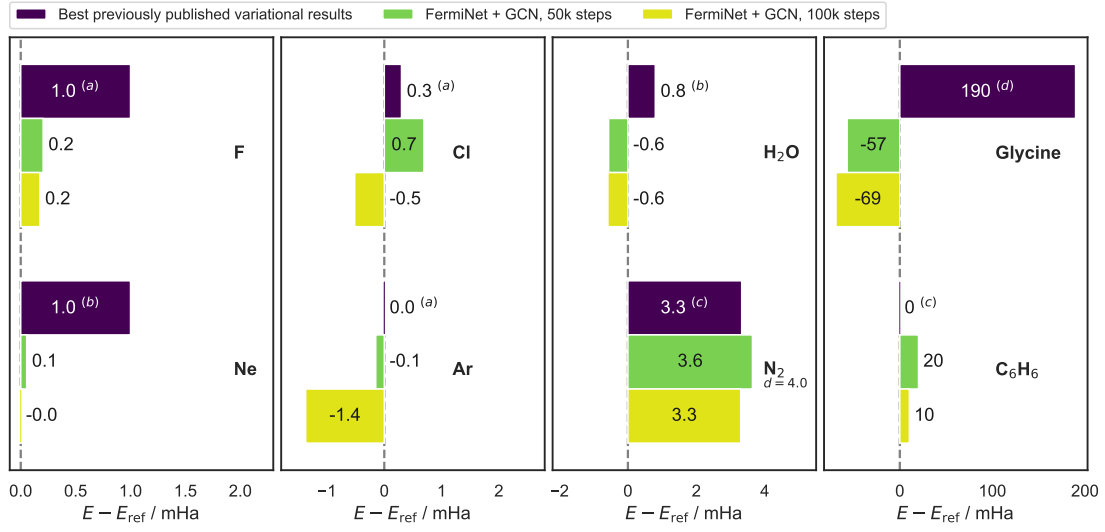


Figure 3.1: Energies compared to the previous best estimate for the ground-state energy in mHa (dotted grey line). Blue represents the best published variational energy. Green and yellow show our approach (FermiNet + GCN) with different amounts of optimization steps. The footnotes represent a: FermiNet VMC (Pfau et al., 2020), b: Conventional DMC (Clark et al., 2011; Nemec, Towler, and Needs, 2010; Seth, Ríos, and Needs, 2011), c: FermiNet DMC (Ren et al., 2023), d: MRCI-F12 (Gerard et al., 2022). The full figure can be found in Gerard et al. (2022).

convolution operations in (3.1) with the summation over edges in (3.2), it becomes apparent that the graph convolution is strictly more expressive by simply considering the linear layers in (3.1) as identity and constant one vector. This motivates the fusion of the two architectures to generate messages of the following form:

$$\mathbf{f}_i^l = \text{MLP} \left(\left[\mathbf{h}_i^l, \frac{1}{n_\uparrow} \sum_{j=1}^{n_\uparrow} \mathbf{h}_j^l, \frac{1}{n_\downarrow} \sum_{j=1}^{n_\downarrow} \mathbf{h}_j^l, \mathbf{m}_i^{\text{el-el}}, \mathbf{m}_i^{\text{el-nuc}} \right] \right).$$

The use of more expressive inter-particle features in the form of graph convolution between electrons and nuclei results in improved accuracy on a variety of molecules (c.f. Fig. 3.1).

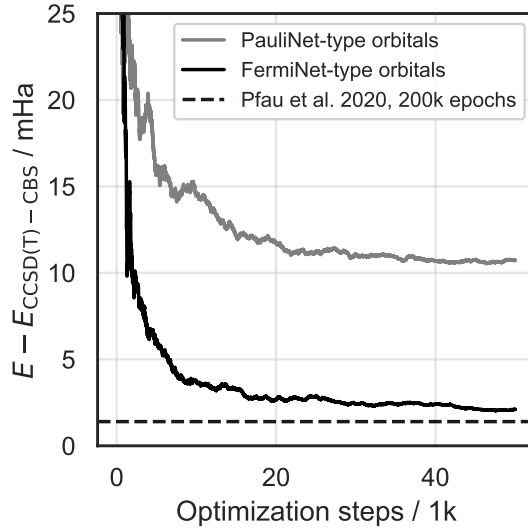


Figure 3.2: Comparison of PauliNet-type orbitals (3.3) with FermiNet-type orbitals (3.4) keeping the rest of the ansatz fixed. Both models calculated the ground-state energy of ammonia (NH_3). The full figure can be found in Gerard et al. (2022).

In the second part of the paper, we investigate the effect of introducing prior knowledge into the network architecture in the form of the envelope functions φ_k , $k \in \{1, \dots, n_{\text{el}}\}$ using computational methods such as CASSCF or Hartree Fock. As highlighted in Sec. 2.4, PauliNet relies on Gaussian-type orbitals for the d -th determinant of the form

$$\varphi_{kd}(\mathbf{r}_i, \mathbf{R}) = \sum_{I=1}^{n_{\text{nuc}}} \sum_{b=1}^{n_{\text{basis}}} c_{kIbd} \exp(-\alpha_{kIbd} r_{iI}^2) \quad (3.3)$$

initializing the parameters c_{kId} and α_{kId} using coefficients from a CASSCF calculation and freezing them during optimization. FermiNet uses a more flexible approach with Slater-type orbitals

$$\varphi_{kd}(\mathbf{r}_i, \mathbf{R}) = \sum_{I=1}^{n_{\text{nuc}}} c_{kId} \exp(-\alpha_{kId} r_{iI}) \quad (3.4)$$

and trainable parameters c_{kId} , α_{kId} . We compare both envelopes in Fig. 3.2, keeping the rest of the neural network architecture fixed while using a PauliNet-like embedding architecture. The results indicate that CASSCF orbitals, as prior knowledge, introduce a bias in the wave function model that proves challenging to overcome during the optimization phase. The observation is additionally supported by experiments to match a reference calculation using supervised pretraining (c.f. Chapter I).

3.2 Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks

The paper presented here marks the inception of a series aimed to develop a generalized neural network ansatz capable of representing the ground-state wave function for a wide range of molecules and geometries using a single neural network. This approach holds the promise of faster optimization, achieving an overall reduction in computational cost. As demonstrated in the previous section, the combination of Variational Monte Carlo with Deep Learning can achieve accuracy comparable to state-of-the-art conventional approaches with a theoretical scaling of $O(n_{\text{el}}^4)$ as elaborated in Pfau et al. (2020). Albeit these promising results, the originally proposed neural network ansatz faces challenges in the form of a large constant computational prefactor compared to conventional methods such as CCSD(T) or DFT. This is primarily attributable to the large number of trainable parameters and the need for Monte Carlo integration of the energy estimator. As a result, such limitation hinders the method from being more widely utilized. Particularly in cases where calculations for many different geometries are necessary, for example, in computational structure search or when serving as training data for supervised machine learning methods, e.g., machine-learned interatomic potentials (Batzner et al., 2022). This motivation led to an effort to enhance the computational efficiency of neural network-based wave functions by exploiting potential regularities within a molecule’s chemical and geometrical space. As an initial step, the paper focuses on the geometrical space of a single molecule.

We present a technique named *weight sharing*. In this approach, we generate

multiple instances of a neural network-based wave function, each corresponding to a different geometrical configuration of a molecule. During the optimization process, a significant portion of the trainable parameters are shared among these network instances. Specifically, the shared components of the network yield identical function outputs across different geometries. This method is inspired by the recent successes in other machine learning domains, particularly transfer learning. Transfer learning involves utilizing parts of a pretrained model to address tasks different from the ones it was initially trained on (Alayrac et al., 2022; Devlin et al., 2018).

The neural network architecture in the paper relies on an improved version of the PauliNet ansatz using a CASSCF calculation as an initialization of the envelope functions. As discussed in Sec. 2.4, the construction of the orbitals consists of an embedding network κ , a backflow matrix $\mathbf{F}$ and the envelope functions φ (c.f. eq. (2.5) and (2.6)). We hypothesized that the trainable parameters of the embedding function κ are suited to share across geometries since the embedding serves as a feature extractor which is not directly dependent on the specific geometry (Scherbela, Reisenhofer, Gerard, Marquetand, and Grohs, 2022). On the other hand, the backflow factor $\mathbf{F}$ directly interacts with the geometry-specific CASSCF envelopes and, therefore, cannot be shared across different geometries. Essentially, sharing nearly every trainable parameter, except for the final linear mapping $\mathbf{F}$, accounts for 95% of all network parameters. The optimization process is streamlined to initially generate a set of different geometries (around 10 to a 100) of the same molecule. During each step of the variational optimization phase, we select a geometry in a round-robin manner, keeping a substantial portion of the network weights fixed and only exchanging the backflow factor.

By using weight sharing, it is possible to achieve chemical accuracy by an order of magnitude faster than the traditional approach of optimizing each geometry independently (refer to Fig. 3.3 for details for an Ethene molecule). Chemical accuracy is usually defined as an error margin of 1.6 mHa relative to experimental results or highly accurate reference calculations. The results in Fig. 3.3 a. suggest that a large portion of the neural network can be identical without losing expressive power. Additionally, it highlights that our model contains general computational building blocks for approximating various wave functions. We hypothesized that these building blocks could also effectively generalize to molecular geometries that have not been previously used for optimizing the shared weights. Therefore, we use the pretrained weights from the shared optimization run to initialize the weights of a neural network ansatz for new geometries. The remaining 5% of the trainable parameters are randomly initialized. Fig. 3.3 b. shows that the computation can be accelerated when using those pretrained weights, and chemical accuracy is

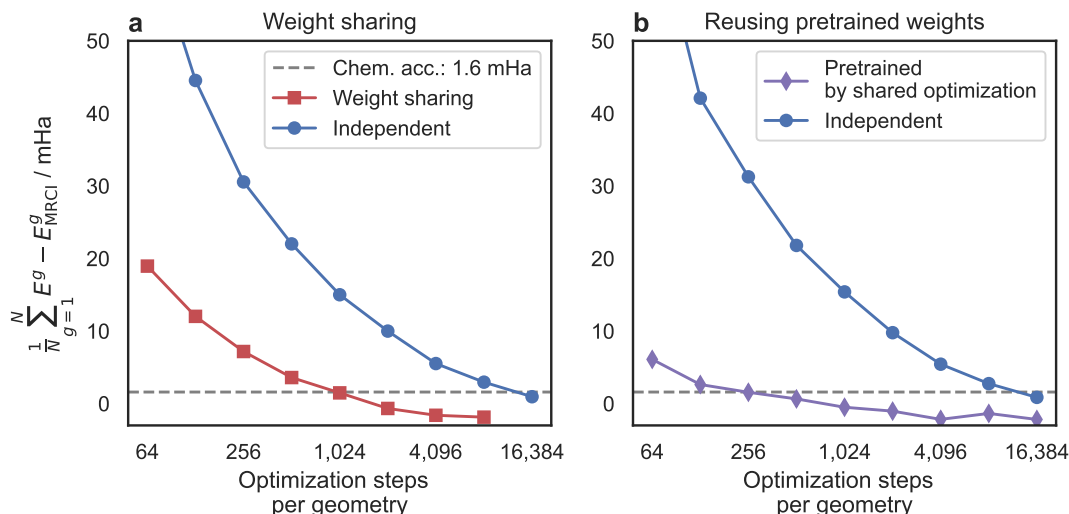


Figure 3.3: Each panel depicts the mean energy error over different geometries of an Ethene molecule with respect to high-accuracy MRCI reference calculations (Scherbela, Reisenhofer, Gerard, Marquetand, and Grohs, 2022) as a function of the number of epochs per geometry. Panel a: The red curve shows results when sharing 95% of the trainable weights on up to 30 distinct geometries and compares against independently calculated results (blue). Panel b: The purple curve shows results when reusing pretrained weights from a shared optimization calculation for 20 new geometries and compares against independent calculations (blue). The full figure can be found in Scherbela, Reisenhofer, Gerard, Marquetand, and Grohs (2022).

again reached by an order of magnitude faster. Additional experiments on various Hydrogen molecules can be found in the publication. Furthermore, the method is evaluated on a transition path of the Ethene molecule. As an additional observable, we also computed forces for a Hydrogen chain.

3.3 Towards a transferable fermionic neural wavefunction for molecules

As an extension of the previous paper, the following article presents an ansatz that allows us to learn a single neural network to represent the wave function of multiple compounds and geometrical confirmations. First, let us recapitulate the construction of the orbitals (see also Sec. 2.4):

$$\begin{aligned}\Phi_{ik} &= \varphi_k(\mathbf{r}_i, \mathbf{R}) \phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}) \\ \phi_k(\mathbf{r}_i, \{\mathbf{r}_\uparrow\}, \{\mathbf{r}_\downarrow\}) &= \langle \mathbf{F}_k, \mathbf{h}_i \rangle \in \mathbb{R}\end{aligned}$$

As discussed in the context of the weight-sharing approach (Sec. 3.2), it is important to note that the backflow factor $\mathbf{F}$ not only directly corresponds to the envelope functions, which are highly geometry-specific, especially in the case of CASSCF orbitals, but is also dependent on the number of electrons. The backflow factor is essentially a weight matrix of a linear layer of shape $n_{\text{el}} \times d_{\text{emb}}$ and trainable during optimization. Therefore, the number of trainable parameters in the neural network depends on the number of electrons. In general, this is not a bottleneck when considering, per calculation, a single molecule. However, when generalizing the ansatz to represent the wave function of multiple molecules with varying number of particles, it is not straightforward to share those weights. The following paper discusses an idea how to generalize the backflow mapping and trains a first base model on a chemically diverse molecule dataset that allows efficient reuse of the pretrained weights to new molecules and geometrical confirmations.

The overall idea of the approach is to use orbital descriptors from a computationally cheap Hartree Fock calculation in the form of expansion coefficients similar to the PauliNet-like orbitals (c.f. eq. (3.3)). However, instead of using Hartree Fock coefficients to initialize the weights c_{kIb} of the envelope functions, we use them as input to a neural network to map those descriptors to represent the backflow factor matrix. This allows us to generate from low-accuracy Hartree Fock orbitals highly accurate deep-learning-based orbitals. In other words, let $\mathbf{h}_i$ be again the electron embeddings using the FermiNet+GCN approach discussed in Sec. 3.1.

The orbitals are then computed as

$$\begin{aligned}\varphi(\mathbf{r}_i, \mathbf{R}_I, \mathbf{z}_{Ik}) &= \exp(-|\mathbf{r}_i - \mathbf{R}_I| g(\mathbf{z}_{Ik})) \\ \Phi_{ik} &= \sum_I^{n_{\text{nuc}}} \varphi(\mathbf{r}_i, \mathbf{R}_I, \mathbf{z}_{Ik}) \langle f(\mathbf{z}_{Ik}), \mathbf{h}_i \rangle\end{aligned}\quad (3.5)$$

with g and f being trainable functions (e.g. MLPs) and $\mathbf{z}_{Ik} \in \mathbb{R}^{n_{\text{basis}}}$, $k \in \{1, \dots, n_{\text{el}}\}$, $I \in \{1, \dots, n_{\text{nuc}}\}$ denoting the expansion coefficients from a Hartree Fock calculation with n_{basis} basis functions. To increase the expressivity of the descriptors $\mathbf{z}_{Ik}$ and to allow better transferability between physical systems, we perform orbital localization and message passing with respect to the nuclei index using graph convolution. We refer to these orbitals in eq. (3.5) as *transferable atomic orbitals* (TAOs) in the subsequent discussion.

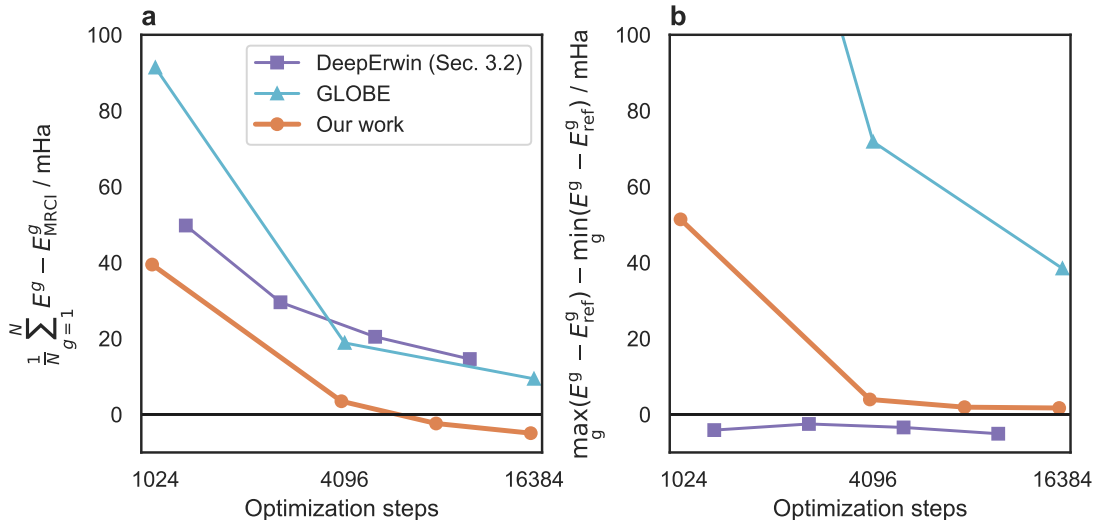


Figure 3.4: Transferring a pretrained wave function model on Methane (CH_4) to Ethene (C_2H_4). DeepErwin depicts the method described in Sec. 3.2 ("Pretrained by shared optimization"). GLOBE refers to a concurrent work, which is also fully universal in the sense of a constant parameter count across varying molecule sizes (Gao and Günnemann, 2023a). Panel a depicts the mean energy error over different geometries g of an Ethene molecule with respect to highly accurate MRCI reference calculations as a function of optimization steps. Panel b highlights the relative energy error. The full figure can be found in Scherbela, Gerard, and Grohs (2024).

This approach fundamentally maintains a constant parameter count and can be used to train a neural network to represent the wave functions of a diverse set of

molecules. To assess its effectiveness, we conduct several experiments, with an initial test focusing on determining whether a network, once trained on geometries of a single molecule, could adapt to larger molecules. In Fig. 3.4, we first train a model on multiple conformations of a Methane molecule (10 electrons) and subsequently apply this trained neural network wave function to the transition path of an Ethene molecule (16 electrons). Compared to previously published methods, including the method detailed in Sec. 3.2, our approach outperforms the existing baseline and achieves lower energy predictions than those from the highly accurate MRCI reference. As a second experiment, we pretrain a single neural network ansatz on a set of molecules with 18 molecules in total. Each molecule contains up to two heavy atoms such as Oxygen, Nitrogen, or Carbon. This dataset comprises 360 geometries with 20 distinct geometries per molecule. Subsequently, we assess the performance of this pretrained model on a set of four distinct molecules, each with three heavy atoms. Notably, the target molecules are up to 50% larger in terms of the number of heavy atoms compared to the molecules in the training set. In Fig. 3.5, we contrast our method with the conventional

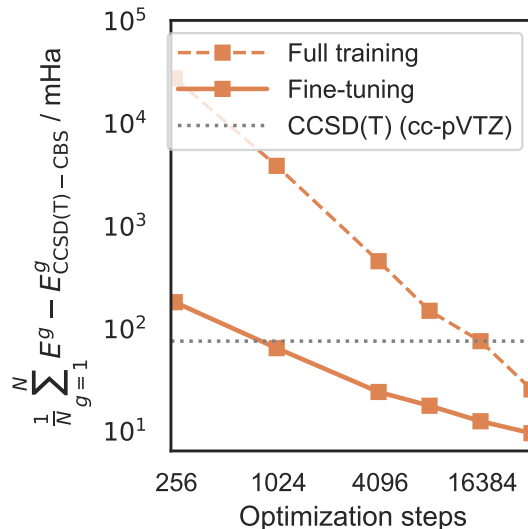


Figure 3.5: Pretraining a neural network-based wave function on 18 molecules with up to two heavy atoms (including Oxygen, Nitrogen, and Carbon) with 20 geometries each and transferring the model to a set of molecules with up to three heavy atoms (70 geometries in total). The plot shows the mean error with respect to CCSD(T) in a complete basis set limit (CBS) as a function of optimization steps. It compares the pretrained model ("Fine-tuning") against a randomly initialized model ("Full training"). The full figure can be found in Scherbela, Gerard, and Grohs (2024).

approach, which begins with a randomly initialized wave function ansatz (termed "Full-training"). By leveraging a model pretrained on our dataset, we achieve 20x faster the accuracy of CCSD(T) in a fixed basis set. This marks the first instance of an ab-initio wave function model being trained on a diverse set of molecules and effectively transferred to new molecules. Additionally, our paper investigates the impact of various factors, such as the size of the pretraining dataset, the number of trainable parameters, and the extent of necessary pretraining steps. These investigations are crucial in taking the first step towards developing a foundation model for the Schrödinger equation.

3.4 Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions

The concluding paper in this series is based directly on the approach outlined in Sec. 3.3. In the previous work, we introduced transferable atomic orbitals, allowing efficient training across various molecules and the transfer of a pretrained model to new molecules. Extending this idea, we improve large parts of the wave function architecture to scale to molecules up to 7 heavy atoms.

Instead of depending on iterative methods like Hartree Fock, we introduce an orbital prediction model that takes only the geometry of the molecule $\{(\mathbf{R}, \mathbf{Z})\}$ as input and directly predicts orbital descriptors $\mathbf{z}_{kI}$. We adapt the orbital prediction model known as PhisNet, proposed by Unke et al. (2021), which utilizes SE(3)-equivariant operations and thus making our wave function model fully end-to-end machine-learnable. Furthermore, we enhance the embedding architecture by employing the graph neural network architecture described in Sec. 2.4, moving away from the FermiNet+GCN approach. As a final step, we refine the molecule dataset for pretraining the neural wave function. This involves increasing the dataset to include 98 distinct molecules, each having up to four heavy atoms. Additionally, we eliminate the need to rely on a discrete set of geometries for each molecule, during the optimization process. Instead, we introduce a continuous sampling scheme for the geometries based on normal mode distortion and data augmentation. The augmentation involves rotating the entire geometry of the molecule by a random rotation.

We conduct an ablation study to gain a deeper understanding of the accuracy contributions from individual components. Starting from the initial architecture outlined in Sec. 3.3 we add sequentially new improvements (c.f. Fig. 3.6). We assess the performance of these pretrained models in two scenarios: first, in a zero-shot regime, where we apply the pretrained model directly to new molecules without additional variational optimization, and second, after 4000 variational optimization

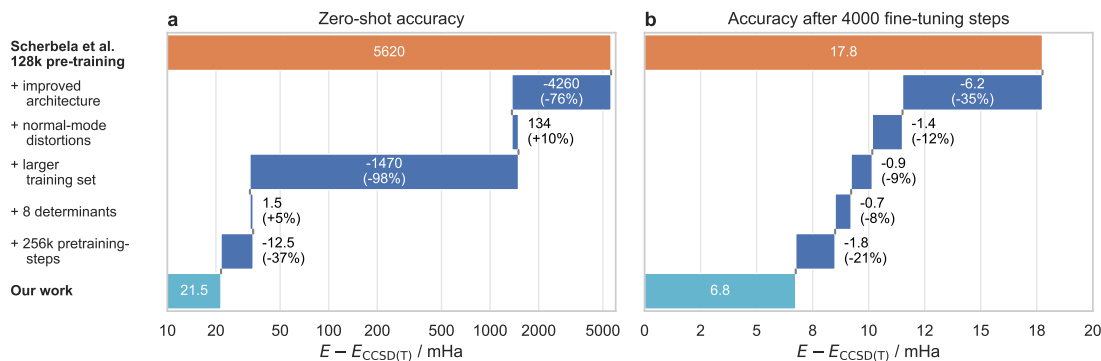


Figure 3.6: Ablation study comparing the transferable atomic orbital approach introduced in Sec. 3.3 with the improved version ("Our work") on a set of geometries of molecules with 3 heavy atoms. Panel a shows the accuracy gain for zero-shot predictions (no additional optimization is performed). Panel b shows the accuracy after additional 4000 fine-tuning steps. The full figure can be found in Scherbela, Gerard, and Grohs (2023).

steps also referred to as fine-tuning steps. The test set holds four molecules, each with three heavy atoms measured as the number of non-hydrogen atoms containing Oxygen, Nitrogen, or Carbon. In both scenarios, a significant portion of the improvement can be attributed to the enhancements in the embedding function. A larger pretraining dataset notably boosts the zero-shot performance.

In addition, we assess the performance of the best pretrained wave function model, with all the mentioned enhancements, across various molecules categorized by their size. Each test set comprises four randomly perturbed molecules, with the largest test set accommodating molecules with up to 7 heavy atoms. We made sure that no molecule from the test set was part of the training set. To evaluate the performance, we reexamine the two scenarios employed in the ablation study. Additionally, we compare the energy error over optimization steps against other state-of-the-art DL-VMC methods, including our own approach introduced in Section 3.1 and an attention-based architecture known as PsiFormer (von Glehn, Spencer, and Pfau, 2023) (refer to Fig. 3.7). We improve the zero-shot performance by two orders of magnitude (c.f. Fig. 3.7 a.). Further, we are reaching without any additional optimization steps accuracy levels comparable to CCSD(T) in a fixed basis set. We find that fine-tuning a pretrained wave function model can be more cost-efficient than a traditional method. For a molecule with five heavy atoms, we require only 9.5 node-hours compared to a CCSD(T)-4Z calculation with 10.5 node-hours or 53 node-hours using the state-of-the-art DL-VMC method PsiFormer (von Glehn,

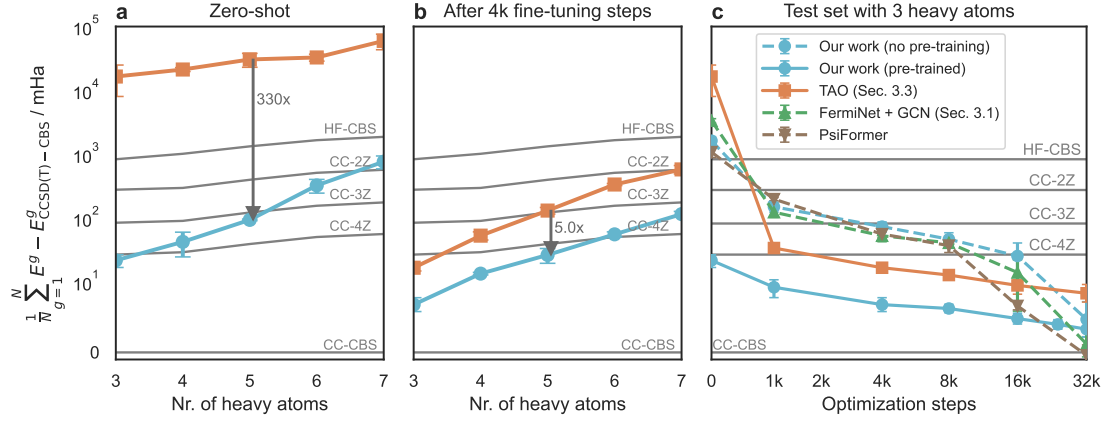


Figure 3.7: Mean energy error compared to CCSD(T) in a complete basis set limit (CBS) when reusing a pretrained wave function model. Panel a depicts the results on a variety of molecules grouped into molecule size defined by number of heavy atoms for the zero-shot regime. Panel b shows the setting when using additional 4000 fine-tuning steps. In both panels we compare against the method introduced in Sec. 3.3. Panel c shows the mean error as a function of optimization steps vs. state-of-the-art DL-VMC methods. PsiFormer denotes an attention-based architecture (von Glehn, Spencer, and Pfau, 2023). The full figure can be found in Scherbela, Gerard, and Grohs (2023).

Spencer, and Pfau, 2023). Fig. 3.7 c. highlights that we significantly reach faster certain levels of accuracy but, in the long run, still lose accuracy compared to state-of-the-art DL-VMC approaches. In the remaining publication, we discuss limitations regarding the relative energies.

4 Discussion

DL-VMC in first quantization is a highly accurate approach for achieving ground-state energy predictions. The architecture introduced by Pfau et al. (2020) has been established as the method of choice in this research area. Although it was already originally capable of reaching accuracy levels comparable to the gold-standard conventional approach CCSD(T), a lot of improvements to the architecture were proposed to increase the accuracy even further (Gerard et al., 2022; Ren et al., 2023; Spencer et al., 2020; von Glehn, Spencer, and Pfau, 2023) and reduce the computational cost (Li et al., 2023). Besides the improvements to the architecture, the method was not only applied to the prediction of ground-state energies for gas-phase molecules but also to excited states (Entwistle et al., 2023; Pfau et al., 2023), forces (Qian et al., 2022; Scherbela et al., 2022), ground-state energies for solids (Cassella et al., 2023; Pescia et al., 2023) and real-solids (Li, Li, and Chen, 2022).

The primary focus of this thesis was to enhance neural network architectures and introduce an approach for reducing computational overhead by devising a novel technique for learning a generalized wave function. We highlighted the potential detrimental consequences of introducing prior physical knowledge into the architecture by incorporating computational methods like Hartree Fock or CASSCF. Nevertheless, certain physical priors remain essential, such as antisymmetrization, to find the ground-state wave function. Furthermore, supervised pretraining against a Hartree Fock reference proves advantageous for overall numerical stability. In contrast to our work, a recent preprint by Pescia et al. (2023), introduced a novel neural network architecture for the homogenous electron gas (HEG). The neural wave function model exploits attention mechanisms and incorporates Hartree Fock orbitals, denoting a notable improvement over a FermiNet-like architecture. The results underscore the high complexity involved in constructing an accurate neural network-based wave function, where slight differences in the task (e.g. Hamiltonian for HEG vs. molecules) can have a substantial impact on the design choices within the neural network architecture. Another impactful prior in other areas of *AI for Science* is the use of SE(3) symmetries, particularly for supervised machine learning potentials (Batatia et al., 2022; Batzner et al., 2022). Notably, no one has yet efficiently integrated such priors into the neural wave function. Ideas like an equivariant coordinate system, such as the ones proposed by Gao and Günnemann (2022) and Gerard et al. (2022), are a possible way forward to such problems, but as discussed in Gao and Günnemann (2022), certain types of edge cases hinder

this idea to be a complete solution to the problem.

The second part of the thesis introduced the construction of a generalized wave function. We successfully established the first neural network-based wave function capable of training across a wide range of molecules and their various geometrical confirmations. This allowed us to efficiently transfer a pretrained model to entirely new molecules using either no additional optimization steps (zero-shot) or a short fine-tuning phase. In a concurrent effort, Gao and Günnemann (2023a) constructed a generalized wave function by reparametrization of the orbitals combined with heuristics. However, transferring their pretrained wave function has proven difficult. In our case, although the results are promising, we want to also emphasize current limitations. Specifically for relative energies, additional fine-tuning steps are required to receive an accurate prediction, and zero-shot evaluations remain a challenging task. Additionally, despite the reduced optimization time, scaling to larger systems still proves difficult and resource-intensive with GPU days on sophisticated hardware. Improvements, such as the work by Li et al. (2023), proposing an efficient way to compute the laplacian for neural network-based wave functions, show a way forward to overcome those limitations. Another potential area for improvement with a similar impact could be the development of a more efficient method for antisymmetrizing the wave function model, which currently is due to the determinant one of the leading scaling factors. While methods proposed by Richter-Powell et al. (2023) show promise, they are currently at the proof-of-concept stage.

In conclusion, the present thesis contributes to improvements of previously established neural network-based approaches for solving the electronic Schrödinger equation for gas-phase molecules. We introduced transferable atomic orbitals and made the first step towards a pretrained base model for neural network-based wave functions. Through continued enhancements of the embedding architecture and orbital descriptors, alongside dedicated engineering efforts, it might become feasible to train a wave function model on hundreds of molecules. This would further improve our ansatz and potentially also allow for an efficient transfer to relative energies in a zero-shot manner.

References

- Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoaj Bikowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan (2022). “Flamingo: a Visual Language Model for Few-Shot Learning.” In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., pp. 23716–23736.
- Amari, Shun-ichi (1998). “Natural Gradient Works Efficiently in Learning.” In: *Neural Computation* 10.2, pp. 251–276. DOI: 10.1162/089976698300017746.
- Barone, Vincenzo, Malgorzata Biczysko, Julien Bloino, and Cristina Puzzarini (2013). “Accurate structure, thermodynamic and spectroscopic parameters from CC and CC/DFT schemes: the challenge of the conformational equilibrium in glycine.” In: *Physical Chemistry Chemical Physics* 15.25. The Royal Society of Chemistry, pp. 10094–10111. DOI: 10.1039/C3CP50439E.
- Batatia, Ilyes, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor N. C. Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi (2022). *The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials*. arXiv: 2205.06643.
- Batzner, Simon, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky (2022). “E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials.” In: *Nature Communications* 13.1, p. 2453. DOI: 10.1038/s41467-022-29939-5.
- Becca, Federico and Sandro Sorella (2017). *Quantum Monte Carlo Approaches for Correlated Systems*. Cambridge University Press. DOI: 10.1017/9781316417041.
- Cassella, Gino, Halvard Sutterud, Sam Azadi, N. D. Drummond, David Pfau, James S. Spencer, and W. M. C. Foulkes (2023). “Discovering Quantum Phase Transitions with Fermionic Neural Networks.” In: *Phys. Rev. Lett.* 130 (3), p. 036401. DOI: 10.1103/PhysRevLett.130.036401.
- Casula, Michele, Claudio Attaccalite, and Sandro Sorella (2004). “Correlated geminal wave function for molecules: An efficient resonating valence bond approach.”

- In: *The Journal of Chemical Physics* 121.15, pp. 7110–7126. DOI: 10.1063/1.1794632.
- Clark, Bryan K., Miguel A. Morales, Jeremy McMinis, Jeongnim Kim, and Gustavo E. Scuseria (2011). “Computing the energy of a water molecule using multide-terminants: A simple, efficient algorithm.” In: *The Journal of Chemical Physics* 135.24. American Institute of Physics, p. 244105. DOI: 10.1063/1.3665391.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv: 1810.04805.
- Entwistle, Mike. T., Zeno Schätzle, Paola A. Erdman, Jan Hermann, and Frank Noé (2023). “Electronic excited states in deep variational Monte Carlo.” In: *Nature Communications* 14.1, p. 274. DOI: 10.1038/s41467-022-35534-5.
- Foulkes, W. M. C., L. Mitas, R. J. Needs, and G. Rajagopal (2001). “Quantum Monte Carlo simulations of solids.” In: *Rev. Mod. Phys.* 73 (1), pp. 33–83. DOI: 10.1103/RevModPhys.73.33.
- Gao, Nicholas and Stephan Günnemann (2022). “Ab-Initio Potential Energy Sur-faces by Pairing GNNs with Neural Wave Functions.” In: *International Confer-ence on Learning Representations*.
- (2023a). “Generalizing Neural Wave Functions.” In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 10708–10726.
 - (2023b). “Sampling-free Inference for Ab-Initio Potential Energy Surface Net-works.” In: *The Eleventh International Conference on Learning Representations*.
- Gerard, Leon, Michael Scherbela, Philipp Marquetand, and Philipp Grohs (2022). “Gold-Standard Solutions to the Schrödinger Equation Using Deep Learning: How Much Physics Do We Need?” In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., pp. 10282–10294.
- Han, Jiequn, Linfeng Zhang, and Weinan E (2019). “Solving many-electron Schrödinger equation using deep neural networks.” In: *J. Comput. Phys.* 399, p. 108929. DOI: 10.1016/j.jcp.2019.108929.
- Hermann, Jan, Zeno Schätzle, and Frank Noé (2020). “Deep-Neural-Network So-lution of the Electronic Schrödinger Equation.” In: *Nature Chemistry* 12.10, pp. 891–897. DOI: 10.1038/s41557-020-0544-y.

- Inui, Koji, Yasuyuki Kato, and Yukitoshi Motome (2021). “Determinant-free fermionic wave function using feed-forward neural networks.” In: *Phys. Rev. Res.* 3 (4), p. 043126. DOI: 10.1103/PhysRevResearch.3.043126.
- Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980.
- Li, Ruichen, Haotian Ye, Du Jiang, Xuelan Wen, Chuwei Wang, Zhe Li, Xiang Li, Di He, Ji Chen, Weiluo Ren, and Liwei Wang (2023). *Forward Laplacian: A New Computational Framework for Neural Network-based Variational Monte Carlo*. arXiv: 2307.08214.
- Li, Xiang, Zhe Li, and Ji Chen (2022). “Ab initio calculation of real solids via neural network ansatz.” In: *Nature Communications* 13.1, p. 7895. DOI: 10.1038/s41467-022-35627-1.
- Martens, James and Roger Grosse (2015). *Optimizing Neural Networks with Kronecker-factored Approximate Curvature*. arXiv: 1503.05671.
- Nemec, Norbert, Michael D. Towler, and R. J. Needs (2010). “Benchmark all-electron ab initio quantum Monte Carlo calculations for small molecules.” In: *The Journal of Chemical Physics* 132.3. American Institute of Physics, p. 034111. DOI: 10.1063/1.3288054.
- Pescia, Gabriel, Jannes Nys, Jane Kim, Alessandro Lovato, and Giuseppe Carleo (2023). *Message-Passing Neural Quantum States for the Homogeneous Electron Gas*. arXiv: 2305.07240.
- Pfau, David, Simon Axelrod, Halvard Sutterud, Ingrid von Glehn, and James S. Spencer (2023). *Natural Quantum Monte Carlo Computation of Excited States*. arXiv: 2308.16848.
- Pfau, David, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes (2020). “Ab Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks.” In: *Phys. Rev. Res.* 2.3, p. 033429. DOI: 10.1103/PhysRevResearch.2.033429.
- Qian, Yubing, Weizhong Fu, Weiluo Ren, and Ji Chen (2022). “Interatomic force from neural network based variational quantum Monte Carlo.” In: *The Journal of Chemical Physics* 157.16, p. 164104. DOI: 10.1063/5.0112344.
- Ren, Weiluo, Weizhong Fu, Xiaojie Wu, and Ji Chen (2023). “Towards the ground state of molecules via diffusion Monte Carlo on neural networks.” In: *Nature Communications* 14.1, p. 1860. DOI: 10.1038/s41467-023-37609-3.

- Richter-Powell, Jack, Luca Thiede, Alán Asparu-Guzik, and David Duvenaud (2023). *Sorting Out Quantum Monte Carlo*. arXiv: 2311.05598.
- Schätzle, Zeno, Bernát Szabó, Matej Mezera, Jan Hermann, and Frank Noé (2023). “DeepQMC: An open-source software suite for variational optimization of deep-learning molecular wave functions.” In: *The Journal of Chemical Physics* 159.9, p. 094108. DOI: 10.1063/5.0157512.
- Scherbela, Michael, Leon Gerard, and Philipp Grohs (2023). “Variational Monte Carlo on a Budget - Fine-tuning pre-trained Neural Wavefunctions.” In: *Thirty-seventh Conference on Neural Information Processing Systems*.
- (2024). “Towards a transferable fermionic neural wavefunction for molecules.” In: *Nature Communications* 15.1, p. 120. DOI: 10.1038/s41467-023-44216-9.
- Scherbela, Michael, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs (2022). “Solving the Electronic Schrödinger Equation for Multiple Nuclear Geometries with Weight-Sharing Deep Neural Networks.” In: *Nature Computational Science* 2.5, pp. 331–341. DOI: 10.1038/s43588-022-00228-x.
- Seth, P., P. López Ríos, and R. J. Needs (2011). “Quantum Monte Carlo study of the first-row atoms and ions.” In: *The Journal of Chemical Physics* 134.8, p. 084105. DOI: 10.1063/1.3554625.
- Slater, J. C. (1929). “The Theory of Complex Spectra.” In: *Phys. Rev.* 34 (10), pp. 1293–1322. DOI: 10.1103/PhysRev.34.1293.
- Spencer, James S., David Pfau, Aleksandar Botev, and W. M. C. Foulkes (2020). *Better, Faster Fermionic Neural Networks*. arXiv: 2011.07125.
- Szabo, Attila and Neil S. Ostlund (1996). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications. ISBN: 9780486691862.
- Umrigar, Cyrus J., M. P. Nightingale, and K. J. Runge (1993). “A diffusion Monte Carlo algorithm with very small timestep errors.” In: *The Journal of Chemical Physics* 99.4, pp. 2865–2890. DOI: 10.1063/1.465195.
- Unke, Oliver, Mihail Bogojeski, Michael Gastegger, Mario Geiger, Tess Smidt, and Klaus-Robert Müller (2021). “SE(3)-Equivariant Prediction of Molecular Wavefunctions and Electronic Densities.” In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 14434–14447.
- von Glehn, Ingrid, James S Spencer, and David Pfau (2023). “A Self-Attention Ansatz for Ab-initio Quantum Chemistry.” In: *The Eleventh International Conference on Learning Representations*.

Zhang, Xuan, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji (2023). *Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems*. arXiv: 2307.08423.

Publications

I	Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?	51
II	Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks	71
III	Towards a transferable fermionic neural wavefunction for molecules . . .	91
IV	Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions	109

I Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?

Comments

E-Print: [arXiv:2205.09438](https://arxiv.org/abs/2205.09438)[cs.LG]

Conference: Poster presentation at NeurIPS 2022.

Authorship: L.G. and M.S. share first authorship.

Contribution

P.G., P.M., M.S. and L.G. conceived the project. M.S. and L.G. developed the network architecture with equal contributions and performed the numerical experiments. M.S. developed the local invariant input features and the orbital envelope initialization. L.G. designed and ran experiments regarding the impact of physical prior knowledge. M.S. and L.G. wrote the manuscript with equal contribution, with input from P.G. P.M. performed reference calculations. P.G. obtained funding and supervised the project.

Bibliographic Information

Gerard, Leon, Michael Scherbela, Philipp Marquetand, and Philipp Grohs (2022). “Gold-Standard Solutions to the Schrödinger Equation Using Deep Learning: How Much Physics Do We Need?” In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., pp. 10282–10294.

Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?

Leon Gerard^{†,a,*}, Michael Scherbela^{†,*}, Philipp Marquetand^{†,§}, and Philipp Grohs^{†,‡,¶}

[†]Research Network Data Science @ Uni Vienna, Kolingasse 14-16, A-1090 Vienna, Austria

[‡]Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria

[§]Institute of Theoretical Chemistry, Faculty of Chemistry, University of Vienna, Währinger Straße 17, 1090 Vienna, Austria

[¶]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenbergerstrasse 69, 4040 Linz, Austria

^aleon.gerard@univie.ac.at

*These authors contributed equally

Abstract

Finding accurate solutions to the Schrödinger equation is the key unsolved challenge of computational chemistry. Given its importance for the development of new chemical compounds, decades of research have been dedicated to this problem, but due to the large dimensionality even the best available methods do not yet reach the desired accuracy. Recently the combination of deep learning with Monte Carlo methods has emerged as a promising way to obtain highly accurate energies and moderate scaling of computational cost. In this paper we significantly contribute towards this goal by introducing a novel deep-learning architecture that achieves 40-70% lower energy error at 6x lower computational cost compared to previous approaches. Using our method we establish a new benchmark by calculating the most accurate variational ground state energies ever published for a number of different atoms and molecules. We systematically break down and measure our improvements, focusing in particular on the effect of increasing physical prior knowledge. We surprisingly find that increasing the prior knowledge given to the architecture can actually decrease accuracy.

1 Introduction

The challenge of the Schrödinger Equation Accurately predicting properties of molecules and materials is of utmost importance for many applications, including the development of new materials or pharmaceuticals. In principle, any property of any molecule can be calculated from its wavefunction, which is obtained by solving the Schrödinger equation. In practice, computing accurate wavefunctions and corresponding energies is computationally extremely difficult for two reasons: First, the wavefunction is a high-dimensional function, depending on all coordinates of all electrons, subjecting most methods to the curse of dimensionality. Second, the required level of accuracy is extremely high. While total energies of small molecules are typically hundreds of Hartrees, the chemically relevant energy differences are on the order of 1 milli-Hartree as depicted in Fig. 1. Decades of research have produced a plethora of methods, which all require a trade-off between accuracy and computational cost: On one end of the spectrum are approximate methods such as Hartree-Fock (HF) or Density Functional Theory (DFT), which was awarded the Nobel prize in 1998. These methods can treat thousands of particles but can often only crudely approximate chemical properties. On the other end of the spectrum are "gold-standard" methods such as FCI (Full Configuration Interaction) or CCSD(T) (Coupled Clusters Singles Doubles (Perturbative Triples))

which yield energies that often closely agree with experiments, but can only treat up to 100 particles. Despite all these efforts, even for small molecules there do currently not exist highly accurate energy calculations. A 2020 benchmark of state-of-the-art methods for the benzene molecule found a spread of 4 mHa across different methods [1] – as a matter of fact, our results show that the absolute energies calculated in [1] are off by at least 600 mHa, due to the small basis set used in their calculations. An important characteristic of a method is the type of approximation being made: Hartree-Fock or

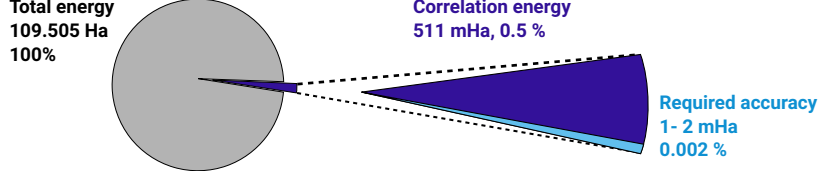


Figure 1: Conceptual visualization of the required accuracies on the example of an N_2 molecule: For chemical applications, total energies must achieve accuracies of $\sim 99.998\%$.

FCI are "variational", meaning that their predicted energies at least upper-bound the ground-truth energy. Since a lower energy is always guaranteed to be a closer approximation of the true energy, this makes assessment of these methods straight-forward. In contrast, CCSD(T) or DFT do not have any guarantees on the accuracy of their results. The approximations resulting from such methods, while working well for many common systems, often fail for chemically challenging situations such as breaking of chemical bonds [2, 3].

Deep-learning-based variational Monte Carlo Combining deep learning and Monte Carlo methods has recently emerged as a promising new approach for solving the Schrödinger equation [4, 5]. These methods offer high accuracy, moderate scaling of computational cost with system size and obey the variational principle. Within a few years deep-learning-based methods have managed to outperform conventional high-accuracy methods for many different molecules, potentially defining a new gold-standard for high-accuracy solutions. In the Born-Oppenheimer approximation a molecule, consisting of n_{nuc} nuclei and n_{el} electrons, is fully described by its Hamiltonian in atomic units

$$H = -\frac{1}{2} \sum_i \nabla_{\mathbf{r}_i}^2 + \sum_{i>j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{I>J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{i,I} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|}.$$

Here $\mathbf{R}_I$, Z_I , $I \in \{1, \dots, n_{\text{nuc}}\}$ denote the coordinates and charges of the nuclei, $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_{\uparrow}}, \dots, \mathbf{r}_{n_{\text{el}}}) \in \mathbb{R}^{3 \times n_{\text{el}}}$ denotes the set of n_{el} Cartesian electron coordinates differentiated between $n_{\uparrow}$ spin-up and $n_{\downarrow}$ spin-down electrons. We define the inter-particle vectors $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{r}_{iJ} = \mathbf{r}_i - \mathbf{R}_J$. All properties of the molecule depend on the wavefunction $\psi(\mathbf{r})$, which must fulfill the antisymmetry constraint: $\psi(\mathcal{P}\mathbf{r}) = -\psi(\mathbf{r})$ for any permutation $\mathcal{P}$ of two electrons with the same spin [6]. The wavefunction ψ can be found as the solution to the Schrödinger equation $H\psi = E_0\psi$ with the ground-state energy and smallest eigenvalue E_0 . By the Rayleigh-Ritz principle [7], the ground-state energy and the corresponding wavefunction can be found through minimization of the loss

$$\mathcal{L}(\psi_\theta) = \mathbb{E}_{\mathbf{r} \sim \psi_\theta^2(\mathbf{r})} \left[\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})} \right] \geq E_0 \quad (1)$$

for a trial wavefunction ψ_θ , parameterized by parameters θ . The trial function ψ_θ is represented by a neural network and typically has the form

$$\psi_\theta(\mathbf{r}) = \sum_{d=1}^{n_{\text{det}}} \det[\Lambda_{ki}^d(\mathbf{r}) \Omega_k^{\alpha_i}(\mathbf{r}_i)]_{k,i=1,\dots,n_{\text{el}}} \quad (2)$$

with $\Lambda_{ki}^d : \mathbb{R}^{3 \times n_{\text{el}}} \rightarrow \mathbb{R}$, $\Omega_k^d : \mathbb{R}^3 \rightarrow \mathbb{R}$, $\alpha_i \in \{\uparrow, \downarrow\}$, $i \in \{1, \dots, n_{\text{el}}\}$, $k \in \{1, \dots, n_{\text{el}}\}$. Each determinant is taken over a $n_{\text{el}} \times n_{\text{el}}$ matrix, with row-indices k running over orbitals and column-indices i running over electrons. The determinant enforces antisymmetry, Ω_k^d are envelope functions enforcing the boundary condition $\lim_{|\mathbf{r}| \rightarrow \infty} \psi_\theta(\mathbf{r}) = 0$, and Λ_{ki}^d are neural networks. The local energy $\frac{H\psi}{\psi}$ can be evaluated using automatic differentiation and the loss can be minimized by gradient based methods. The computation of the expectation value in eq. 1 over the high-dimensional space $\mathbb{R}^{3 \times n_{\text{el}}}$ is done using Monte Carlo integration by sampling electron coordinates $\mathbf{r}$ distributed according to ψ_θ^2 using the Metropolis-Hastings [8] algorithm. A thorough discussion of deep-learning-based variational Monte Carlo (DL-VMC) can be found in [9].

Related work Two major neural network architectures and their extensions have emerged throughout literature: PauliNet [9] and FermiNet [10]. PauliNet puts emphasis on maximizing physical prior knowledge, by focusing on the the envelope function. They use the output of CASSCF (Complete Active Space Self Consistent Field, a sophisticated conventional quantum-chemistry method) as Ω and use a relatively small ($\sim 100k$ weights) neural network for Λ . FermiNet on the other hand uses a simple exponential function as envelope Ω and uses a large ($\sim 700k$ weights) neural network for Λ . Both approaches have been applied with great success to many different systems and properties, such as energies of individual molecules [4, 10, 9], ionization energies [10], potential energy surfaces [11, 12], forces [11], excited states [13], model-systems for solids [14, 15] and actual solids [16]. Several approaches have been proposed to increase accuracy or decrease computational cost, most notably architecture simplifications [17], alternative antisymmetric layers [18], effective core potentials [19] and Diffusion Monte Carlo (DMC) [20, 21]. FermiNet commonly reaches lower (i.e. more accurate) energies than PauliNet[10], but PauliNet has been observed to converge faster [12]. It has been proposed [9] that combining the embedding of FermiNet and the physical prior knowledge of PauliNet could lead to a superior architecture.

Our contribution In this work we present the counter-intuitive observation that the opposite approach might be more fruitful. By combining a PauliNet-like neural network embedding with the envelopes of FermiNet and adding several improvements to the embedding, input features, and initialization of parameter (Sec. 2), we obtain the currently best neural network architecture for the numerical solution of the electronic Schrödinger equation. Combining our new architecture with VMC we establish a new benchmark by calculating the most accurate variational ground state energies ever published for a number of different atoms and molecules - both when comparing to deep-learning-based methods, as well as when comparing to classical methods (Sec. 3). Across systems we reduce energy errors by 40-100% and achieve these results with 3-4x fewer optimization epochs compared to FermiNet. In Sec. 4 we systematically break down which changes cause these improvements. We hypothesize that including too much physical prior knowledge can actually hinder optimization and thus deteriorate accuracy – we provide ample experimental evidence in Sec. 5.

2 Improved approach

Similar to FermiNet, our architecture expresses Λ_{ki}^d as a linear combination of high-dimensional electron embeddings $\mathbf{h}_i^L$, and the envelopes $\Omega_k^{d\alpha_i}$ as a sum of exponential functions

$$\Lambda_{ki}^d(\mathbf{r}) = W_k^{d\alpha_i} \mathbf{h}_i^L \quad \Omega_k^{d\alpha_i}(\mathbf{r}_i) = \sum_{I=1}^{n_{\text{nuc}}} \pi_{kI}^{d\alpha_i} \exp(-\omega_{kI}^{d\alpha_i} |\boldsymbol{\rho}_{iI}|), \quad (3)$$

where $W_k^{d\alpha_i}, \pi_{kI}^{d\alpha_i}, \omega_{kI}^{d\alpha_i}$ are trainable parameters and we enforce $\omega_{kI}^{d\alpha_i} \geq 0$. We compute these embeddings $\mathbf{h}_i^L$ by first transforming the inputs $\mathbf{R}_I, \mathbf{r}_i$ into feature vectors

$$\mathbf{h}_i^0 = \left[|\boldsymbol{\rho}_{iI}|, \tilde{\boldsymbol{\rho}}_{iI} \right]_{I \in \{1, \dots, n_{\text{nuc}}\}} \quad \mathbf{v}_{iI}^0 = \left[|\boldsymbol{\rho}_{iI}|, \tilde{\boldsymbol{\rho}}_{iI} \right] \quad \mathbf{g}_{ij}^0 = |\mathbf{r}_{ij}|$$

where $[\cdot]$ denotes the concatenation operation and then applying L iterations of an embedding network (Fig. 2a). The local difference vectors $\tilde{\boldsymbol{\rho}}_{iI}$ are obtained by applying rotation matrices onto $\boldsymbol{\rho}_{iI}$ as described in Sec. 2.2.

2.1 Convolutional layers in embedding

Our embedding network uses four residual neural network streams (Fig. 2b): A primary one-electron stream that embeds a single electron, and three auxiliary streams modelling the two-particle-interactions (electrons with same spins, electrons with different spin, and electron-ion).

$$\mathbf{h}_i^{l+1} = \mathbf{A}_{\text{one}}^l(\mathbf{f}_i^l) + \mathbf{h}_i^l \quad \mathbf{g}_{ij}^{l+1} = \mathbf{A}_{\sigma_{ij}}^l(\mathbf{g}_{ij}^l) + \mathbf{g}_{ij}^l \quad \mathbf{v}_{iI}^{l+1} = \mathbf{A}_{\text{nuc}}^l(\mathbf{v}_{iI}^l) + \mathbf{v}_{iI}^l \quad (4)$$

Here l denotes the embedding iteration, $\mathbf{A}^l$ denote fully connected neural networks, and $\mathbf{g}_{ij}^l, \mathbf{v}_{iI}^l$ denote electron-electron- and electron-nucleus-embeddings. We use $\sigma_{ij} = \text{'same'}$ for same-spin

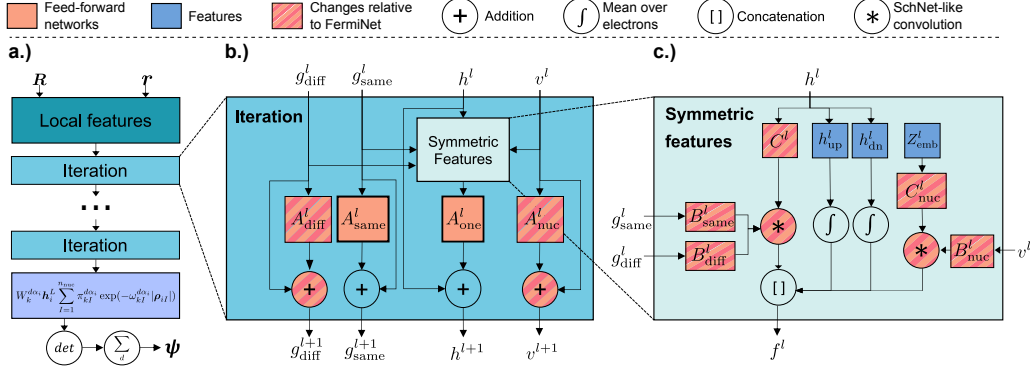


Figure 2: Our architecture: a) High-level overview b) One single embedding iteration c) Sub-block of assembling symmetric features

pairs of electrons and $\sigma_{ij} = \text{'diff'}$ for pairs of electrons with different spin. Similar to FermiNet, in each iteration we assemble the input f_i^l to the primary stream from the auxiliary streams (Fig. 2c):

$$f_i^l = \left[h_i^l, \frac{1}{n_{\uparrow}} \sum_{j=1}^{n_{\uparrow}} h_j^l, \frac{1}{n_{\downarrow}} \sum_{j=1+n_{\uparrow}}^{n_{el}} h_j^l, s_i^{l,\text{el}}, s_i^{l,\text{nuc}} \right]. \quad (5)$$

Inspired by the success of SchNet [22] and the efficiency of the PauliNet embedding, we use the sum of element-wise multiplication ($\odot$), effectively forming a convolution, to aggregate the auxiliary two-particle streams:

$$s_i^{l,\text{el}} = \sum_{j=1}^{n_{el}} B_{\sigma_{ij}}^l(g_{ij}^l) \odot C_{\sigma_{ij}}^l(h_j^l) \quad s_i^{l,\text{nuc}} = \sum_{I=1}^{n_{ion}} B_{\text{nuc}}^l(v_{iI}^l) \odot C_{\text{nuc}}^l(Z_I^{\text{emb}}) \quad (6)$$

Eq. 5 and 6 form the core of the architecture and are the key difference between FermiNet, PauliNet and our architecture. The PauliNet architecture emphasizes two-particle interactions and essentially only uses convolutions as input features: $f_i^l = [s_i^{l,\text{el}}, s_i^{l,\text{nuc}}]$. In addition to not using the h_i as input features, PauliNet also limits its effective depth by making the convolutional kernels B functions of the electron-electron distances $|r_{ij}|$ instead of the high-dimensional embedded representations g_{ij} . The FermiNet architecture on the other hand emphasizes the one-electron stream and only uses sums over g_{ij} as input features, essentially corresponding to $B^l = \text{Id}$, $C(\cdot) = 1$. Furthermore FermiNet does not contain an explicit stream for electron-nucleus interactions. Since our architecture adequately models both the one-electron-embedding as well as the two-particle-interactions, we expect our architecture to be more expressive than either predecessor, as demonstrated in Sec. 4.

2.2 Local, invariant input features

The first stage of any VMC wavefunction model is typically the computation of suitable input features from the raw electron coordinates r and nuclear coordinates $\{R_I\}$. While the subsequent embedding stage could in principle take the raw coordinates, appropriate features allow to explicitly enforce symmetries and improve the model's transferability and accuracy. Input features should have three properties: First, they should be sufficiently expressive to encode any physical wavefunction. Second, the features should be invariant under geometric transformations. Third, the features should primarily depend on the local environment of a particle, i.e. similar local geometries should generate similar local features, mostly independent of changes to the geometry far from the particle in question. Published architectures have so far not been able to address all three points: PauliNet [10] uses only distances as input features, making them invariant and local, but not sufficiently expressive, as demonstrated by [12]. FermiNet [10] uses raw distances and differences, making the inputs expressive and local, but not invariant under rotations. PESNet [12] proposes a global coordinate system along the principle axes of a molecule, making the inputs invariant and sufficiently expressive, but not local.

We propose using local coordinate systems centered on every nucleus and evaluating the electron-nuclei differences in these local coordinate systems. Effectively this amounts to applying a rotation

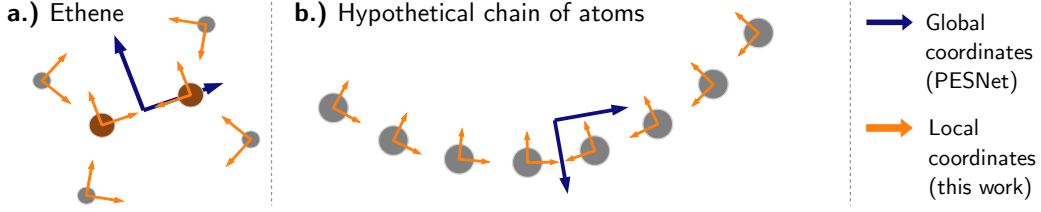


Figure 3: Visualization of resulting coordinate systems for 2 example molecules: a) Ethene b) A hypothetical bent chain of atoms.

matrix U_J to the raw electron-nucleus differences: $\tilde{\rho}_{iJ} = U_J \rho_{iJ}$. As opposed to [12] where U is constant for all nuclei, in our approach U_J can be different for every nucleus. These electron-nucleus differences $\tilde{\rho}_{iJ}$ are invariant under rotations, contain all the information contained in raw Cartesian coordinates and depend primarily on the local environment of an atom. To compute the 3x3 rotation matrices U_J , we first run a single Hartree-Fock calculation using a minimal basis set to obtain a density matrix D . For each atom J we choose the 3x3 block of the density matrix which corresponds to the 3 p-orbitals of the atom J and compute its eigenvectors $U_J = \text{eig}(D_J)$. To make our coordinate system unique we sort the eigenvectors of D_J by their corresponding eigenvalues. If the eigenvalues are degenerate, we pick the rotation of the eigenvectors that maximizes the overlap with the coordinate-axes of the previous nucleus. Fig. 3 shows the resulting coordinate systems spanned by U_J . Note that the local coordinate system generally depicts more physically meaningful directions such as "along the chain". We find that these local coordinates slightly increase the accuracy for single geometries, but more importantly we expect the wavefunction to generalize better across different molecules or geometries. This should improve the accuracy of approaches that attempt to learn wavefunctions for multiple geometries at once [11, 12].

2.3 Initialization of orbital envelope weights

When considering a single atom, the entries of the wavefunction determinant have essentially the form

$$\Lambda_{ki}(\mathbf{r}) \exp(-\omega_k |\rho_{iI}|).$$

In [10], the exponential envelope was purely motivated by the boundary condition that the orbitals must decay to zero, and initialization with $\omega_k = 1$ was proposed. However, when comparing this ansatz to analytical solutions, an interesting parallel can be found: Analytical solutions to the Schrödinger equation for atoms with a single electron – the only systems that have analytical solutions – are of the form

$$\tilde{\Lambda}_k(\rho_{iI}) \exp\left(-\frac{Z}{n_k} |\rho_{iI}|\right),$$

where $\tilde{\Lambda}_k(\rho_{iI})$ is a product of a Laguerre polynomial and a spherical harmonic, and $n_k \in \mathbb{N}^+$ is known as the principal quantum number. This suggests $\omega_k \approx Z/n_k$, which we also find when analyzing the weights of a fully trained wavefunction. When initializing with $\omega_k = Z/n_k$ instead of $\omega_k = 1$, we observe faster convergence, lower final energies, and lower variance of the energies (Sec. 4). The effect is most notable for molecules containing nuclei with large Z , where $Z/n_k \gg 1$.

2.4 Improved hyperparameters

Beyond the improved neural network architecture we fine-tuned the hyperparameters to reduce the number of optimization steps required for convergence. Starting from the hyperparameters proposed by [10], we increased the norm constrain by 3x for the second-order optimizer KFAC [23, 24], decreased learning rate by 0.5x, and decreased learning rate decay time by 0.4x. We observe that these changes stabilize the optimization and enable usage of 50% fewer Monte Carlo walkers, which results in ~ 2 x faster optimization and reduced memory allocation. A complete set of hyperparameters can be found in appendix B.

3 Results of improved approach

We evaluated the accuracy of our approach by comparing our computed energies against the most accurate references available in the literature. Fig. 4 compares our energies against variational methods – for which lower energies are guaranteed to be more accurate – as well as non-variational high-accuracy methods. We find that across many different systems (small and large atoms, molecules at equilibrium geometry, molecules in transition states), our approach yields substantially lower – and thus more accurate – energies than previous variational results. Across all tested systems, we outperform almost all existing variational methods, both deep-learning-based methods as well as classical ones. When comparing to high-accuracy FermiNet VMC calculations, we not only reach substantially lower energies, but also do so using 3-4x fewer training steps, with each step being 40% faster (cf. appendix C). Comparing to a concurrently published Diffusion Monte Carlo approach, which used $\sim 10\times$ more computational resources, we achieve similar or better accuracy for molecules like N_2 and cyclobutadiene and slightly lower accuracy for benzene. Non-variational methods (e.g. CCSD(T)) yield slightly lower energies than our calculations for some molecules, but since those methods do not provide upper bounds or uncertainty guarantees they do not provide a ground-truth. For many applications not only absolute energies are important, but energy differences between

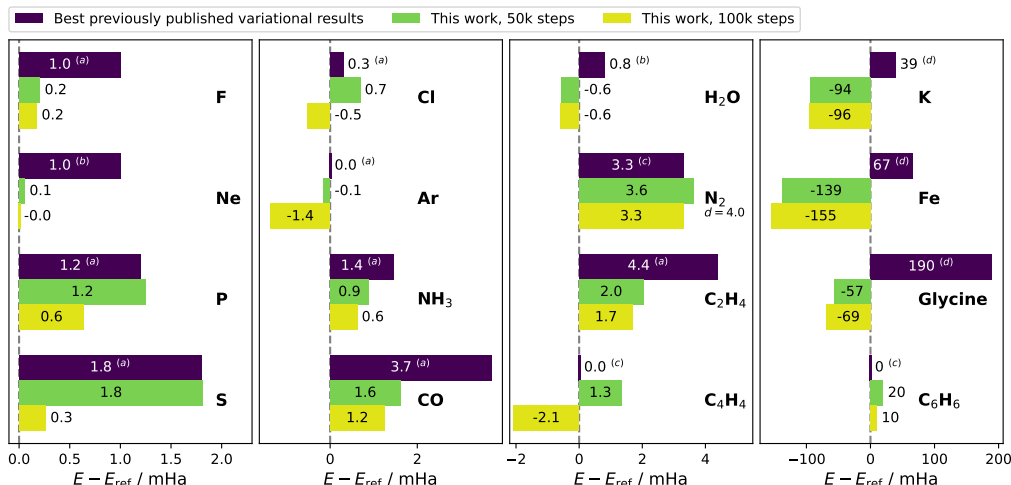


Figure 4: Energies relative to the previously known best estimate, (lower is better). Blue bars depict best published variational energies, footnotes mark the method: a: FermiNet VMC [10, 17], b: Conventional DMC [25, 26, 27], c: FermiNet DMC [21], d: MRCI-F12. A table of absolute energies and methods for E_{ref} can be found in appendix A. Note that E_{ref} is not necessary variational and thus may underestimate the true energy.

different molecules or geometries are of interest, for example to determine the energy required to break a chemical bond. A particularly difficult challenge is the dissociation of the N_2 molecule, i.e. the energy of an N_2 molecule at different bond lengths (inset Fig. 5). Even methods that are generally regarded as highly accurate, such as CCSD(T), predict energies that deviate far from experimental data at bond-lengths from 2.5 - 4 bohr. Fig. 5 depicts this deviation between experimental and computed energy for our method and the best available reference calculations. We find that our results are closer to the experimental absolute energies than all previous work, and are similar to concurrently published FermiNet-DMC results which require 5-10x more computational resources. When comparing relative energies, our approach outperforms all other deep-learning-based methods and CCSD(T), and is only beaten by the highly specialized r12-MR-ACPF method [28]. Similar to absolute energies, we also find that our relative energies converge substantially faster than for other deep-learning-based methods, with relative energies being almost fully converged after 50k epochs.

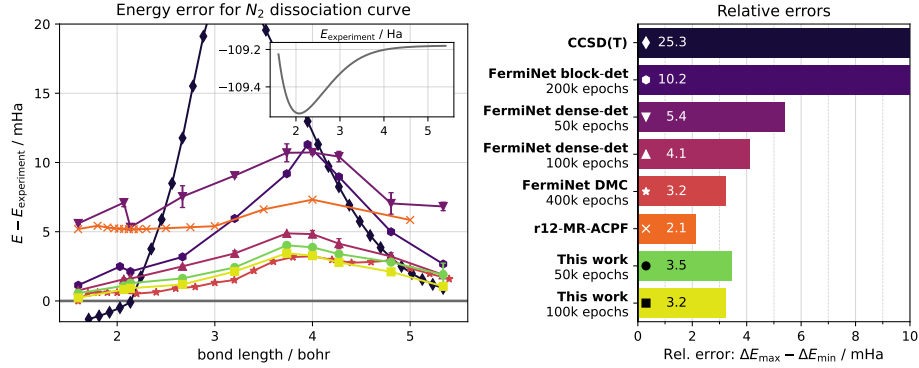


Figure 5: Comparison of energy error $E - E_{\text{experiment}}$ for the dissociation of the N_2 molecule across various methods. Errorbars corresponds to the standard deviation wrt. two different seeds. Errorbars for our work are too small to be visible (~ 0.1 mHa). Results for $E_{\text{experiment}}$ can be found in [29], FermiNet block-det & CCSD(T) in [10], FermiNet DMC in [21] and r12-MR-ACPF in [28].

4 Ablation study

To investigate which specific changes lead to the observed improvements in accuracy, we start from the improved FermiNet architecture proposed in [17] and incrementally add improvements in the following order: First, we use dense $n_{\text{el}} \times n_{\text{el}}$ determinants introduced by the FermiNet authors [10, 17] in their GitHub repository and described in [18] instead of block-diagonal determinants. This generalization increases computational cost and parameter count (cf. appendix C) but has been found to better describe the wavefunction’s nodal surface and thus increase expressiveness of the ansatz. Second, we change hyperparameters as described in Sec. 2.4, which increases throughput by $\sim 2\times$. Third, we augment the electron embedding using our new SchNet-like neural network architecture described in Sec. 2.1. This leads to a moderate increase in parameter count and computational cost. Fourth, we switch to local, invariant input features as described in Sec. 2.2 and remove the electron-electron difference vectors $\mathbf{r}_{ij}$ as inputs. Lastly we switch to initializing $\omega_{kI}^d = Z/n_k$ as described in Sec. 2.3, resulting in our proposed final method. We note that the accuracy gains of these changes are not fully independent of each other and the relative attribution depends on the order in which they are applied: Earlier changes will generally generate larger energy improvements compared to later changes. At each step we compute total energies for three different molecules: ethene, N_2 at the challenging bond-length of 4.0 bohr, and the K-atom. Fig. 6 depicts the energy of our implementation of FermiNet, and the energy change caused by each subsequent improvement. Each experiment was repeated two times with different RNG seeds (appendix D), the errorbars depict the spread in energy. Overall we find that all our changes combined yield a $\sim 3\text{-}20\times$ improvement in the energy error. For ethene, the dominant contribution (3.7 mHa) comes from improved hyperparameters, which lead to the results being mostly converged after 50k epochs vs. the original settings which require 200k epochs for convergence. Using a lower learning rate in combination with a larger gradient-norm-constraint ensures that more optimization steps are taken according to curvature estimated by KFAC and fewer steps are clipped by the gradient-norm-constraint. Architectural improvements (embedding and input features) lower the energy error by additional 1.4 mHa. Because our embedding is a strict generalization of both FermiNet and PauliNet, our ansatz is more expressive and can therefore reach lower energies than previous ansätze. For N_2 it has already been observed that a single dense determinant can outperform models with multiple block-diagonal determinants [18]. We find the corresponding result that 32 dense determinants substantially lower the energy relative to an ansatz with 32 block-diagonal determinants. Comparing N_2 to ethene, we observe larger contributions from our architectural improvements and smaller contributions from improved hyperparameters. For the K atom, the overall gains are largest, totalling 60mHa, with substantial accuracy gains from all improvements. Since K has a much larger nuclear charge ($Z=19$) than the constituents of ethene ($Z=1,6$) and N_2 ($Z=7$), also the physics-inspired initialization of the envelope parameters yields a substantial contribution. This improved initialization leads to a better initial guess for the wavefunction, which not only reduces the number of required optimization steps, but also leads to more accurate initial sampling.

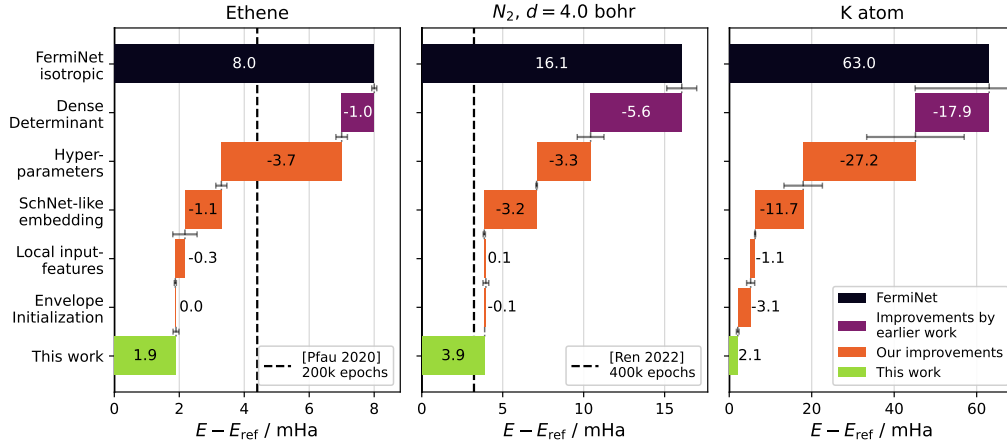


Figure 6: Breakdown of accuracy improvements for three different molecules, each trained for 50k epochs. The black dashed line depicts the best published variational result and as reference we use the best available estimates: CCSD(T) [10] for ethene, experimental data for N_2 [29], and our own VMC calculations after 100k epochs for the K atom.

5 Incorporating prior knowledge

To further understand the effect of incorporating prior knowledge into neural network architectures for physical problems as the electronic Schrödinger equation, we examined two distinct ways of increasing prior information in our model: First, by including a built-in approximate physical model, analogous to PauliNet. Second, by increasing the number of pre-training steps to more closely match a reference wavefunction before starting the optimization.

Explicitly include CASSCF PauliNet maximizes the physical prior information by computing the envelopes Ω with CASSCF, a sophisticated conventional method, and explicitly enforcing the Kato cusp conditions [30]. Starting from our proposed architecture, we modified our approach step-by-step until we arrived at a PauliNet-like architecture. Fig. 7a shows the energies of an NH_3 molecule trained for 50k epochs at each step.

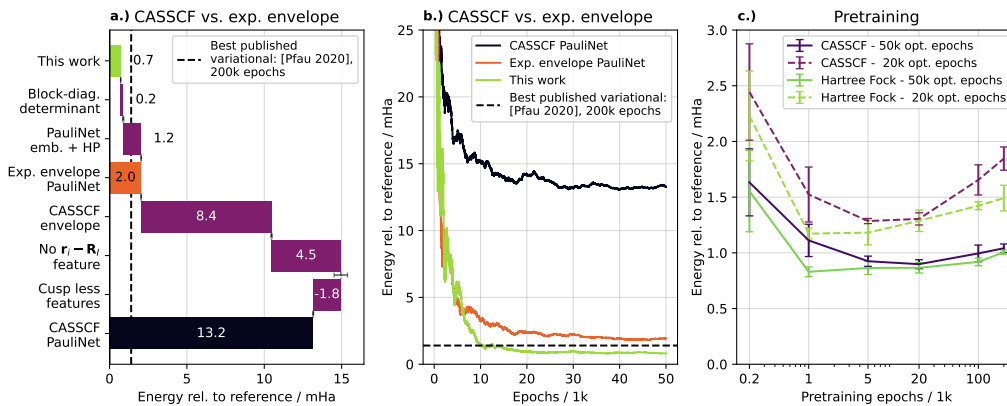


Figure 7: Effect of prior knowledge for NH_3 : a) Change in accuracy after 50k optimization epochs when transitioning from our approach to a PauliNet-like architecture, by increasing prior knowledge. b) Energy error for 3 architectures over number of training epochs. c) Mean and standard deviation of the energy error as a function of pre-training steps. Pre-training used CASSCF or HF as a reference, and the energy was evaluated after 20k or 50k subsequent variational optimization epochs.

First, we switch from dense determinants to block-diagonal determinants as used by the original PauliNet, leading to small loss in accuracy. Second, we exchange our embedding for the PauliNet-like embedding using the hyperparameters proposed in [11], leading to a substantial loss in accuracy, presumably caused by a loss in expressiveness. Next, we replace the simple exponential envelopes by the more physically inspired CASSCF-envelopes, causing a large loss in accuracy. We then remove the vector $\mathbf{r}_i - \mathbf{R}_I$ as input feature (keeping only its absolute value) as done in the original PauliNet architecture [9]. This again deteriorates accuracy, presumably due to enforcing rotational invariance which is too restrictive of a symmetry class as pointed out by [12]. Lastly we replace the electron-electron distances $|\mathbf{r}_{ij}|$ (which are not smooth at $\mathbf{r}_{ij} = \mathbf{0}$ and thus lead to cusps) by smooth, cusplless radial basis functions as input feature and add an explicit term to enforce the electron-electron cusp condition. Since the CASSCF-envelopes are designed to fulfill the electron-ion cusp condition, this change leads to an enforced cusp condition, slightly improving the energy accuracy. The largest loss in accuracy throughout these changes is caused by introducing the CASSCF-envelopes, suggesting that they introduce a strong bias of the wavefunction that cannot be easily overcome during training. Fig. 7b shows that architectures using exponential envelopes converge to lower absolute energies compared to the CASSCF-based PauliNet and outperform PauliNet already after ~ 5000 epochs.

Increase pre-training accuracy Before starting the unsupervised variational optimization of the wavefunction, we run a short supervised pre-training of the wavefunction to roughly match a given reference wavefunction. This is computationally inexpensive because it only requires evaluation of ψ_θ and a back-propagation step to update the neural network weights θ but not the second derivative of the Hamiltonian. If the reference method yields a decent approximation to the true wavefunction, this pre-training significantly speeds-up optimization and avoids unstable parameter regimes [10]. To incorporate more prior knowledge, one could either use a more sophisticated reference method (e.g. CASSCF instead of HF) or increase the number of pre-training steps. In Fig. 7c we pre-trained the wavefunction with a block diagonal determinant for the NH_3 molecule using a CASSCF and Hartree-Fock reference. We increased pre-training iteration steps and evaluated the energy after subsequent 20k and 50k variational optimization epochs, each run was repeated with five different seeds. Increasing the number of pre-training steps initially increases accuracy – since it provides a better starting point for the subsequent variational optimization – but when increasing pre-training beyond 20k steps, accuracy deteriorates for both methods. Surprisingly, we observe a drop in accuracy when using CASSCF as a reference method compared to the simpler Hartree-Fock method. This effect is even more pronounced when increasing the number of pre-training steps. It suggests that excessive pre-training introduces a bias that is hard to overcome during variational optimization, similarly to a built-in reference method.

6 Discussion and Limitations

Discussion We find that our approach yields substantially more accurate absolute energies than all previous work – both classical as well as deep-learning-based – and that we reach these accuracies 4-6x faster than the next best method (FermiNet). Especially for larger systems, such as 4th row atoms or the amino acid glycine, we outperform conventional "gold-standard" methods like MRCI-F12(Q) by ~ 100 mHa. This corroborates the fact that deep-learning-based methods are emerging as a new gold-standard in computational chemistry and showcases the immense potential of machine-learning-based methods in the natural sciences. A concurrent work [21] was able to achieve similar accuracies by applying Diffusion Monte Carlo (DMC) on top of a FermiNet VMC calculation, highlighting the potential of deep-learning Monte Carlo methods. However, [21] required ~ 10 x more computational resources and their VMC results – already by themselves 8x more expensive than our calculations – are consistently inferior to our results. This showcases a promising route towards further improvements by using our substantially cheaper and more accurate VMC results as a starting point for a DMC calculation.

Regarding the question of how much physics to include in the model, we find varying results. For exact physical constraints, such as symmetries or the cusp conditions, inclusion in the model generally appears to be helpful. However for prior knowledge from existing approximate solutions (such as CASSCF) the situation is more subtle. On the one hand, soft physical guidance such as short supervised pre-training or physics-inspired weight initialization accelerates optimization. On the other hand, we show empirically that increasing physical prior knowledge, e.g. by incorporating

CASSCF or extensive supervised pre-training, does not necessarily increase accuracy, but can in fact introduce detrimental biases that are hard to overcome during wavefunction optimization.

Limitations and outlook Despite the proposed improvements and favorable scaling of the method, computation of energies for large molecules still takes days of GPU-time on current hardware. While the same holds true for conventional high-accuracy approaches, substantial speed-ups are still required to make DL-VMC more accessible for practitioners. Additionally, when increasing the nuclear charges, the wavefunction becomes increasingly localised, which leads to a reduction in average Monte Carlo stepsize and potentially correlated samples. We circumvent this effect for 4th row atoms by increasing the number of intermediate Monte Carlo steps, but further research into Monte Carlo sampling methods [31, 32] is required to fully address this issue. Despite our improvements for the accuracy of energy differences between different molecules or geometries, DL-VMC is still outperformed by other, computationally cheaper methods in some cases. Initial research into the regularity of the wavefunction across different molecules [11, 12] provides a promising route to improvements. We note in passing that thanks to the local coordinate input features, our architecture fulfills the required rotational invariance required for these approaches.

7 Code availability

The code alongside a detailed documentation is available as part of the DeepErwin code package on the Python Package Index (PyPI) and github (<https://github.com/mdsunivie/deeperwin>) under the MIT license.

8 Acknowledgements

We gratefully acknowledge financial support from the following grants: Austrian Science Fund FWF Project I 3403 (P.G.), WWTF-ICT19-041 (L.G.). The computational results have been achieved using the Vienna Scientific Cluster (VSC). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript. Additionally, we thank Nicholas Gao for providing his results and data and Rafael Reisenhofer for providing valuable feedback to the manuscript.

References

- [1] Janus J. Eriksen et al. “The Ground State Electronic Energy of Benzene”. In: *The Journal of Physical Chemistry Letters* 11.20 (Oct. 2020). Publisher: American Chemical Society, pp. 8922–8929. DOI: 10.1021/acs.jpclett.0c02621. URL: <https://doi.org/10.1021/acs.jpclett.0c02621> (visited on 04/21/2022).
- [2] Narbe Mardirossian and Martin Head-Gordon. “Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals”. In: *Molecular Physics* 115.19 (2017), pp. 2315–2372. DOI: 10.1080/00268976.2017.1333644. eprint: <https://doi.org/10.1080/00268976.2017.1333644>. URL: <https://doi.org/10.1080/00268976.2017.1333644>.
- [3] Aron J. Cohen, Paula Mori-Sánchez, and Weitao Yang. “Challenges for Density Functional Theory”. In: *Chemical Reviews* 112.1 (Jan. 2012). Publisher: American Chemical Society, pp. 289–320. ISSN: 0009-2665. DOI: 10.1021/cr200107z. URL: <https://doi.org/10.1021/cr200107z>.
- [4] Jiequn Han, Linfeng Zhang, and Weinan E. “Solving many-electron Schrödinger equation using deep neural networks”. In: *J. Comput. Phys.* 399 (Dec. 2019), p. 108929. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.108929. URL: <http://dx.doi.org/10.1016/j.jcp.2019.108929>.
- [5] Giuseppe Carleo and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. In: *Science* 355.6325 (Feb. 2017). Publisher: American Association for the Advancement of Science, pp. 602–606. DOI: 10.1126/science.aag2302. URL: <https://www.science.org/doi/10.1126/science.aag2302> (visited on 05/09/2022).
- [6] Wolfgang Pauli. *Nobel lecture*. Dec. 1946.
- [7] Walter Ritz. “Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik.” In: 1909.135 (1909), pp. 1–61. DOI: doi:10.1515/crll.1909.135.1. URL: <https://doi.org/10.1515/crll.1909.135.1>.
- [8] W. K. Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (Apr. 1970), 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97. eprint: <https://academic.oup.com/biomet/article-pdf/57/1/97/23940249/57-1-97.pdf>. URL: <https://doi.org/10.1093/biomet/57.1.97>.
- [9] Jan Hermann, Zeno Schätzle, and Frank Noé. “Deep-neural-network solution of the electronic Schrödinger equation”. In: *Nat. Chem.* 12.10 (Oct. 2020), 891–897. ISSN: 1755-4349. DOI: 10.1038/s41557-020-0544-y. URL: <https://doi.org/10.1038/s41557-020-0544-y>.
- [10] David Pfau et al. “Ab initio solution of the many-electron Schrödinger equation with deep neural networks”. In: *Phys. Rev. Res.* 2 (3 Sept. 2020), p. 033429. DOI: 10.1103/PhysRevResearch.2.033429. URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429>.
- [11] Michael Scherbela et al. *Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks*. 2021. DOI: 10.48550/ARXIV.2105.08351. URL: <https://arxiv.org/abs/2105.08351>.
- [12] Nicholas Gao and Stephan Günnemann. *Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions*. 2021. DOI: 10.48550/ARXIV.2110.05064. URL: <https://arxiv.org/abs/2110.05064>.
- [13] Mike Entwistle et al. *Electronic excited states in deep variational Monte Carlo*. 2022. DOI: 10.48550/ARXIV.2203.09472. URL: <https://arxiv.org/abs/2203.09472>.
- [14] Max Wilson et al. *Wave function Ansatz (but Periodic) Networks and the Homogeneous Electron Gas*. 2022. DOI: 10.48550/ARXIV.2202.04622. URL: <https://arxiv.org/abs/2202.04622>.
- [15] G. Cassella et al. *Discovering Quantum Phase Transitions with Fermionic Neural Networks*. DOI: 10.48550/ARXIV.2202.05183. URL: <https://arxiv.org/abs/2202.05183>.
- [16] Xiang Li, Zhe Li, and Ji Chen. “Ab initio calculation of real solids via neural network ansatz”. In: *arXiv:2203.15472 [cond-mat, physics:physics]* (Mar. 2022). arXiv: 2203.15472. DOI: 10.48550/ARXIV.2203.15472. URL: <http://arxiv.org/abs/2203.15472> (visited on 04/20/2022).

- [17] James S. Spencer et al. *Better, Faster Fermionic Neural Networks*. 2020. DOI: 10.48550/ARXIV.2011.07125. URL: <https://arxiv.org/abs/2011.07125>.
- [18] Jeffmin Lin, Gil Goldshlager, and Lin Lin. “Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation”. In: *arXiv:2112.03491 [physics]* (Dec. 2021). arXiv: 2112.03491. DOI: 10.48550/ARXIV.2112.03491. URL: <http://arxiv.org/abs/2112.03491> (visited on 03/25/2022).
- [19] Xiang Li et al. “Fermionic neural network with effective core potential”. In: *Physical Review Research* 4.1 (Jan. 2022). Publisher: American Physical Society, p. 013021. DOI: 10.1103/PhysRevResearch.4.013021. URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.4.013021> (visited on 04/26/2022).
- [20] Max Wilson et al. *Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo*. 2021. DOI: 10.48550/ARXIV.2103.12570. arXiv: 2103.12570 [physics.chem-ph].
- [21] Weiluo Ren, Weizhong Fu, and Ji Chen. *Towards the ground state of molecules via diffusion Monte Carlo on neural networks*. 2022. DOI: 10.48550/ARXIV.2204.13903. URL: <https://arxiv.org/abs/2204.13903>.
- [22] Kristof Schütt et al. “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf>.
- [23] James Martens and Roger Grosse. “Optimizing neural networks with kronecker-factored approximate curvature”. In: *International conference on machine learning*. PMLR. 2015, pp. 2408–2417.
- [24] Aleksandar Botev and James Martens. *KFAC-JAX*. Version 0.0.1. 2022. URL: <http://github.com/deepmind/kfac-jax>.
- [25] P. Seth, P. López Ríos, and R. J. Needs. “Quantum Monte Carlo study of the first-row atoms and ions”. In: *The Journal of Chemical Physics* 134.8 (2011), p. 084105. DOI: 10.1063/1.3554625. eprint: <https://doi.org/10.1063/1.3554625>. URL: <https://doi.org/10.1063/1.3554625>.
- [26] Norbert Nemec, Michael D. Towler, and R. J. Needs. “Benchmark all-electron ab initio quantum Monte Carlo calculations for small molecules”. In: *The Journal of Chemical Physics* 132.3 (Jan. 2010). Publisher: American Institute of Physics, p. 034111. ISSN: 0021-9606. DOI: 10.1063/1.3288054. URL: <http://aipscitation.org/doi/full/10.1063/1.3288054> (visited on 04/12/2022).
- [27] Bryan K. Clark et al. “Computing the energy of a water molecule using multideterminants: A simple, efficient algorithm”. In: *The Journal of Chemical Physics* 135.24 (Dec. 2011). Publisher: American Institute of Physics, p. 244105. ISSN: 0021-9606. DOI: 10.1063/1.3665391. URL: <http://aipscitation.org/doi/full/10.1063/1.3665391> (visited on 05/31/2022).
- [28] Robert J. Gdanitz. “Accurately solving the electronic Schrödinger equation of atoms and molecules using explicitly correlated (r12-)MR-CI: the ground state potential energy curve of N₂”. In: *Chemical Physics Letters* 283.5 (1998), pp. 253–261. ISSN: 0009-2614. DOI: [https://doi.org/10.1016/S0009-2614\(97\)01392-4](https://doi.org/10.1016/S0009-2614(97)01392-4). URL: <https://www.sciencedirect.com/science/article/pii/S0009261497013924>.
- [29] Robert J. Le Roy, Yiye Huang, and Calvin Jary. “An accurate analytic potential function for ground-state N₂ from a direct-potential-fit analysis of spectroscopic data”. In: *The Journal of Chemical Physics* 125.16 (2006), p. 164310. DOI: 10.1063/1.2354502. eprint: <https://doi.org/10.1063/1.2354502>. URL: <https://doi.org/10.1063/1.2354502>.
- [30] Tosio Kato. “On the eigenfunctions of many-particle systems in quantum mechanics”. In: *Commun. Pure Appl. Math.* 10.2 (1957), 151–177. DOI: 10.1002/cpa.3160100201. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160100201>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160100201>.
- [31] C. J. Umrigar, M. P. Nightingale, and K. J. Runge. “A diffusion Monte Carlo algorithm with very small time-step errors”. In: *The Journal of Chemical Physics* 99.4 (1993), pp. 2865–2890. DOI: 10.1063/1.465195. eprint: <https://doi.org/10.1063/1.465195>. URL: <https://doi.org/10.1063/1.465195>.

- [32] C. J. Umrigar. “Accelerated Metropolis method”. In: *Phys. Rev. Lett.* 71 (3 July 1993), pp. 408–411. DOI: 10.1103/PhysRevLett.71.408. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.71.408>.
- [33] Subhas J. Chakravorty et al. “Ground-state correlation energies for atomic ions with 3 to 18 electrons”. In: *Phys. Rev. A* 47 (5 May 1993), pp. 3649–3670. DOI: 10.1103/PhysRevA.47.3649. URL: <https://link.aps.org/doi/10.1103/PhysRevA.47.3649>.
- [34] Bruce J. Rosenberg and Isaiah Shavitt. “Ab initio SCF and CI studies on the ground state of the water molecule. I. Comparison of CGTO and STO basis sets near the Hartree–Fock limit”. In: *The Journal of Chemical Physics* 63.5 (Sept. 1975). Publisher: American Institute of Physics, pp. 2162–2174. ISSN: 0021-9606. DOI: 10.1063/1.431596. URL: <http://aipscitation.org/doi/10.1063/1.431596> (visited on 05/31/2022).
- [35] H.-J. Werner et al. “MOLPRO: a general-purpose quantum chemistry program package”. In: *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 2 (2012), 242–253.
- [36] H.-J. Werner et al. *MOLPRO, version 2012.1, a package of ab initio programs*. see <https://www.molpro.net>. Cardiff, UK, 2012.
- [37] Thomas B. Adler, Gerald Knizia, and Hans-Joachim Werner. “A simple and efficient CCSD(T)-F12 approximation”. In: *J. Chem. Phys.* 127.22 (2007), p. 221106. DOI: 10.1063/1.2817618. eprint: <https://doi.org/10.1063/1.2817618>. URL: <https://doi.org/10.1063/1.2817618>.
- [38] Kirk A. Peterson, Thomas B. Adler, and Hans-Joachim Werner. “Systematically convergent basis sets for explicitly correlated wavefunctions: The atoms H, He, B–Ne, and Al–Ar”. In: *J. Chem. Phys.* 128.8 (2008), p. 084102. DOI: 10.1063/1.2831537. eprint: <https://doi.org/10.1063/1.2831537>. URL: <https://doi.org/10.1063/1.2831537>.
- [39] J. Grant Hill, Shivnath Mazumder, and Kirk A. Peterson. “Correlation consistent basis sets for molecular core-valence effects with explicitly correlated wave functions: The atoms B–Ne and Al–Ar”. In: *J. Chem. Phys.* 132.5 (2010), p. 054108. DOI: 10.1063/1.3308483. eprint: <https://doi.org/10.1063/1.3308483>. URL: <https://doi.org/10.1063/1.3308483>.
- [40] Florian Weigend, Filipp Furche, and Reinhart Ahlrichs. “Gaussian basis sets of quadruple zeta valence quality for atoms H–Kr”. In: *J. Chem. Phys.* 119.24 (2003), pp. 12753–12762. DOI: 10.1063/1.1627293. URL: <https://doi.org/10.1063/1.1627293>.

A Absolute energies

Table 1: Ground-state energy in Ha for various methods grouped into best variational energy, lowest overall and this work. The lowest variational energy is highlighted with bold letters and the lowest overall energy by an underscore. As reference we included methods with the best energy approximation found in the literature and if not found elsewhere own computations based on a MRCI method were performed. a: FermiNet VMC [10, 17], b: FermiNet DMC [21], c: Diffusion Monte Carlo [25, 26, 27], d: MRCI-F12, e: CCSD(T)/ CBS [10], f: CAS + experimental corrections [33], g: MRCI-F12(Q), h: experiment [34].

Category	System	Lowest Variational	Lowest All	This work 50k epochs	This work 100k epochs
2nd row atoms	O	-75.0666 ^a	<u>-75.0673</u> ^f	-75.0669	-75.0670
	F	-99.7329 ^a	<u>-99.7339</u> ^f	-99.7337	-99.7337
	Ne	-128.9366 ^c	<u>-128.9376</u> ^f	-128.9375	-128.9376
3rd row atoms	P	-341.2578 ^a	<u>-341.259</u> ^f	-341.2578	-341.2584
	S	-398.1082 ^a	<u>-398.110</u> ^f	-398.1082	-398.1097
	Cl	-460.1477 ^a	<u>-460.148</u> ^f	-460.1473	-460.1485
	Ar	-527.5405 ^a	-527.5405 ^a	-527.5406	-527.5419
4th row atoms	K	-599.7847 ^d	-599.8238 ^g	-599.9178	-599.9195
	Fe	-1263.4281 ^d	-1263.4947 ^g	-1263.633	-1263.650
Small molecules	N ₂ , $d = 2.068$	-109.5416 ^b	<u>-109.5425</u> ^e	-109.5408	-109.5414
	Ethene	-78.5844 ^a	<u>-78.5888</u> ^e	-78.5868	-78.5871
	H ₂ O	-76.4368 ^c	-76.4376 ^h	-76.4382	-76.4382
	CO	-113.3218 ^a	<u>-113.3255</u> ^e	-113.3239	-113.3242
	NH ₃	-56.5630 ^a	<u>-56.5644</u> ^e	-56.5635	-56.5638
Larger molecules	Cyclobutadiene	-154.6770 ^b	-154.6770 ^b	-154.6757	-154.6791
	Benzene	-232.2370 ^b	-232.2370 ^b	-232.2168	-232.2267
	Glycine	-284.1741 ^d	-284.3639 ^g	-284.4201	-284.4328

B Computational settings

Neural network The main hyperparameters used in this work can be found in Tab. 2 with minor changes for larger systems. For the third and fourth row atoms we increased as in [17] the single electron stream to 512. For K and Fe we choose a higher damping for the second-order optimizer KFAC with 4×10^{-3} to stabilize the optimization. Additionally, we observed that with increasing nuclear charge Z the MCMC stepsize tends to decrease because of a high probability peak around the core resulting in two distinct length scales for the core and valence electrons. To circumvent high correlation between samples we increased the number of decorrelation steps to 60 for Fe. We observed for larger systems that 1000 pretraining steps is not sufficiently close to a Hartree-Fock solution and therefore increased for glycine, benzene, cyclobutadiene, K and Fe the number of pretraining steps to 4000. For glycine and benzene the batch size and number of MCMC walkers was decreased to 1700 allowing us to use a single NVIDIA A100 GPU and for glycine, benzene and cyclobutadiene no envelope initialization at the beginning of optimization was performed. All computations were computed on either a single NVIDIA A100 or A40 GPU. We use separate 10k evaluation steps in which we do not update the neural network parameters to compute all final energies, to ensure fully unbiased energies.

Reference calculation with MRCI We carried out reference calculations for several small molecules (see Table 3) and atoms (see Table 4) using the MOLPRO package [35, 36]. On the one hand, as single-reference method, we tested coupled cluster with single and double excitations including perturbative triple excitations in its explicitly correlated F12 formulation (CCSD(T)-F12) [37], which requires a restricted Hartree-Fock (HF) calculation in the employed implementation.

Table 2: Hyperparameter settings used in this work

Orbitals	# determinants	32
	Pretraining basis set	6-311G
	Pretraining steps	1000
Embedding	# determinants n_{det}	32
	# hidden layers $A_{\text{one}}, A_{\sigma_{i,j}}, A_{\text{ion}}, B_{\text{ion}}, B_{\sigma_{i,j}}, C_{\text{ion}}, C_{\sigma_{i,j}}$	0
	# neurons per layer A_{one}	256
	# neurons per layer $A_{\sigma_{i,j}}, A_{\text{ion}}, B_{\text{ion}}, B_{\sigma_{i,j}}, C_{\text{ion}}, C_{\sigma_{i,j}}$	32
	# iterations L	4
	activation function	tanh
Markov Chain Monte Carlo	# walkers	2048
	# decorrelation steps	20
	Target acceptance probability	50%
Optimization	Optimizer	KFAC
	Damping	1×10^{-3}
	Norm constraint	3×10^{-3}
	Batch size	2048
	Initial learning rate lr_0	5×10^{-5}
	Learning rate decay	$\text{lr}(t) = \text{lr}_0(1 + t/6000)^{-1}$
	Optimization steps	50,000 - 100,000

On the other hand, as multi-reference methods, we used multi-reference configuration interaction in its explicitly correlated F12 formulation (MRCI-F12), which necessitates a complete active space self-consistent field (CASSCF) calculation. We also carried out perturbative Davidson corrections on the MRCI-F12 values (abbreviated as MRCI-F12(Q)). Among all of these methods, HF, CASSCF (as used here in its state-specific variant) and MRCI-F12 are variational. In contrast, CCSD(T)-F12 and MRCI-F12(Q) contain perturbative contributions to the ground-state energy and, thus, do not obey the variational principle.

The active spaces for the molecules and atoms is denoted here in the format ^{multiplicity}(number of electrons, number of orbitals) and was chosen as: H_2O ¹(8,6), cyclobutadiene ¹(4,4), benzene ¹(6,6), glycine ¹(6,4), K ²(1,6), Fe ⁵(8,6). For the molecules, the correlation-consistent quadruple-zeta basis set cc-pVQZ-F12 [38] was employed together with the recommended parameter GEM_BETA = 1.5 a_0^{-1} [39] (which lead to convergence problems for the benzene multi-reference calculation). In order to judge the basis set convergence, also calculations with the smaller triple-zeta basis set cc-pVTZ-F12 was carried out together with GEM_BETA = 1.4 a_0^{-1} . These basis sets were not available for the atoms and the def2-QZVPP basis set together with the corresponding density-fitting def2-QZVPP-JKfit basis were employed instead [40].

Table 3: Ground-state energies of selected small molecules computed with various combinations of methods and basis sets, as indicated.

Molecule	Basis set	E_{HF}	E_{CASSCF}	$E_{\text{CCSD(T)-F12}}$	$E_{\text{MRCI-F12}}$	$E_{\text{MRCI-F12(Q)}}$
H₂O	cc-pVQZ-F12	-76.0673	-76.1208	-76.3766	-76.4210	-76.4352
	cc-pVTZ-F12	-76.0669	-76.1189	-76.3752	-76.4183	-76.4324
Cyclobutadiene	cc-pVQZ-F12	-153.7060	-153.7598	-154.4607	-154.5288	-154.6380
	cc-pVTZ-F12	-153.7054	-153.7574	-154.4597	-154.5250	-154.6340
Benzene	cc-pVQZ-F12	-230.7970	—	-231.9115	—	—
	cc-pVTZ-F12	-230.7960	-230.8580	-231.9099	-231.9584	-232.1483
Glycine	cc-pVQZ-F12	-282.9740	-282.9902	-284.1623	-284.1741	-284.3639
	cc-pVTZ-F12	-282.9730	-282.9847	-284.1587	-284.1671	-284.3570

Table 4: Ground-state energies of selected atoms computed with the indicated methods in combination with a def2-QZVPP basis set and the corresponding density-fitting def2-QZVPP-JKfit basis set.

Atom	E _{CASSCF}	E _{MRCI-F12}	E _{MRCI-F12(Q)}
K	-599.1645	-599.7847	-599.8238
Fe	-1262.4459	-1263.4281	-1263.4947

C Computational cost

For a better comparison of the computational cost of our proposed architecture to FermiNet we report in Tab. 5 the runtime per optimization epoch and the number of all trainable parameters for all settings tested in Sec. 4. The majority of the runtime per epoch is due to the second derivative for the kinetic energy of the Hamiltonian. Comparing runtime for the K atom we are $\sim 40\%$ faster per epoch than FermiNet with a block determinant structure and even $\sim 60\%$ faster when using a full determinant. The option of full determinant comes not free of costs and increases substantially the number of total trainable parameters and a slightly more expensive optimization iteration for FermiNet. This effects is compensated by a faster convergence as has been seen in Fig. 6. To enable a full like-for-like comparison we compare all runtimes vs. our own implementation of FermiNet, which achieves the same runtime as the original code. Additionally, we state in Tab. 6 the improvements and speed-ups compared to previously published FermiNet results. Speed-ups are a product of a lower number of training iterations times the faster runtime per optimization epoch (cf. Tab. 5). Energy improvements are obtained by taking the ratio of each methods' energy error with respect to a ground truth solution. In three cases the ground-truth is represented by our computations. We used approximately 50k GPUh in total for this work (10k on A100 GPUs, 40k on A40 GPUs).

Table 5: Comparing the architectures from Fig. 6 with respect to runtime per optimization epoch and number of trainable parameters. All timings were performed using a single NVIDIA A100 GPU.

System	Architecture Setting	Runtime per opt. epoch	Number of parameters
K atom	FermiNet isotropic (our implementation)	5.1 s	2,784,672
	Dense Determinant	6.0 s	3,097,184
	Hyperparameters	3.1 s	3,097,184
	SchNet-like embedding	3.8 s	3,175,492
	Local input features	3.6 s	3,175,108
	Envelope initialization (This work)	3.6 s	3,175,108

Table 6: Energy improvements and speed-ups compared to FermiNet. Energy improvements are computed with respect to the best estimate. For three systems our method improves the best estimate and therefore the best estimate is our calculated energy. Speed ups are computed as factor of faster convergence due to fewer training epochs times a 40% (5.1s/3.6s) faster single epoch (cf. Tab. 5).

System	FermiNet epochs	This work, 50k epochs		This work, 100k epochs	
		accuracy gain	speed-up	accuracy gain	speed-up
F	200,000	61%	6x	63%	3x
Ne	200,000	94%	6x	100%	3x
P	350,000	-4%	10x	46%	5x
S	350,000	0%	10x	85%	5x
Cl	350,000	-48%	10x	100%	5x
Ar	350,000	11%	10x	100%	5x
NH ₃	200,000	38%	6x	57%	3x
CO	200,000	45%	6x	53%	3x
N ₂	200,000	52%	6x	69%	3x
C ₂ H ₄	200,000	53%	6x	61%	3x
C ₄ H ₄	200,000	71%	6x	100%	3x

D Run-to-run variation

Due to the stochastic nature of initialization and Monte Carlo sampling, our results show small run-to-run variations, depending on the initialization of the random number generator. To estimate the extent of the variation we repeated each experiment in the ablation study two times, and depicted mean and standard deviation of these two runs in Fig. 6. To further quantify this uncertainty, we repeated the optimization of an N_2 molecule 10 times. We used two different settings (our proposed approach and FermiNet using dense-determinants, before applying our improvements) and optimized for 50k epochs. We find a standard deviation of 0.2 mHa for our approach and 0.8 mHa for the FermiNet approach. We attribute the lower spread of our approach to the fact that our results are already converged after 50k epochs (with the remaining uncertainty primarily caused by Monte Carlo evaluation uncertainty), while the unconverged FermiNet results depend more strongly on the random initialization.

E Broader impact

Advancements in computational chemistry may prompt new discoveries in chemistry and biology, which could include positive outcomes such as the development of new drugs or materials. Like every computational chemistry method, our work could hypothetically also be misused for the development of chemical weapons or other potential risks to humanity. As of now, this seems highly unlikely due to basic nature of our research.

II Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks

Comments

E-Print: [arXiv:2105.08351](https://arxiv.org/abs/2105.08351)[physics.comp-ph]

Authorship: L.G., M.S. and R.R. share first authorship.

Contribution

P.G., P.M. and R.R. conceived the project. M.S., L.G. and R.R. developed the detailed method. M.S. and L.G. wrote the Python code with contributions from R.R. The numerical experiments were designed and performed by M.S., L.G. and P.M. with support from R.R. R.R., M.S. and L.G. wrote the manuscript with input from P.G. and P.M. P.G. supervised the project. R.R. and P.G. obtained funding.

Bibliographic Information

Scherbela, Michael, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs (2022). “Solving the Electronic Schrödinger Equation for Multiple Nuclear Geometries with Weight-Sharing Deep Neural Networks.” In: *Nature Computational Science* 2.5, pp. 331–341. DOI: [10.1038/s43588-022-00228-x](https://doi.org/10.1038/s43588-022-00228-x).



Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks

Michael Scherbela^{1,5}, Rafael Reisenhofer^{1,2,5}✉, Leon Gerard^{1,5}, Philipp Marquetand^{1,3} and Philipp Grohs^{1,2,4}

The Schrödinger equation describes the quantum-mechanical behaviour of particles, making it the most fundamental equation in chemistry. A solution for a given molecule allows computation of any of its properties. Finding accurate solutions for many different molecules and geometries is thus crucial to the discovery of new materials such as drugs or catalysts. Despite its importance, the Schrödinger equation is notoriously difficult to solve even for single molecules, as established methods scale exponentially with the number of particles. Combining Monte Carlo techniques with unsupervised optimization of neural networks was recently discovered as a promising approach to overcome this curse of dimensionality, but the corresponding methods do not exploit synergies that arise when considering multiple geometries. Here we show that sharing the vast majority of weights across neural network models for different geometries substantially accelerates optimization. Furthermore, weight-sharing yields pretrained models that require only a small number of additional optimization steps to obtain high-accuracy solutions for new geometries.

Using a deep neural network-based ansatz for variational Monte Carlo (VMC) has recently emerged as a novel approach for highly accurate ab initio solutions to the multi-electron Schrödinger equation^{1–5}. It has been observed that such methods can exceed the accuracy of gold-standard quantum-chemistry methods such as coupled clusters with single-, double- and perturbative triplet-excitations (CCSD(T))⁶, with the computational cost per step scaling only with complexity $\mathcal{O}(N^4)$ for the number of electrons N (ref. 4). This suggests a drastic improvement from classical quantum-chemistry methods such as CCSD(T) or configuration interaction singles, doubles, triples, quadruples (CISDTQ), which scale with $\mathcal{O}(N^7)$ and $\mathcal{O}(N^{10})$, respectively. However, due to the large number of free parameters and the need for Monte Carlo integration, the constant runtime prefactor for neural network-based methods is typically much larger than for classical approaches such that even systems of modest size still require days or weeks for computation when using highly optimized implementations on state-of-the-art hardware⁷. This often renders deep neural network (DNN)-based ansatz methods unfeasible in practice, in particular when highly accurate results for a large number of molecular geometries are required.

Among such tasks are computational structure search, determination of chemical transition states and the generation of training datasets for supervised machine learning algorithms in quantum chemistry. The last two methods are applied with great success to interpolate results of established quantum-chemistry methods such as energies and forces^{8–10}, properties of excited states¹¹, underlying objects such as orbital energies¹² or the exchange energy¹³. Given sufficient training data, these interpolations already achieve chemical accuracy relative to the training method (for example, density functional theory)¹⁴, highlighting the need for increasingly

accurate ab initio methods that can be used to generate reference training data.

The goal of making DNN-based VMC applicable for the generation of such high-quality datasets for previously untractable molecules is a key motivation for this work. The apparent success of supervised learning in quantum chemistry suggests a high degree of regularity of the aforementioned properties and the wavefunction itself within the space of molecular geometries.

Here we already aim to exploit potential regularities of the wavefunction within the space of molecular geometries during VMC optimization by applying a simple technique called weight-sharing. Throughout optimizing instances of the same neural network-based wavefunction model for different molecular geometries, we enforce that each instance has the exact same neural network weights for large parts of the model. In particular, this means that on the parts of the model in which weight-sharing is applied, each instance computes precisely the same function. We note that this idea is reminiscent of (and inspired by) the machine learning technique deep transfer learning, where parts of a pretrained model are reused for different similar tasks to breakthrough results, for example, in natural language processing¹⁵ or computer vision¹⁶.

Weight-sharing can be viewed as a regularization technique that requires large parts of the optimized model to work equally well for a potentially wide variety of different nuclear (or even molecular) geometries. Under the assumption that the wavefunctions are sufficiently uniform across geometries, weight-sharing should therefore have a stabilizing effect on the optimization process and yield wavefunctions that also generalize well to new molecular geometries when used as an initial guess before optimization. For a shared weight, each gradient descent update during optimization for a specific geometry is applied to the complete set of considered

¹Research Network Data Science at University of Vienna, Vienna, Austria. ²Faculty of Mathematics, University of Vienna, Vienna, Austria. ³Faculty of Chemistry, University of Vienna, Vienna, Austria. ⁴Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria. ⁵These authors contributed equally: Michael Scherbela, Rafael Reisenhofer, Leon Gerard. ✉e-mail: rafael.reisenhofer@uni-bremen.de

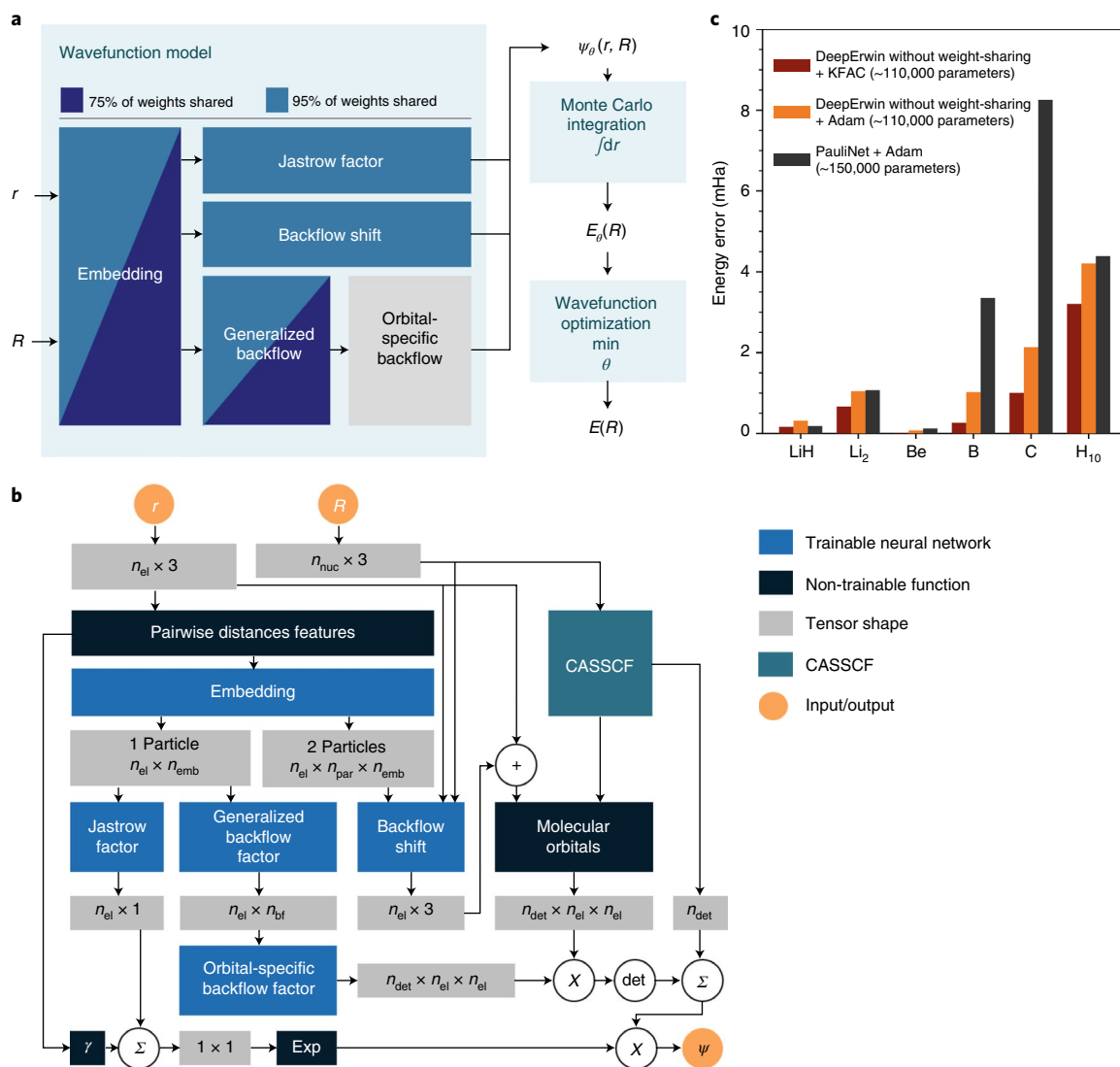


Fig. 1 | Overview of the DeepErwin framework. **a**, Overview of the neural network parts of the wavefunction model implemented in the DeepErwin framework, including a top-level visualization of the two weight-sharing set-ups considered in our experiments. **b**, Overview of the wavefunction model implemented in DeepErwin. Spin dependence has been omitted for clarity, but in practice there are two parallel streams for spin-up and -down, respectively. n_{el} , number of electrons; n_{nuc} , number of nuclei; n_{par} , number of particles (electrons + nuclei); n_{det} , number of determinants; n_{emb} , embedding dimension; n_{bf} , backflow feature dimension. **c**, Energy of optimized wavefunctions for several small atoms and molecules relative to reference calculations⁴ for DeepErwin with first-order optimizer Adam²¹, second-order optimizer K-FAC, and PauliNet². DeepErwin baseline solutions without weight-sharing achieve a higher accuracy than PauliNet across the tested systems despite a smaller number of parameters.

geometries. Weight-sharing therefore has the potential to substantially accelerate the optimization process.

Our main numerical results highlight the benefits of weight-sharing compared with independent optimization and the applicability of pretrained shared weights for new calculations. In particular, we show that by applying these techniques in combination with second-order optimization, it is possible to consistently reach the energies of multi-reference configuration interaction (MRCI)-F12 reference calculations—up to chemical accuracy—for molecules up to the size of ethene after only $\mathcal{O}(10^2)$ optimization epochs per geometry. Note that wavefunctions are typically being optimized from $\mathcal{O}(10^4)$ to $\mathcal{O}(10^5)$ epochs for the most recently proposed DNN-based VMC methods. To further demonstrate the applicability of the proposed framework in practice, we calculate the transition path for H_4^+ between two symmetry-equivalent minima, wavefunctions for a set of differently twisted and stretched ethene configurations, as well as the potential energy surface (PES)—including forces—of a H_{10} chain

on a two-dimensional grid of nuclear coordinates. Our approach yields these forces in addition to energies at a low incremental cost, as opposed to other high-accuracy methods such as domain-based local-pair natural-orbital-coupled cluster¹⁷, for which forces typically have a substantial incremental computational cost.

Results

To investigate weight-sharing for neural network-based models in VMC, we consider a framework called DeepErwin, in which the trial wavefunction is modelled similarly to the recently proposed PauliNet², with modifications leading to an overall smaller network that yields higher accuracies. The basic idea behind this model is to enhance a Slater determinant ansatz with deep neural networks, where initial orbitals are obtained from a complete active space self-consistent field (CASSCF) calculation with a small basis set and a small active space. The resulting wavefunction is then modified by applying a backflow transformation to the orbitals as well

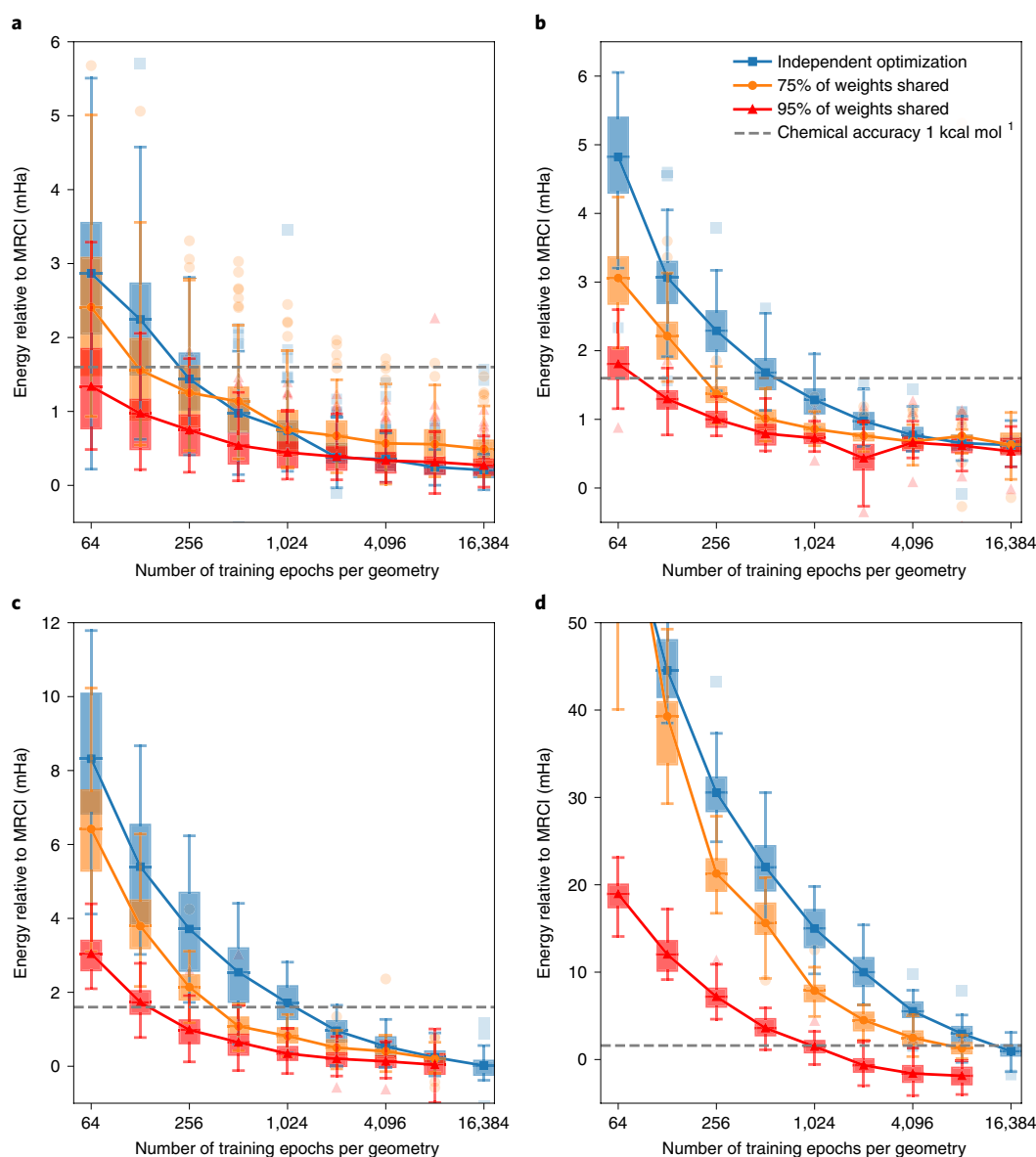


Fig. 2 | Results using weight-sharing during optimization for four different sets of molecules. Each subplot depicts the energy deviation between DeepErwin and a high-accuracy MRCI reference calculation as a function of training epochs. The three coloured lines correspond to three distinct experiments: no weight-sharing (blue), sharing 75% of weights (orange) and sharing 95% of weights (red). Boxplots show the 25–75th percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 interquartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. Dashed horizontal lines (gray) correspond to an error of 1 kcal mol^{−1} (1.6 mHa), commonly referred to as ‘chemical accuracy’. **a**, Weight-sharing for 112 H₄⁺ geometries. **b**, Weight-sharing for 49 H₆ geometries. **c**, Weight-sharing for 49 H₁₀ geometries. Note that the reference energies here differ slightly from the reference energy in Fig. 1c, which was not available for the complete set of 49 geometries. **d**, Weight-sharing for 30 ethene geometries.

as to the electron coordinates and via an additional Jastrow factor. All of these enhancements depend on an embedding of the electron coordinates into a high-dimensional feature space that takes into account interactions with all other particles. This embedding is based on Electronic SchNet¹⁸ and PauliNet. Cusp correction is performed explicitly¹⁹. By contrast to PauliNet, we use an additional equivariant backflow shift for the electron coordinates, smaller embedding networks and no residual in the embedding layers. For a realization of the wavefunction model implemented in the DeepErwin framework, the energy can be approximated through Monte Carlo integration. To eventually obtain the wavefunction of the ground state for a specific molecule, this energy is minimized by applying gradient descent steps to the free parameters of the

wavefunction model. A detailed description of our architecture and the optimization procedure can be found in the Methods.

A top-level overview of the neural network parts of DeepErwin and a detailed visualization of the corresponding wavefunction model are shown in Fig. 1a,b. In comparison with PauliNet, we find that, without using weight-sharing constraints, we achieve a higher accuracy with fewer trainable weights for the small systems tested in ref. ² when training for the same number of epochs (Fig. 1c). This indicates that the architecture implemented in DeepErwin provides a meaningful baseline to investigate the effects of weight-sharing on the optimization process. Application of weight-sharing is of course not limited to this specific model but could equally well be adapted for any neural network-based wavefunction model.

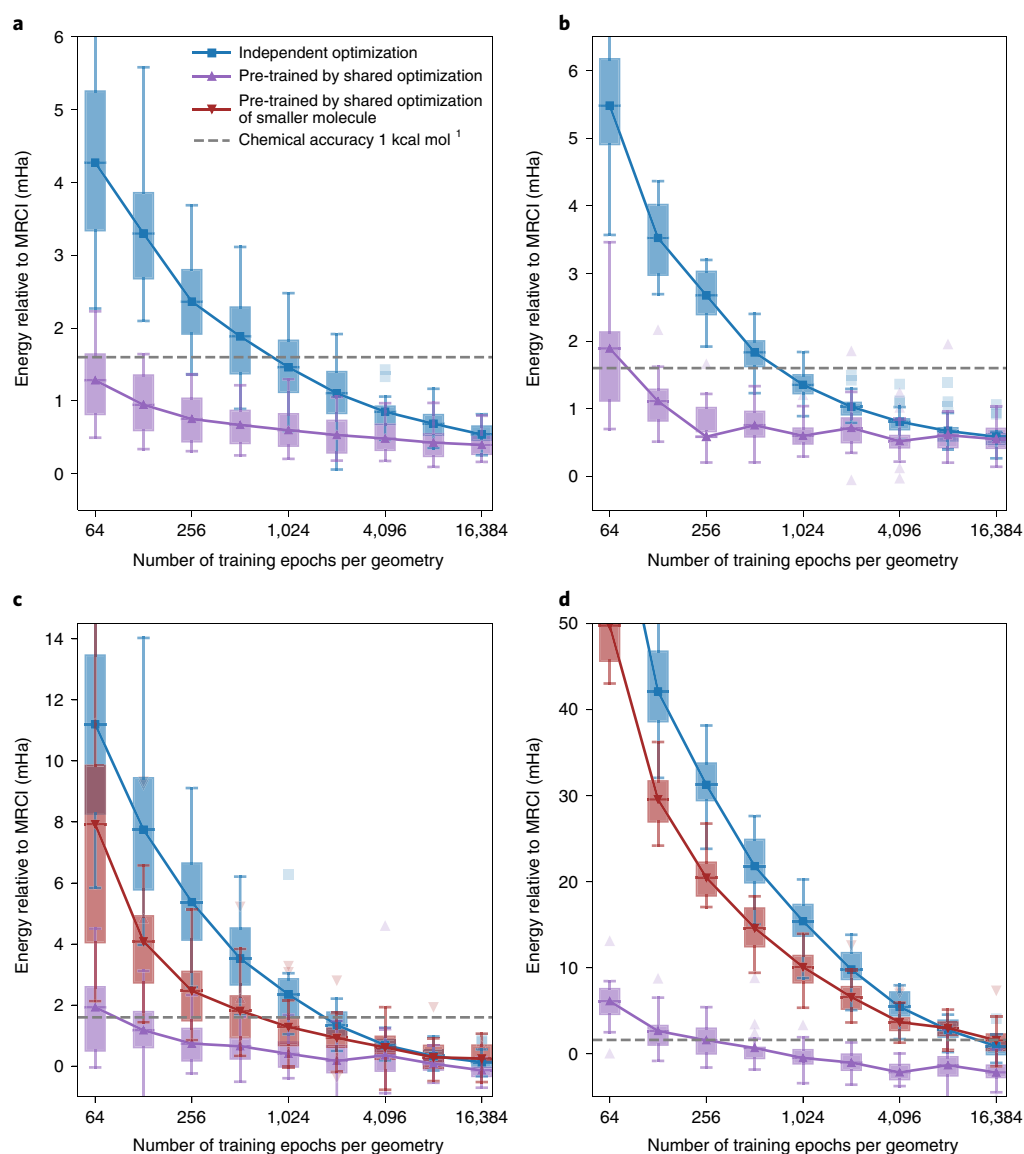


Fig. 3 | Results when reusing pretrained weights for four different sets of molecules. Each panel depicts the energy deviation between DeepErwin and a high-accuracy MRCI reference calculation as a function of training epochs. The three lines correspond to three distinct experiments: independent optimization without reusing weights (blue), reusing weights that were pretrained on different geometries of the same molecule (purple) and reusing weights which were pretrained on a different, smaller molecule (red). Boxplots as in Fig. 2. Dashed horizontal lines (gray) correspond to an error of 1 kcal mol⁻¹ (1.6 mHa), commonly referred to as ‘chemical accuracy’. **a**, Reusing weights for 16 H₄⁺ geometries. **b**, Reusing weights for 23 H₆ geometries. **c**, Reusing weights for 23 H₁₀ geometries. Pretraining on smaller molecules was performed on H₂. Note that the reference energies here differ slightly from the reference energy in Fig. 1c, which was not available for the complete set of considered H₁₀ geometries. **d**, Reusing weights for 20 ethene geometries. Pretraining on smaller molecules was done on methane.

All subsequently reported results were obtained via second-order optimization using a Kronecker-factored approximate curvature (K-FAC)²⁰, which was already implemented for FermiNet⁴. To show that our findings are consistent across different types of optimization, we also report results regarding the accelerated optimization through weight-sharing and the applicability of weight-sharing for pretraining when using the well known Adam algorithm²¹ (Supplementary Figs. 1 and 2).

Accelerated optimization through weight-sharing. The implemented architecture and the hyperparameters used in our experiments are designed to allow for a maximum number of free parameters to be potentially shared across geometries. Two parts of the model that should be particularly well suited for weight-sharing

are the electron coordinate embedding and the generalized part of the backflow factor. Both basically serve as feature extractors that compute high-dimensional embeddings of electron coordinates and are therefore not necessarily required to perform geometry-specific computations. The orbital-specific part of the backflow factor, on the other hand, has a one-to-one correspondence to the orbitals yielded by CASSCF for a given geometry, suggesting that the neural network weights defining it cannot be shared across geometries in a meaningful way.

When weight-sharing is restricted to the embedding and the generalized part of the backflow factor, usually about 75% of the weights in the model are covered by weight-sharing constraints. In the most extreme case, when all weights are being shared—except the ones defining the orbital-specific backflow—this number

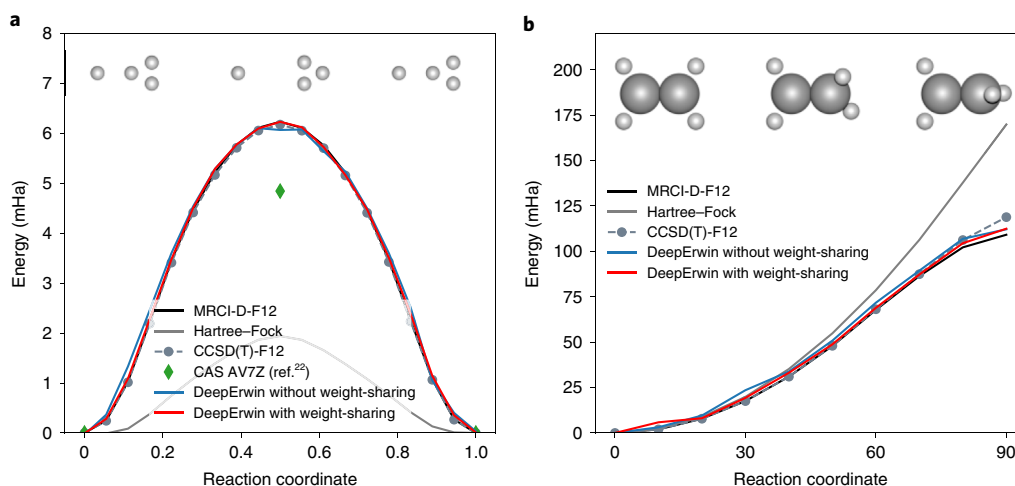


Fig. 4 | Transition barriers for H_4^+ and ethene. Insets depict molecular geometries along the transition path (large grey atoms are carbon and small white atoms are hydrogen), and graphs plot the corresponding energy differences to the ground-state energy along the path. **a**, Energy of H_4^+ along reaction path between symmetry-equivalent minima, via transition state 1 as defined in ref. ²². DeepErwin with weight-sharing constraints is in perfect agreement with MRCI (MRCI-F12(Q)) and coupled clusters (CCSD(T)-F12) after 7,000 optimization epochs per geometry. Hartree-Fock and a complete active space calculation (CAS AV7Z) underestimate the barrier height. **b**, Energies for ten geometries that describe a twist around the carbon-carbon double bond for twisted ethene from 0° to 90° . Results for DeepErwin with and without weight-sharing are plotted after 8,192 optimization epochs per geometry.

grows to roughly 95% (Fig. 1a). Note that precise counts for the numbers of total and shared model parameters used for the experiments in this section slightly differ between different molecules (Supplementary Table 2).

We evaluate both of these set-ups by computing the PES of four different molecules; namely, H_4^+ , the linear hydrogen chains H_6 and H_{10} , and ethene. For H_4^+ , we consider a diverse set of 112 different configurations, covering both low-energy relaxed geometries and strongly distorted configurations. For both hydrogen chains, we calculate the wavefunction of the ground state for 49 different nuclear geometries that lie on a regular grid with respect to a parametrization based on the distance a of two adjacent hydrogen atoms and the distance x between these H_2 pairs. The corresponding PES for H_{10} is visualized in the subsequent section on calculating transition paths and forces. In the case of twisted ethene, the set of 30 geometries iterates over ten different twist angles and three different bond lengths for the C=C bond. The section on calculating transition paths and forces contains a plot of the obtained minimum-energy-path from the non-twisted equilibrium geometry to the 90° rotated molecule, considering for each twist angle the C=C bond length with the lowest energy.

For all four molecules, we compare the optimization of the wavefunction models when applying weight-sharing with the respective independent optimizations. The results of these experiments are compiled in Fig. 2. Across all physical systems, the optimization converges fastest when 95% of the weights are being shared. In particular, we find that in this case, the reference energy (MRCI-F12(Q)/cc-pVQZ-F12; see Methods for computational details) can be reached, up to chemical accuracy, between 6 and 13 times faster than when optimizing the respective geometries independently of each other.

To test our findings in this section against a more difficult benchmark, we also performed an additional experiment in which independent optimizations for different ethene configurations were fully initialized with weights from a wavefunction that had already been optimized for a similar but different molecular configuration using an independent optimization scheme. Although this approach seems to be advantageous during early optimization over a scheme that applies a weight-sharing constraint for 95% of

the weights in the model, at the time that the wavefunctions reach chemical accuracy, shared optimization without pretraining yields an improvement in accuracy that is comparable with the results shown in Fig. 2. Detailed results for this experiment can be found in Supplementary Fig. 3.

Shared optimization as pretraining. Results from the previous section show that even in a setting where ground-state wavefunctions are closely approximated for a wide range of nuclear geometries by different instances of our model, the overwhelming number of free parameters can in fact be identical across those instances. This suggests that parts of our model that were successfully optimized using a weight-sharing constraint encode general computational building blocks that are well suited for the approximation of different wavefunctions. In particular, one could hope that those building blocks also generalize well to molecular geometries for which they were not previously optimized.

To test this hypothesis, we consider for each of the four molecules from the previous experiment a small new set of nuclear geometries that were not part of the original shared optimization. For these sets, we compare two types of independent optimization. In one case, a default random method is used to initialize the weights of the neural networks before optimization. In the second case, we reuse results from the previous optimization in the sense that all weights of the model that were shared in the first experiment are now initialized from the result of this optimization. For the remaining 5% of weights, default random initialization is applied.

Pushing this approach even further, we also use previously optimized shared weights to initialize wavefunction models for an entirely different molecule. In particular, we use shared weights that were optimized for the hydrogen chain H_6 to initialize models for H_{10} , and weights that were optimized for methane to initialize models for ethene. This is possible as the embedding network architecture is independent of the number of particles in the respective molecule. If successful, this method can be used to pretrain weights for large and expensive molecules by solving the Schrödinger equation for smaller, computationally cheaper systems.

The results of these experiments are shown in Fig. 3. For all four considered molecules, we used weights that were optimized with

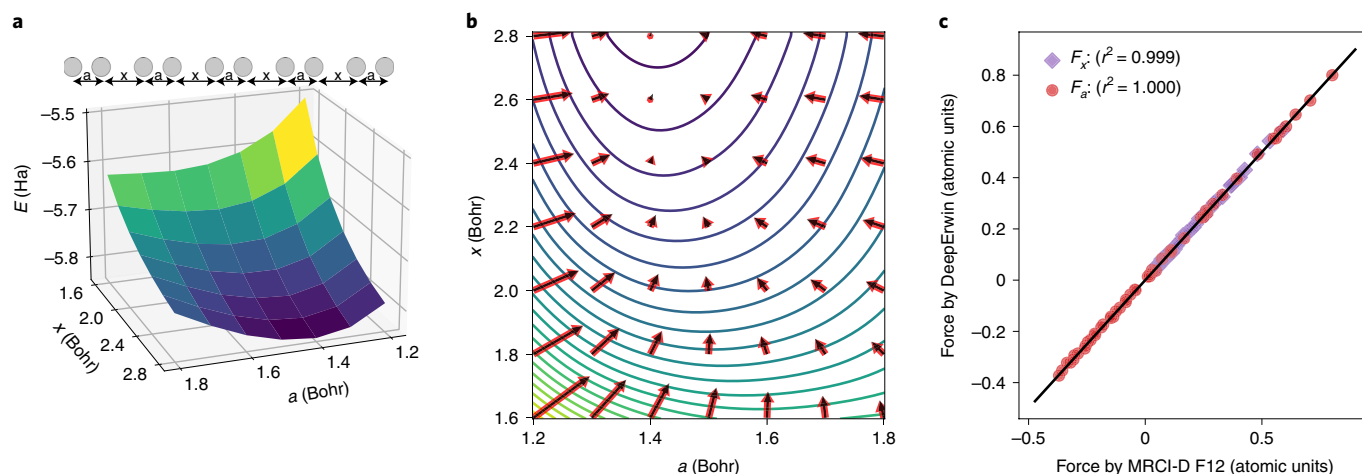


Fig. 5 | Validation of nuclear forces. **a**, Potential energy surface for the H_{10} chain. The variable a represents the distance between two hydrogen atoms and x is the distance between two H_2 molecules. The lowest ground state of the PES describes the dimerization. **b**, Cubic interpolation of the PES with force vectors for the H_{10} chain. Red arrows depict force vectors computed by DeepErwin via the Hellmann–Feynman theorem, whereas black arrows represent numerical gradients that are based on finite differences of MRCI-F12 reference calculations. **c**, Forces computed by DeepErwin plotted against the respective forces obtained from finite differences of the MRCI-F12 reference calculation.

a shared weight constraint on a set of different geometries (Fig. 2) for 8,192 optimization epochs per geometry. Across all systems, pretraining via shared optimization with different geometries of the same molecules dramatically accelerates the subsequent optimization such that the reference energy can be consistently reached up to chemical accuracy after little more than a hundred optimization epochs. In the case of H_{10} , the usage of weights that were pretrained on different configurations of a smaller molecule also yields substantial, albeit much smaller improvements. When using methane configurations to pretrain a wavefunction model for ethene, however, we could only find slight improvements during early optimization.

Calculating transition paths and forces. The substantial speed-ups obtained through weight-sharing enable efficient computational studies for systems that consist of many different geometries of the same molecule. We demonstrate the capabilities of our approach on two exemplary tasks: finding transition paths and calculating potential energy surfaces. As a first example, we calculate the transition path for H_4^+ between two symmetry-equivalent minima via a specific transition state previously proposed in the literature²². For all 19 points along the transition path, wavefunctions are optimized simultaneously for 7,000 epochs per geometry using a weight-sharing constraint that covers about 95% of total weights in the model. We furthermore compute the energies along a reaction path for twisted ethene which describes a rotation of the twist by 90° . The ten geometries considered in this task are a subsample of the 30 geometries previously used in the computations shown in Fig. 2. As a baseline, we consider independent optimization without a weight-sharing constraint as well as classical methods from computational chemistry.

The results of these calculations are shown in Fig. 4a,b. We find our method to be in agreement with high-accuracy reference calculations: DeepErwin with weight-sharing predicts barrier heights that agree with MRCI within $1 \mu\text{Ha}$ (0.02%) for H_4^+ , and 3.3 mHa (3%) for ethene. This leads us to believe that the barrier height for H_4^+ has been underestimated by approximately 1 mHa in previous high-accuracy calculations²². For the electronically challenging case of twisted ethene, both Hartree–Fock as well as CCSD(T)-F12 overestimate the energy of the 90° twisted molecule. DeepErwin, however, yields barrier energies that are much closer to the MRCI-D-F12 reference calculations.

For calculating the transition paths, we used predefined sets of geometries as a given input. In many cases, however, it is not a priori clear which nuclear geometries are of interest in a given task, and a careful exploration of the respective PES is required. Not only are energies required to do this efficiently, but also forces on the nuclei (that is, gradients of the energy with respect to the nuclear coordinates). For realizations of the DeepErwin wavefunction model, these forces can be calculated in a straightforward and computationally efficient fashion via the Hellman–Feynman theorem²³ and by applying established variance correction schemes to accelerate convergence of the Monte Carlo integration (Methods). The PES and the corresponding forces for the linear hydrogen chain H_{10} , evaluated on a regular grid of 49 geometries, are depicted in Fig. 5a,b. The corresponding wavefunctions were optimized for 8,192 epochs per geometry using a weight-sharing constraint for approximately 95% of the model weights. We find that the energetic minimum is given by the dimerization into five H_2 molecules with a covalent bond of $a = 1.4$ Bohr each, rather than for an equally spaced arrangement of atoms. This is an instance of the well known Peierls distortion²⁴.

Figure 5c shows that the force vectors obtained by DeepErwin via the Hellman–Feynman theorem are in agreement with the forces computed from finite differences of MRCI-F12 reference calculations. Our computational experiments do not show any signs of spurious Pulay forces²⁵, which occur when the approximated wavefunction is not an eigenfunction of the Hamiltonian. This suggests that DeepErwin yields not only accurate energies, but also accurate wavefunctions.

Discussion

We found that optimized shared weights yield highly applicable initial weights when considering nuclear geometries for which the wavefunction model was not previously optimized. Even for molecules such as ethene, pretraining with shared optimization makes it possible to reach an MRCI-F12(Q)/cc-pVQZ-F12 reference calculation up to chemical accuracy after only a few hundred optimization epochs. A possibly attractive route towards making VMC optimization tractable for more complex molecules could be to pretrain large parts of a wavefunction model on small, computationally cheap systems. In our experiments we found that the

optimization of H_{10} wavefunctions can in fact be improved substantially when using shared optimization for a set of H_6 geometries to pretrain the respective models. In the case of ethene, however, we could only see small improvements during early optimization when parts of the respective model were pretrained on sets of smaller methane geometries. Due to the fact that any optimization of DNN-based wavefunction models is highly sensitive to changes in the architecture and optimization hyperparameters, we would consider our findings as preliminary evidence that merits further research towards the development of a kind of universal wavefunction, whose neural network parts were optimized for a great number of diverse molecular geometries and which is therefore capable of closely approximating wavefunctions of the ground state for many physical systems after only a brief step of additional optimization.

Our findings suggest a strong regularity of wavefunctions within the space of nuclear geometries. Further evidence for this is also provided by a concurrent method named PESNet, which was released shortly after DeepErwin was first available as a preprint²⁶. PESNet is based on the Ferminet model and employs a meta graph neural network (GNN) to simultaneously learn wavefunctions for a complete PES such that after optimization, the model parameters for a new geometry can be predicted via a simple forward pass through the meta GNN. The fact that this approach reliably yields high-accuracy results for different configurations that were sampled from a PES further underlines our observation that the regularity of wavefunctions within the space of geometries can be heavily exploited for DNN-based quantum Monte Carlo methods by also enforcing regularity on large domains of the parameter space.

Our results do not provide a conclusive answer to whether—for the investigated sets of molecular geometries—the considered weight-sharing set-ups actually limit the capability of our model to approximate the true wavefunctions of the ground state. In general, we would expect this to be the case, but judging from our experimental results, the loss of expressiveness introduced by weight-sharing regularization might often be negligible. This is evidenced by the fact that across all experiments, sharing 95% of model weights yields the same or even lower energies than the respective independent optimizations. Furthermore, it is not clear yet on what an optimal algorithm that exploits weight-sharing for a given task could look like. Based on our results so far, for an exhaustive study of the PES of a molecule, we would suggest a procedure where—possibly guided by estimates of the forces on the nuclei—geometries of interest are iteratively included in a shared optimization, which is eventually concluded by an additional step of independent optimization for some or all of the considered geometries.

The proposed method of weight-sharing is not limited to the specific architecture used in this work but could potentially be exploited for any neural network-based wavefunction model. Due to the interesting regularization properties, it could even be beneficial to apply weight-sharing in a context where only the wavefunction for a single molecular geometry is of interest.

A potential drawback of the proposed method in practice is that shared optimization cannot easily be parallelized across multiple devices (GPUs or CPUs), because each geometry is dependent on updates from all other geometries. One possibility to overcome this issue would be to consider an average loss across all geometries during gradient descent. Such a loss could easily be parallelized by using a separate device for each geometry. In our current implementation of the DeepErwin framework, however, for each epoch only a single geometry is considered during shared optimization, and it is therefore only possible to distribute the Monte Carlo samples within a batch across multiple devices, as it is common practice⁷.

Methods

For a molecule with n_{nuc} nuclei, $n_{\uparrow}$ spin-up electrons and $n_{\downarrow}$ spin-down electrons, we write $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_{\uparrow}}, \dots, \mathbf{r}_{n_{\uparrow}+n_{\downarrow}})$ to denote the set of cartesian electron coordinates, and $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_{n_{\text{nuc}}})$ for the set of coordinates of nuclei. The electron coordinates $\mathbf{r}$ are always assumed to be ordered such that the first $n_{\uparrow}$ entries correspond to spin-up electrons, whereas the last $n_{\downarrow}$ entries are coordinates of spin-down electrons. We write $n_d = n_{\uparrow} + n_{\downarrow}$ for the total number of electrons and Z_i for the charge of the i th nucleus.

Wavefunction model. The model implemented in DeepErwin is closely related to the recently proposed PauliNet⁷. Let θ denote the set of all free (trainable) parameters in the model and n_{det} the number of enhanced Slater determinants. With a high-dimensional embedding of electron coordinates $\mathbf{x}(\mathbf{r}; \mathbf{R}) = (\mathbf{x}_1, \dots, \mathbf{x}_{n_{\uparrow}}, \dots, \mathbf{x}_{n_{\uparrow}+n_{\downarrow}})$ and an explicit term $\gamma(\mathbf{r})$ for cusp correction in the Jastrow factor, a realization ψ_{θ} of the DeepErwin wavefunction model can be written as

$$\psi_{\theta}(\mathbf{r}) = e^{J(\mathbf{x}(\mathbf{r}; \mathbf{R})) + \gamma(\mathbf{r})} \sum_{d=1}^{n_{\text{det}}} \alpha_d \det[\Phi_d^{\uparrow}(\mathbf{r}, \mathbf{x}(\mathbf{r}; \mathbf{R}))] \det[\Phi_d^{\downarrow}(\mathbf{r}, \mathbf{x}(\mathbf{r}; \mathbf{R}))], \quad (1)$$

where $\alpha_d \in \mathbb{R}$ is a trainable weight, the scalar function J defining the Jastrow factor is represented by two fully connected feedforward neural networks, and Slater determinants for spin-up electrons are defined via

$$\Phi_d^{\uparrow}(\mathbf{r}, \mathbf{x}(\mathbf{r}; \mathbf{R})) = \begin{bmatrix} \varphi_1^{\uparrow,d}(\mathbf{r}_1 + \mathbf{s}_1(\mathbf{r}, \mathbf{x}; \mathbf{R})) \eta_1^{\uparrow,d}(\mathbf{x}_1) & \dots & \varphi_{n_{\uparrow}}^{\uparrow,d}(\mathbf{r}_{n_{\uparrow}} + \mathbf{s}_{n_{\uparrow}}(\mathbf{r}, \mathbf{x}; \mathbf{R})) \eta_{n_{\uparrow}}^{\uparrow,d}(\mathbf{x}_{n_{\uparrow}}) \\ \vdots & \ddots & \vdots \\ \varphi_{n_{\uparrow}}^{\uparrow,d}(\mathbf{r}_1 + \mathbf{s}_1(\mathbf{r}, \mathbf{x}; \mathbf{R})) \eta_{n_{\uparrow}}^{\uparrow,d}(\mathbf{x}_1) & \dots & \varphi_{n_{\uparrow}}^{\uparrow,d}(\mathbf{r}_{n_{\uparrow}} + \mathbf{s}_{n_{\uparrow}}(\mathbf{r}, \mathbf{x}; \mathbf{R})) \eta_{n_{\uparrow}}^{\uparrow,d}(\mathbf{x}_{n_{\uparrow}}) \end{bmatrix}. \quad (2)$$

Matrices $\Phi_d^{\downarrow}(\mathbf{r}, \mathbf{x}(\mathbf{r}; \mathbf{R}))$ for spin-down electrons can be defined analogously. The single-electron orbitals $\varphi_i^{\uparrow,d}$ are obtained from the n_{det} most significant Slater determinants from a CASSCF method and remain fixed throughout optimization. The backflow shifts $\mathbf{s}_i$ as well as the backflow factors $\eta_i^{\uparrow,d}$ are represented by fully connected feedforward neural networks. The embedded coordinate $\mathbf{x}_i$ of the i th electron takes into account all particle positions in the system independent of particle type and spin. Further details on our implementation of the coordinate embedding, the Jastrow factor, backflow transformation and cusp correction are given below.

Electron coordinate embedding. The embedding $\mathbf{x}(\mathbf{r}; \mathbf{R})$ is a slightly simplified version of the SchNet embedding used for PauliNet^{2,18}. For brevity, we extend our notation to also include the nuclear coordinates in the coordinates $\mathbf{r}_j$ by defining $\mathbf{r}_j = \mathbf{R}_{j-n_d}$ for $n_d < j \leq n_{\text{nuc}}$. To embed the coordinates $\mathbf{r}_i$ of the i th electron, we consider input features based on pairwise differences and distances with respect to all other particles in the system. Let $i \in \{1, \dots, n_d\}$, $j \in \{1, \dots, n_d + n_{\text{nuc}}\}$ and n_{rbf} denote the number of radial basis features. We use $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $r_{ij} = |\mathbf{r}_{ij}|$ to denote pairwise differences and distances, respectively, and define the pairwise feature vector

$$\mathbf{h}_{ij} = \left(e^{-(r_{ij}-\mu_1)^2/\sigma_1^2}, \dots, e^{-(r_{ij}-\mu_{n_{\text{rbf}}})^2/\sigma_{n_{\text{rbf}}}^2}, \frac{1}{r_{ij} + 0.01} \right) \in \mathbb{R}^{n_{\text{rbf}}+1+3}, \quad (3)$$

where the mean and variance parameters are defined as

$$\mu_k = cq_k^2 \text{ and } \sigma_k = \frac{1}{7}(1 + cq_k), \quad (4)$$

respectively, for an index $k \in \{1, \dots, n_{\text{rbf}}\}$, a parameter q_k that is chosen from an equidistant grid of the interval $[0, 1]$, and a cutoff parameter $c \in \mathbb{R}$.

For a fixed integer L and an embedding dimension n_{emb} , let $(\mathbf{g}^l, \mathbf{w}_{\text{same}}^l, \mathbf{w}_{\text{op}}^l, \mathbf{w}_{\text{nuc}}^l)_{l=0}^L$ and $(\mathbf{f}_{\text{same}}^l, \mathbf{f}_{\text{op}}^l)_{l=1}^L$ denote sequences of vector-valued functions, where each function is represented by a fully connected feedforward neural network with output dimension n_{emb} . To embed the coordinates of the i th electron based on the pairwise feature vectors $\mathbf{h}_{ij}$, we use $\odot$ to denote element-wise multiplication and define for $i \in \{1, \dots, n_d\}$

$$\mathbf{x}_i^0 = \mathbf{g}^0 \left(\sum_{j=1}^{n_d} \mathbf{w}_{\sigma_{ij}}^0(\mathbf{h}_{ij}) \odot \mathbf{f}_{\sigma_{ij}}^0 + \sum_{j=n_d+1}^{n_d+n_{\text{nuc}}} \mathbf{w}_{\text{nuc}}^0(\mathbf{h}_{ij}) \odot \mathbf{f}_{Z_j-n_d}^0 \right), \quad (5)$$

$$\mathbf{x}_i^l = \mathbf{g}^l \left(\sum_{j=1}^{n_d} \mathbf{w}_{\sigma_{ij}}^l(\mathbf{h}_{ij}) \odot \mathbf{f}_{\sigma_{ij}}^l(\mathbf{x}_j^{l-1}) + \sum_{j=n_d+1}^{n_d+n_{\text{nuc}}} \mathbf{w}_{\text{nuc}}^l(\mathbf{h}_{ij}) \odot \mathbf{f}_{Z_j-n_d}^l \right) \text{ with } 1 \leq l \leq L, \quad (6)$$

where σ_{ij} = same for same-spin pairs of electrons and σ_{ij} = op for pairs of electrons with opposite spin; $\mathbf{f}_{\text{same}}^0$, $\mathbf{f}_{\text{op}}^0$ and $\mathbf{f}_{Z_1-n_d}^0, \dots, \mathbf{f}_{Z_j-n_d}^0$ denote trainable vectors of length n_{emb} . The embedding of the i th electron coordinates $\mathbf{r}_i$ is eventually defined as

$$\mathbf{x}_i = \mathbf{x}_i^L. \quad (7)$$

The embedding originally applied in PauliNet has an additional residual term in equation (6) and considers specific functions $(\mathbf{g}^l)_{l=0}^L$ for same-spin, opposite-spin and nuclear input channels.

Backflow transformation. Based on the embedding $\mathbf{x}(\mathbf{r}; \mathbf{R})$, our model applies spin-dependent backflow shifts and factors to the single-electron orbitals in the Slater determinants (equation (2)). For simplicity, this section only considers the spin-up case.

Let $\eta_{\text{gen}}^\uparrow$ denote a vector-valued function that is represented by a fully connected feedforward neural network, and $\omega_{\text{bf}} \in \mathbb{R}$ a single spin-independent trainable weight. For the i th orbital and the j th electron in the d th Slater determinant, the backflow factor is computed as

$$\eta_i^{\uparrow,d}(\mathbf{x}_i) = 1 + \omega_{\text{bf}} \widehat{\eta}_i^{\uparrow,d}(\eta_{\text{gen}}^\uparrow(\mathbf{x}_i)), \quad (8)$$

where $\widehat{\eta}_i^{\uparrow,d}$ denotes an orbital-specific function that is also represented by a feedforward neural network.

The inner function $\eta_{\text{gen}}^\uparrow$ can be seen as an extension of the embedding layer. Apart from electron spin orientation, it remains unchanged across electrons and determinants and is hence called a general backflow factor. The outer function $\widehat{\eta}_i^{\uparrow,d}$, on the other hand, is specific to the i th orbital in the d th determinant obtained from a CASSCF method. The motivation behind defining the backflow factor via the composition of these two functions was to maximize the number of neural networks weights than can possibly be shared across nuclear geometries. Note that for two distinct nuclear geometries, CASSCF does in general not yield the same number of unique orbitals. Although this does not raise an issue for the general backflow factor $\eta_{\text{gen}}^\uparrow$, it implies that sharing the neural network weights that define $\widehat{\eta}_i^{\uparrow,d}$ across nuclear geometries is in general not possible in a meaningful way.

Based on ideas from ref. 27, the backflow shift is based on rotation-invariant features and rotation-equivariant pairwise differences. It is furthermore split into an electron–electron and an electron–nucleus part. Aside from the embedding, we also use side products of the embedding network as inputs and define the following feature vectors

$$\mathbf{f}_i^{\text{el}} = (\mathbf{w}_{\sigma_{i1}}^L(\mathbf{h}_{i1}) \odot \mathbf{f}_{\sigma_{i1}}^L(\mathbf{x}_1^{L-1}), \dots, \mathbf{w}_{\sigma_{in_d}}^L(\mathbf{h}_{in_d}) \odot \mathbf{f}_{\sigma_{in_d}}^L(\mathbf{x}_{n_d}^{L-1})) \quad (9)$$

and

$$\mathbf{f}_i^{\text{nuc}} = (\mathbf{w}_{\text{nuc}}^L(\mathbf{h}_{i(n_d+1)}) \odot \mathbf{f}_{Z_1}^L, \dots, \mathbf{w}_{\text{nuc}}^L(\mathbf{h}_{i(n_d+n_{\text{nuc}})}) \odot \mathbf{f}_{Z_{n_d+n_{\text{nuc}}}}^L) \quad (10)$$

for $i \in \{1, \dots, n_d\}$. All features are obtained without further cost at the end of the embedding loop (equation (6)). The electronic part of the shift for the i th electron is defined as

$$\mathbf{s}_i^{\text{el}}(\mathbf{r}, \mathbf{x}_i, \mathbf{f}_i^{\text{el}}) = \sum_{k=1}^{n_d} \widehat{s}_{\text{el}}(\mathbf{x}_i, \mathbf{f}_{ik}^{\text{el}}) \frac{\mathbf{r}_{ik}}{1 + r_{ik}^3}, \quad (11)$$

$$k \neq i$$

In contrast to the backflow factor, we do not differentiate between different spins to reduce complexity. Similarly, the electron–nuclear shift is computed via

$$\mathbf{s}_i^{\text{nuc}}(\mathbf{r}, \mathbf{R}, \mathbf{x}_i, \mathbf{f}_i^{\text{nuc}}) = \sum_{k=n_d+1}^{n_d+n_{\text{nuc}}} \widehat{s}_{\text{nuc}}(\mathbf{x}_i, \mathbf{f}_{ik}^{\text{nuc}}) \frac{\mathbf{r}_{ik}}{1 + r_{ik}^3}, \quad (12)$$

where $\widehat{s}_{\text{el}}$ and $\widehat{s}_{\text{nuc}}$ are represented by feedforward neural networks. A decay in the vicinity of nuclei ensures that the applied backflow shifts do not lead to a violation of the Kato cusp condition²⁸. The complete backflow shift is computed by

$$\mathbf{s}_i(\mathbf{r}, \mathbf{R}, \mathbf{x}_i, \mathbf{f}_i^{\text{el}}, \mathbf{f}_i^{\text{nuc}}) = \omega_{\text{sh}} \prod_n \tanh(2Z_n |\mathbf{r}_i - \mathbf{R}_n|)^2 \left(\mathbf{s}_i^{\text{el}}(\mathbf{x}_i, \mathbf{f}_i^{\text{el}}) + \mathbf{s}_i^{\text{nuc}}(\mathbf{x}_i, \mathbf{f}_i^{\text{nuc}}) \right), \quad (13)$$

with a spin-independent trainable weight $\omega_{\text{sh}} \in \mathbb{R}$.

Jastrow factor. With two spin-dependent scalar functions $J_\uparrow$ and $J_\downarrow$ that are represented by fully connected feedforward neural networks, the Jastrow factor (equation (1)) is defined as

$$J(\mathbf{x}(\mathbf{r}; \mathbf{R})) = \sum_{i=1}^{n_\uparrow} J_\uparrow(\mathbf{x}_i) + \sum_{i=n_\uparrow+1}^{n_d} J_\downarrow(\mathbf{x}_i). \quad (14)$$

Cusp correction. The Kato cusp condition²⁸ is a necessary condition for eigenstates of the Hamiltonian H . It ensures that the local energies of a wavefunction ψ are finite by forcing the kinetic energy term $\nabla^2 \psi$ to diverge in such a way that it exactly cancels the divergence caused by the potential energy term $Z_n/|\mathbf{r}_i - \mathbf{R}_n|$ when the i th electron approaches the n th nucleus. The orbitals $\varphi_i^{\uparrow,d}(\varphi_i^{\downarrow,d})$ yielded by a CASSCF method (equation (2)) in general do not lead to enhanced Slater determinants that satisfy the cusp condition. We follow the approach outlined in ref. 19 to address this issue: within a radius R_{cusp} around the nuclei, we replace the molecular orbitals by an exponentially decaying function that satisfies the Kato cusp condition, transitions smoothly into the orbitals at R_{cusp} and minimizes the variance of the local energy of this orbital. Cusp correction for the enhanced Slater determinants is performed after the initial set of orbitals has been obtained from a CASSCF calculation and remains fixed during the optimization of the wavefunction parameters θ .

To account for electron–electron cusps in the Jastrow factor (equation (1)), we use an explicit term

$$\gamma(\mathbf{r}) = \sum_{i=1}^{n_d} \sum_{j=i+1}^{n_d} \frac{r_{ij}}{r_{ij} + 1}, \quad (15)$$

similar to the one applied in ref. 1.

Variational Monte Carlo. We use a standard variational Monte Carlo approach to optimize our wavefunction ansatz. Let

$$H = E_{\text{kin}} + E_{\text{pot}} \quad (16)$$

denote the electronic Hamiltonian as obtained within the Born–Oppenheimer approximation, with

$$E_{\text{kin}} = -\frac{1}{2} \sum_i \nabla_{\mathbf{r}_i}^2, \quad (17)$$

$$E_{\text{pot}} = \sum_{i>j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{n>m} \frac{Z_n Z_m}{|\mathbf{R}_n - \mathbf{R}_m|} - \sum_{i,n} \frac{Z_n}{|\mathbf{r}_i - \mathbf{R}_n|}, \quad (18)$$

where E_{kin} accounts for the kinetic energy of the electrons and E_{pot} accounts for the attraction and repulsion between particles in the system. By the Rayleigh–Ritz variational principle, it holds for any wavefunction ψ_θ that its energy E_θ is greater than or equal to the ground-state energy E_0 of the eigenfunction of H associated to the smallest eigenvalue, that is,

$$E_\theta = \int \frac{\psi_\theta(\mathbf{r}) H \psi_\theta(\mathbf{r})}{\Omega_\theta} d\mathbf{r} \geq E_0, \quad (19)$$

where $\Omega_\theta = \int \psi_\theta(\mathbf{r})^2 d\mathbf{r}$ denotes a normalization factor.

To evaluate the energy E_θ for a given set of parameters θ , we use Markov chain Monte Carlo integration (MCMC) and sample electron coordinates according to the probability density

$$p_\theta(\mathbf{r}) = \frac{\psi_\theta(\mathbf{r})^2}{\Omega_\theta}. \quad (20)$$

This allows us to express the total energy E_θ as the expected value of a local energy

$$E_{\text{loc}}(\mathbf{r}) = \frac{H \psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})}, \quad (21)$$

in the sense that

$$E_\theta = \int E_{\text{loc}}(\mathbf{r}) p_\theta(\mathbf{r}) d\mathbf{r} = \langle E_{\text{loc}} \rangle \approx \frac{1}{N} \sum_{k=1}^N E_{\text{loc}}(\mathbf{r}^k), \quad (22)$$

where N denotes the number of sampled electron coordinates and $\mathbf{r}^k \sim p_\theta$.

To optimize the parameters θ , we use K-FAC²⁰ which was already implemented in FermiNet⁴ to minimize the local energy $E_{\text{loc}}(\mathbf{r}^k)$ at the location of electron coordinates $\mathbf{r}^k$. DeepErwin also supports the limited-memory Broyden–Fletcher–

Goldfarb–Shanno algorithm²⁹, which is another second-order method, as well as standard first-order stochastic gradient descent as implemented by the Adam algorithm²¹. After each optimization epoch, (that is, after each sample of coordinates was used exactly once for a gradient descent step), the coordinates $\mathbf{r}^k$ are resampled to reflect the updated probability distribution p_θ . By applying this procedure iteratively for a large number of epochs, we eventually obtain parameters θ such that the wavefunction ψ_θ closely approximates the wavefunction of the ground state for the considered molecule.

To increase numerical stability, the implementation of our ansatz does not directly model ψ_θ , but the logarithm of its square. The local energy can then be computed as

$$E_{\text{loc}}(\mathbf{r}) = E_{\text{pot}}(\mathbf{r}) - \frac{1}{4} \nabla_{\mathbf{r}}^2 \phi_\theta(\mathbf{r}) - \frac{1}{8} (\nabla_{\mathbf{r}} \phi_\theta)^2(\mathbf{r}), \quad (23)$$

where $\phi_\theta = \log(\psi_\theta^2)$.

Although the local energy already contains second derivatives (with respect to $\mathbf{r}$), calculating the gradient with respect to θ does not require the calculation of third derivatives, as H is Hermitian³⁰. Precisely, it holds that

$$\nabla_\theta E_\theta = \langle E_{\text{loc}} \nabla_\theta \phi_\theta \rangle - \langle E_{\text{loc}} \rangle \langle \nabla_\theta \phi_\theta \rangle \quad (24)$$

$$\approx \frac{1}{N} \sum_{k=1}^N E_{\text{loc}}(\mathbf{r}^k) \nabla_\theta \phi_\theta(\mathbf{r}^k) - \frac{1}{N^2} \left(\sum_{k=1}^N E_{\text{loc}}(\mathbf{r}^k) \right) \left(\sum_{k=1}^N \nabla_\theta \phi_\theta(\mathbf{r}^k) \right), \quad (25)$$

for N samples of electron coordinates $\mathbf{r}^k \sim p_\theta$.

To approximate the distribution of electron coordinates defined by the density p_θ , we typically use about 2,000 independent MCMC chains (walkers), where each walker is initialized before optimization with a large number (~1,000) of burn-in steps. During optimization, walkers are updated after every epoch with a small number (~10) of extra MCMC steps. To precisely estimate the energy E_θ after optimization has concluded, we apply a similar procedure, but collect the local energies for all walkers from roughly 1,000 intermediate steps in the respective MCMC chain.

For single steps in the MCMC chains, we use a Metropolis–Hastings algorithm³¹. Given a current walker state $\mathbf{r}$, a proposal state $\mathbf{r}^{\text{prop}}$ for the Metropolis–Hastings algorithm is generated according to the probability density

$$p(\mathbf{r}^{\text{prop}}|\mathbf{r}) = \prod_{i=1}^{n_d} p_{\mathcal{N}}(\mathbf{r}_i^{\text{prop}}|\mu = \mathbf{r}_i, \sigma^2 = (d_i)^2), \quad (26)$$

where $p_{\mathcal{N}}$ denotes the density of a three-dimensional normal distribution and the variance parameters d_i are defined for fixed parameters $d_0, d_{\min}, d_{\max}, \delta \geq 0$ as

$$d_i = \delta \min \left(d_{\min} + \frac{|\mathbf{r}_i - \mathbf{R}_n|}{d_0}, d_{\max} \right), \quad (27)$$

where $\mathbf{R}_n$ denotes the position of the nucleus closest to $\mathbf{r}$, that is, $n = \arg\min_{j=1, \dots, n_{\text{nuc}}} |\mathbf{r}_i - \mathbf{R}_j|$. We then consider the canonical acceptance probability

$$p_{\text{acc}}(\mathbf{r}^{\text{prop}}|\mathbf{r}) = \min \left\{ 1, \frac{p(\mathbf{r}|\mathbf{r}^{\text{prop}})p_\theta(\mathbf{r}^{\text{prop}})}{p(\mathbf{r}^{\text{prop}}|\mathbf{r})p_\theta(\mathbf{r})} \right\}, \quad (28)$$

and for a sample $\alpha \in [0, 1]$ from a uniform distribution, the proposed sample $\mathbf{r}^{\text{prop}}$ is accepted in the case $p_{\text{acc}}(\mathbf{r}^{\text{prop}}|\mathbf{r}) > \alpha$ and rejected otherwise.

The parameters d_i defined in equation (27) can be seen as step size parameters that regulate the average distance between electron coordinates $\mathbf{r}$ and a proposal $\mathbf{r}^{\text{prop}}$. In general, smaller step sizes lead to higher acceptance rates for proposed samples. The parameter δ is a general step size parameter that is gradually adapted during optimization to yield an average acceptance rate of 50%. The second factor in equation (27) was specifically designed to take into account that wavefunctions are usually most complex in the proximity of a nucleus; that is, the step size d_i is chosen for each electron i to depend on its distance to the closest ion to encourage smaller steps close to the nuclei (where the wavefunction varies rapidly) and larger steps (when an electron is further away from the nuclei).

Note that the step sizes d_i and thus the proposal probability, depend on the electron positions $\mathbf{r}$. In general it is therefore not true that $p(\mathbf{r}^{\text{prop}}|\mathbf{r}) = p(\mathbf{r}|\mathbf{r}^{\text{prop}})$. However, due to our choice for the acceptance probability p_{acc} , the so-called detailed balance condition is still satisfied; that is, being in a state $\mathbf{r}$ and transitioning to $\mathbf{r}^{\text{prop}}$ is as probable as being in $\mathbf{r}^{\text{prop}}$ and transitioning to $\mathbf{r}$.

Forces. For a given molecule, let ψ_0 denote the wavefunction of the ground state, that is, ψ_0 is the eigenfunction of the H associated with the smallest eigenvalue. To calculate the electronic forces $\mathbf{F}_m$ acting on the m th nuclei, we apply the Hellmann–Feynman theorem²³ and compute

$$\mathbf{F}_m = -\nabla_{\mathbf{R}_m} E_0 = -\frac{1}{\Omega_0} \int \psi_0(\mathbf{r}) ((\nabla_{\mathbf{R}_m} H) \psi_0)(\mathbf{r}) d\mathbf{r} \quad (29)$$

$$= Z_m \frac{1}{\Omega_0} \int \psi_0(\mathbf{r})^2 \left(\sum_i \frac{\mathbf{r}_i - \mathbf{R}_m}{|\mathbf{r}_i - \mathbf{R}_m|^3} \right) d\mathbf{r} - \sum_{n \neq m} Z_n \frac{\mathbf{R}_n - \mathbf{R}_m}{|\mathbf{R}_n - \mathbf{R}_m|^3} \quad (30)$$

$$\approx Z_m \left(\frac{1}{N} \sum_k \sum_i \frac{\mathbf{r}_i^k - \mathbf{R}_m}{|\mathbf{r}_i^k - \mathbf{R}_m|^3} - \sum_{n \neq m} Z_n \frac{\mathbf{R}_n - \mathbf{R}_m}{|\mathbf{R}_n - \mathbf{R}_m|^3} \right), \quad (31)$$

for N samples of electron coordinates $\mathbf{r}^k \sim \psi_0(\mathbf{r})^2/\Omega_0$, and where $\Omega_0 = \int \psi_0(\mathbf{r})^2 d\mathbf{r}$ denotes the L^2 -norm of ψ_0 .

As it is no longer necessary to compute derivatives of the wavefunction, evaluating equation (31) should be relatively easy. However, unlike the local energy (equation (22)), which has zero variance for eigenstates of the Hamiltonian, naive Monte Carlo sampling cannot be applied in this case due to the divergence of $|\mathbf{r}_i - \mathbf{R}_i|^{-3}$ when the i th electron approaches the k th nucleus.

This issue can be addressed by observing that for each diverging term on one side of a nucleus, there is an equally diverging term with the opposite sign on the other side of the nucleus. Different variance reduction methods have been proposed to exploit this property, such as fitting the force density close to a nucleus with a function that is constrained to be zero at $\mathbf{r}_i = \mathbf{R}_k$ (ref. ³²) or antithetic sampling^{32,33}.

We minimize the variance of force samples by combining antithetic sampling with a truncated $1/r$ potential: for each sample of electron coordinates $\mathbf{r}^k$ yielded by a Markov chain, we also consider an additional sample $\tilde{\mathbf{r}}^k$ in which each electron within a distance of R_{core} to the closest nucleus is mirrored to the opposite side of this nucleus.

$$\tilde{\mathbf{r}}^{k,i} = \mathbf{r}_i^k + 2(\mathbf{R}_m - \mathbf{r}_i^k) \quad (32)$$

for $i \in \{1, \dots, n_d\}$ and $m = \arg\min_n |\mathbf{r}_i - \mathbf{R}_n|$; therefore, whenever an electron comes close to a nucleus and thus generates a large force in one direction, there will always be a cancelling contribution in the opposite direction. Note that the samples $\tilde{\mathbf{r}}^{k,i}$ do not necessarily have the same probability as the original sample $\mathbf{r}^k$. Their contribution to the Monte Carlo estimate is thus weighted by $\frac{\psi(\tilde{\mathbf{r}}^{k,i})^2}{\psi(\mathbf{r}^k)^2}$. Furthermore, we avoid numerical instabilities by replacing the raw Coulomb forces with a scaled version that decays towards zero, as electrons approach a nucleus:

$$\frac{\mathbf{r}_i^k - \mathbf{R}_m}{|\mathbf{r}_i^k - \mathbf{R}_m|^3} \rightarrow \frac{\mathbf{r}_i^k - \mathbf{R}_m}{|\mathbf{r}_i^k - \mathbf{R}_m|^3} \tanh^3 \left(\frac{|\mathbf{r}_i^k - \mathbf{R}_m|}{R_{\text{cut-off}}} \right) \quad (33)$$

Shared optimization of model parameters. For N distinct nuclear geometries $\mathbf{R}^1, \mathbf{R}^2, \dots, \mathbf{R}^N$, let θ^{sh} denote the set of model parameters that are shared across all geometries, and θ^k denote the set of parameters that are specific to the k th set of nuclear coordinates $\mathbf{R}^k$. The full set of parameters for the k th geometry can then be written as $\theta^k = (\theta^{\text{sh}}, \theta^k)$, and the associated realization of the wavefunction model is denoted by ψ_{θ^k} . During each shared optimization epoch, we consider a single nuclear geometry $\mathbf{R}^k$ and update the respective model weights θ^k with respect to the local energies of the wavefunction ψ_{θ^k} at MCMC walker positions that were sampled from the probability density p_{θ^k} (equation (20)). That is, during each shared optimization epoch, not only the geometry-specific weights are updated, but also the shared weights θ^{sh} , and therefore the wavefunctions for all nuclear geometries.

A straightforward way of deciding which geometry to consider for a shared optimization epoch is to employ a simple round-robin scheme. Note that for each eigenstate of the Hamiltonian, the local energies defined in equation (21) have zero variance. Therefore, another approach to select geometries for a shared optimization epoch is to use the standard deviation of local energies as a proxy of how closely a realization of the wavefunction model already approximates the wavefunction of the ground state. By selecting the geometry for which the current wavefunction has the highest standard deviation in the local energies, we ensure that geometries for which the parameters have not yet been well optimized get more attention during optimization. In practice, we train a few initial epochs using the round-robin scheme to obtain a reliable starting point for each wavefunction and then switch to the standard-deviation-based scheme, to ensure homogeneous convergence of the accuracy across all geometries.

In our implementation of the shared optimization scheme, we use a single instance of the optimizer to update the set of shared weights θ^{sh} , as well as an additional instance for each of the geometry-specific sets of model parameters. In particular, due to the fact that geometry-specific weights receive fewer gradient descent updates than shared weights, the learning rate for the geometry-specific optimizers are usually chosen about ten times larger than the learning rate for the optimizer of the shared weights (Supplementary Table 1).

After the shared optimization process, the wavefunctions can again be treated as fully independent realizations of our wavefunction model and are not constrained by any additional dependencies. In particular, this means that the evaluation of their exact energy can easily be parallelized across geometries.

Reference calculations. To validate our deep learning method, we compared the obtained energies to reference values, which we computed using the MOLPRO package^{34,35}. We employed both single- and multi-reference explicitly correlated F12 methods: coupled cluster with singles, doubles and perturbative triples (CCSD(T)-F12)³⁶ and MRCI-F12³⁷. The basis set cc-pVQZ-F12³⁸ was used for all geometries of H_4^+ , H_6 and C_2H_4 . Due to convergence issues with this basis set for some geometries of H_{10} , the smaller cc-pVTZ-F12 basis set was employed for this molecular system. For the MRCI-F12 calculations, a CASSCF reference was used with a full-valence active space (H_4^+ : three electrons in three orbitals abbreviated as (3, 3), H_6 : (6, 6), H_{10} : (10, 10)). A full-valence active space was prohibitively expensive for C_2H_4 , which is why we resorted to an active space of (2, 2). The recommended GEM_BETA coefficients were used (that is, for MRCI-F12(Q)/cc-pVQZ-F12 calculations GEM_BETA = $1.5 a_0^{-1}$ and for MRCI-F12(Q)/cc-pVTZ-F12 calculations GEM_BETA = $1.4 a_0^{-1}$)³⁹. The Davidson-corrected⁴⁰ energy values (MRCI-F12(Q)) were extracted, as provided by the energy variable in MOLPRO.

We note that these accurate methods and basis sets were only used to validate our results, but not as a starting point for our deep learning method. The orbitals used as a starting point for our method were generated using the CASSCF method implemented in PySCF⁴¹, using a 6-311G Pople basis set.

Computational settings for DeepErwin. Reference values for the results shown in Fig. 2 were obtained via MRCI-F12(Q)/cc-pVQZ-F12. A detailed description of these calculations can be found in the previous section.

In all cases, we used the described method from the previous section as a reference; however, two geometries were excluded from the reference set for the initialization from the H_{10} run, as these were also included in the pretrained geometry grid.

The main hyperparameters used for all computations are listed in Supplementary Table 1. Detailed counts of total, trainable and shared parameters for the used wavefunction models for the four main molecules considered in our numerical experiments are compiled in the Supplementary Table 2.

Data availability

All data in this manuscript were generated using the Python package DeepErwin or the quantum-chemistry code MOLPRO as described in Methods. All data required to perform the reported calculations as well as the processed data that was used to generate figures are available on Code Ocean⁴². Source data are provided with this paper.

Code availability

The DeepErwin package alongside a detailed documentation is available on the Python Package Index (PyPI) and GitHub (<https://github.com/mdsunivie/deeperwin>) under the MIT license. All codes and configuration files that were used to perform the reported calculations are also available on Code Ocean⁴².

Received: 27 May 2021; Accepted: 16 March 2022;

Published online: 19 May 2022

References

- Han, J., Zhang, L. & E, W. Solving many-electron Schrödinger equation using deep neural networks. *J. Comput. Phys.* **399**, 108929 (2019).
- Hermann, J., Schätzle, Z. & Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.* **12**, 891–897 (2020).
- Manzhos, S. Machine learning for the solution of the Schrödinger equation. *Mach. Learn. Sci. Technol.* **1**, 013002 (2020).
- Pfau, D., Spencer, J. S., Matthews, A. G. D. G. & Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Res.* **2**, 033429 (2020).
- Wilson, M., Gao, N., Wudarski, F., Rieffel, E. & Tubman, N. M. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo. *Phys. Rev. Res.* **4**, 013021 (2021).
- Bartlett, R. J. & Musiał, M. Coupled-cluster theory in quantum chemistry. *Rev. Mod. Phys.* **79**, 291–352 (2007).
- Spencer, J. S., Pfau, D., Botev, A. & Foulkes, W. M. C. Better, faster fermionic neural networks. Preprint at <https://arxiv.org/abs/2011.07125> (2020).
- Unke, O. T. et al. Machine learning force fields. *Chem. Rev.* **121**, 10142–10186 (2021).
- Behler, J. Four generations of high-dimensional neural network potentials. *Chem. Rev.* **121**, 10037–10072 (2021).
- Kirkpatrick, J. et al. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science* **374**, 1385–1389 (2021).
- Westermayr, J. & Marquetand, P. Machine learning for electronically excited states of molecules. *Chem. Rev.* **121**, 9873–9926 (2020).
- Schütt, K. T., Gastegger, M., Tkatchenko, A., Müller, K.-R. & Maurer, R. J. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nat. Commun.* **10**, 5024 (2019).
- Bogojeski, M., Vogt-Maranto, L., Tuckerman, M. E., Müller, K.-R. & Burke, K. Quantum chemical accuracy from density functional approximations via machine learning. *Nat. Commun.* **11**, 5223 (2020).
- Faber, F. A. et al. Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: pre-training of deep bidirectional transformers for language understanding. Preprint at <https://arxiv.org/abs/1810.04805> (2018).
- Tan, C. et al. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks* 270–279 (Springer, 2018).
- Matthews, D. A. Analytic gradients of approximate coupled cluster methods with quadruple excitations. *J. Chem. Theory Comput.* **16**, 6195–6206 (2020).
- Schütt, K. et al. SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Proc. 31st Conference on Neural Information Processing Systems* (eds Guyon, I. et al.) 992–1002 (Curran Associates, 2017).
- Ma, A., Towler, M. D., Drummond, N. D. & Needs, R. J. Scheme for adding electron–nucleus cusps to Gaussian orbitals. *J. Chem. Phys.* **122**, 224322 (2005).
- Martens, J. & Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning* 2408–2417 (PMLR, 2015).
- Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).
- Aljiah, A. & Varandas, A. J. C. H_4^+ : what do we know about it? *J. Chem. Phys.* **129**, 034303 (2008).
- Feynman, R. P. Forces in molecules. *Phys. Rev.* **56**, 340–343 (1939).
- Peierls, R. E. & Peierls, R. S. *Quantum Theory of Solids* (Oxford Univ. Press, 1955).
- Pulay, P. Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules. *Mol. Phys.* **17**, 197–204 (1969).
- Gao, N. & Günnemann, S. Ab-initio potential energy surfaces by pairing GNNs with neural wave functions. In *International Conference on Learning Representations* (2022).
- Ríos, P. L., Ma, A., Drummond, N. D., Towler, M. D. & Needs, R. J. Inhomogeneous backflow transformations in quantum Monte Carlo calculations. *Phys. Rev. E* **74**, 066701 (2006).
- Kato, T. On the eigenfunctions of many-particle systems in quantum mechanics. *Commun. Pure Appl. Math.* **10**, 151–177 (1957).
- Liu, D. C. & Nocedal, J. On the limited memory bfgs method for large scale optimization. *Math. Prog.* **45**, 503–528 (1989).
- Ceperley, D., Chester, G. V. & Kalos, M. H. Monte Carlo simulation of a many-fermion study. *Phys. Rev. B* **16**, 3081–3099 (1977).
- Hastings, W. K. Monte carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
- Chiesa, S., Ceperley, D. M. & Zhang, S. Accurate, efficient, and simple forces computed with quantum monte carlo methods. *Phys. Rev. Lett.* **94**, 036404 (2005).
- Kalos, M. H. & Whitlock, P. A. *Monte Carlo Methods* (Wiley, 1986); <https://cds.cern.ch/record/109491>
- Werner, H.-J., Knowles, P. J., Knizia, G., Manby, F. R. & Schütz, M. MOLPRO: a general-purpose quantum chemistry program package. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2**, 242–253 (2012).
- Werner, H.-J. et al. MOLPRO, version 2012.1. A package of ab initio programs (MOLPRO, 2012); <https://www.molpro.net>
- Adler, T. B., Knizia, G. & Werner, H.-J. A simple and efficient CCSD(T)-F12 approximation. *J. Chem. Phys.* **127**, 221106 (2007).
- Shiozaki, T., Knizia, G. & Werner, H.-J. Explicitly correlated multireference configuration interaction: MRCI-F12. *J. Chem. Phys.* **134**, 034113 (2011).
- Peterson, K. A., Adler, T. B. & Werner, H.-J. Systematically convergent basis sets for explicitly correlated wavefunctions: the atoms H, He, B–Ne, and Al–Ar. *J. Chem. Phys.* **128**, 084102 (2008).
- Hill, J. G., Mazumder, S. & Peterson, K. A. Correlation consistent basis sets for molecular core-valence effects with explicitly correlated wave functions: the atoms B–Ne and Al–Ar. *J. Chem. Phys.* **132**, 054108 (2010).
- Langhoff, S. R. & Davidson, E. R. Configuration interaction calculations on the nitrogen molecule. *Int. J. Quantum Chem.* **8**, 61–72 (1974).
- Sun, Q. et al. PySCF: the Python-based simulations of chemistry framework. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **8**, e1340 (2018).
- Scherbela, M., Reisenhofer, R., Gerard, L., Marquetand, P. & Grohs, P. DeepErwin—a framework for solving the Schrödinger equation with deep neural networks. *CodeOcean* <https://doi.org/10.24433/CO.8193370.v1> (2022).

Acknowledgements

We gratefully acknowledge financial support from the following grants: Austrian Science Fund FWF-I-3403 (L.G.), FWF-M-2528 (R.R.) and WWTF-ICT19-041 (L.G.). The

computational results have been achieved using the Vienna Scientific Cluster (VSC). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Author contributions

P.G., P.M. and R.R. conceived the project. M.S., L.G. and R.R. developed the detailed method. M.S. and L.G. wrote the Python code with contributions from R.R. The numerical experiments were designed and performed by M.S., L.G. and P.M. with support from R.R. R.R., M.S. and L.G. wrote the manuscript with input from P.G. and P.M. P.G. supervised the project. R.R. and P.G. obtained funding.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s43588-022-00228-x>.

Correspondence and requests for materials should be addressed to Rafael Reisenhofer.

Peer review information *Nature Computational Science* thanks Huan Tran, Linfeng Zhang and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available. Primary Handling Editor: Jie Pan, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2022

Supplementary information

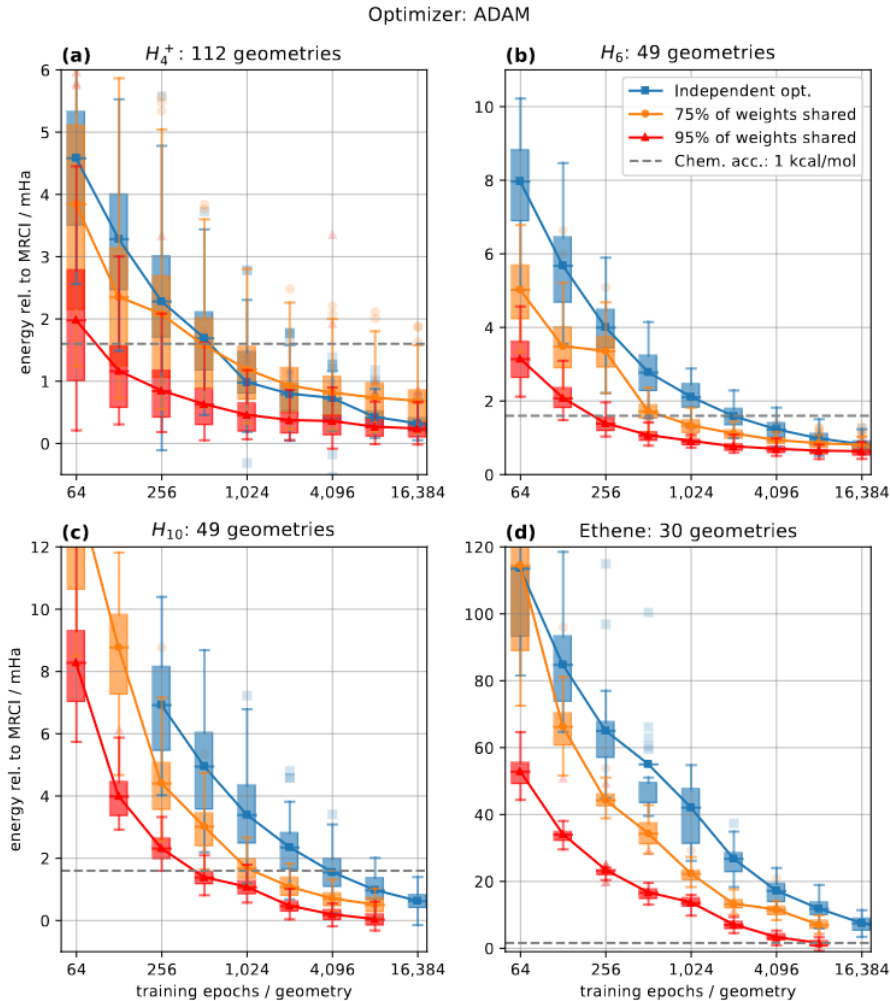
Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks

In the format provided by the authors and unedited

1 Supplementary information

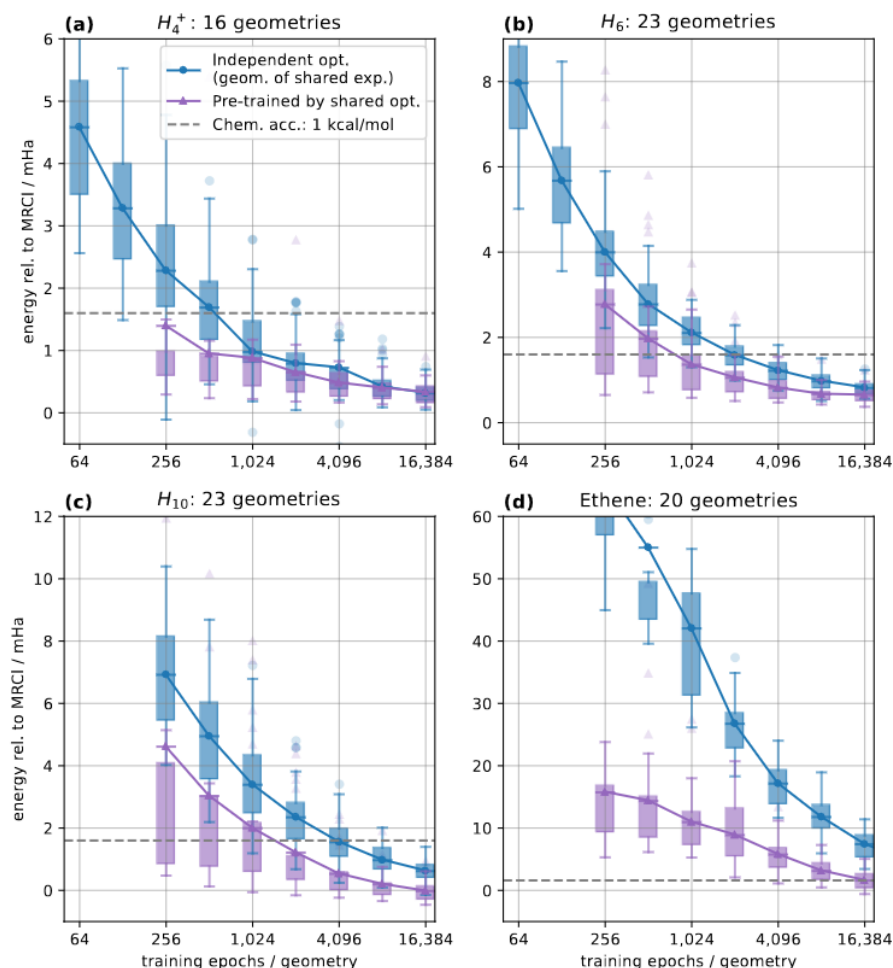
1.1 Weight sharing and pre training with ADAM

Supplementary Figures 1 and 2 show results for the experiments presented in the main text in Figures 2 and 3 when using the first-order ADAM optimizer [1] instead of second-order K-FAC optimization.



Supplementary Figure 1: Weight-sharing when using first-order optimization (ADAM). The plots depict the evaluation error relative to the reference calculation (MRCI-F12(Q)/cc-pVQZ-F12) as a function of training epochs per geometry for four different sets of nuclear geometries. Box-plots show the 25-75 percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 inter-quartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. (a) Weight-sharing for 112 H_4^+ geometries (b) Weight-sharing for 49 H_6 geometries, (c) Weight-sharing for 49 H_{10} geometries. (d) Weight-sharing for 30 ethene geometries.

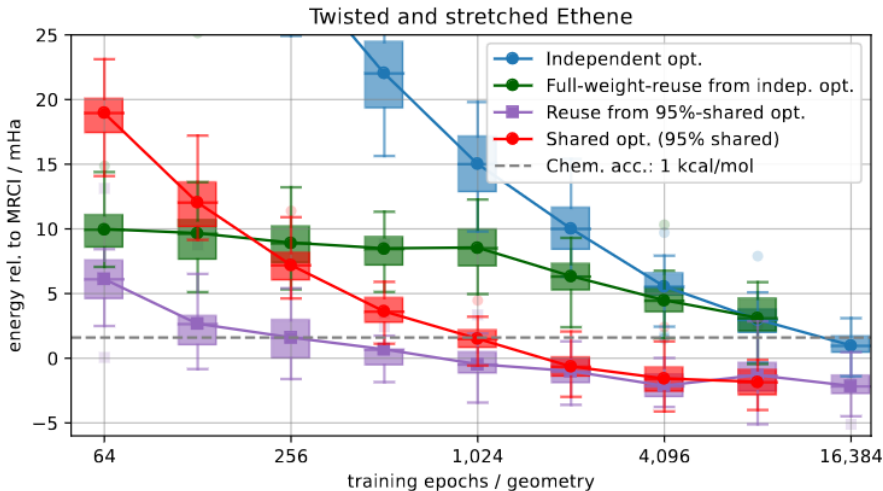
Optimizer: ADAM



Supplementary Figure 2: Pre-training with shared optimization when using first-order optimization (ADAM). The plots depict the evaluation error relative to a reference calculation (MRCI-F12(Q)/cc-pVQZ-F12 (Molpro)) as a function of training epochs per geometry for four different sets of nuclear geometries. Plots show energy errors for pre-training on different geometries of the same molecules compared to a standard non-pretrained optimization. Box-plots show the 25-75 percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 inter-quartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. (a) Re-using weights for 16 H_4^+ geometries (b) Re-using weights for 23 H_6 geometries (c) Re-using weights for 23 H_{10} geometries. (d) Re-using weights for 20 ethene geometries.

1.2 Independent optimization as pre training

In the main text, mere independent optimization was used as a benchmark for all numerical experiments. Here, we consider an additional baseline, for which independent optimizations for different ethene configurations were fully initialized with weights from a wavefunction that had already been optimized for a similar but different molecular configuration using an independent optimization scheme. Optimization hyperparameters for the new benchmark are the same as the ones reported in Supplementary Table 1. The respective results, including a comparison with shared optimization and independent optimization that was pre-trained on a set of different ethene configurations using shared-weight constraints, are shown in Supplementary Figure 3. While the new benchmark seems to be advantageous during early optimization as compared to a scheme that applies a weight-sharing constraint for 95% of the weights in the model, at the time the wavefunctions reach chemical accuracy, shared optimization without pre-training outperforms this new baseline to a degree comparable with the results shown in Figure 2 in the main text.



Supplementary Figure 3: Comparison of two different schemes for pre-training. The plots depict the evaluation error relative to the reference calculation (MRCI-F12(Q)/cc-pVQZ-F12) as a function of training epochs per geometry for a set of differently twisted and stretched ethene configurations. Box-plots show the 25-75 percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 inter-quartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually.

1.3 Tables

The DeepErwin hyperparameters for all numerical experiments reported in the main text are listed in Supplementary Table 1. Detailed counts of total, trainable, and shared parameters for the used wavefunction models for the four main molecules considered in our numerical experiments are compiled in Supplementary Table 2. The optimization hyperparameters for first-order ADAM optimization, which was used for the calculations presented in Supplementary Figures 1 and 2, are shown in Supplementary Table 3.

Supplementary Table 1: Hyperparameter settings for DeepErwin. As baseline method, we use complete active space self consistent field (CASSCF). The applied optimization method is Kronecker-factored approximate curvature (K-FAC). Initial learning rates are denoted by lr_0 , and the change over time by $\text{lr}(t)$.

Baseline method (CASSCF)	# determinants	20
	Basis set	6-311G
Embedding	# radial basis features n_{rbf}	16
	# hidden layers g	1
	# hidden layers w	2
	# hidden layers f	2
	# neurons per layer f, w, g	40
	# iterations L	2
	Embedding dimension x	64
Back ow / Jastrow	# hidden layers $_{\text{gen}, s, J}$	2
	# neurons per layer $_{\text{gen}, s, J}$	40
	# hidden layers $_i$	0
	# neurons per layer $_i$	1
Markov Chain Monte Carlo	# walkers	2048
	# decorrelation steps	5
	Target acceptance probability	50%
Optimization general	Optimizer	KFAC
	Damping	$5 \cdot 10^{-4}$
	Norm constraint	$1 \cdot 10^{-3}$
	Batch size	512
Independent optimization	Initial learning rate lr_0	$2 \cdot 10^{-3}$
	Learning rate decay	$\text{lr}(t) = \text{lr}_0(1 + t/1000)^{-1}$
Shared optimization (Figure 2)	lr_0 for non-shared weights	$5 \cdot 10^{-3}$
	lr_0 for shared weights	$5 \cdot 10^{-4}$
	Learning rate decay	$\text{lr}(t) = \text{lr}_0(1 + t/10000)^{-1}$
	Training scheduler	Std. deviation
Opt. after pre-training (Figure 3)	lr_0 for non-shared pre-trained weights	$2 \cdot 10^{-3}$
	lr_0 for shared pre-trained weights	$2 \cdot 10^{-4}$
	Learning rate decay	$\text{lr}(t) = \text{lr}_0(1 + t/1000)^{-1}$
Force evaluation	Antithetic sampling radius R_{core}	0.2
	Coulomb cutoff $R_{\text{cut-o}}$	0.01

Supplementary Table 2: Counts of total, trainable and shared model parameters for each of the four main molecules considered in the numerical experiments.

Molecule	Optimization	# total	# non-trainable	# trainable	# shared
H₄⁺	independent	107,548	368 (0.3 %)	107,180	0 (0.0 %)
	75 % shared	107,548	368 (0.3 %)	107,180	83,673 (77.8 %)
	95 % shared	107,548	368 (0.3 %)	107,180	105,920 (98.5 %)
H₆	independent	109,228	788 (0.7 %)	108,440	0 (0.0 %)
	75 % shared	109,228	788 (0.7 %)	108,440	83,673 (76.6 %)
	95 % shared	109,228	788 (0.7 %)	108,440	105,920 (97.0 %)
H₁₀	independent	112,140	2,020 (1.8 %)	110,120	0 (0.0 %)
	75 % shared	112,140	2,020 (1.8 %)	110,120	83,673 (74.6 %)
	95 % shared	112,140	2,020 (1.8 %)	110,120	105,920 (94.5 %)
Ethene	independent	115,868	3,228 (2.8 %)	112,640	0 (0.0 %)
	75 % shared	115,868	3,228 (2.8 %)	112,640	83,673 (72.2 %)
	95 % shared	115,868	3,228 (2.8 %)	112,640	105,920 (91.4 %)

Supplementary Table 3: Optimization hyperparameters for DeepErwin when using first-order ADAM optimization.

Optimization general	Optimizer	ADAM
	Batch size	512
Independent optimization	Initial learning rate lr_0	1.5 10
	Learning rate decay	$lr(t) = \frac{lr}{1+t/1000}$
Shared optimization (Supplementary Fig. 1)	lr_0 for non-shared weights	1.5 10
	lr_0 for shared weights	1.5 10
	Learning rate decay	$lr(t) = lr_0(1 + t/10000)^{-1}$
	Training scheduler	Std. deviation
Opt. after pre-training (Supplementary Fig. 2)	lr_0 for non-shared pre-trained weights	1.5 10
	lr_0 for shared pre-trained weights	1.5 10
	Learning rate decay	$lr(t) = lr_0(1 + t/1000)^{-1}$

Changes to the hyperparameters from Supplementary Table 1 are written in boldface.

References

- [1] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

III Towards a transferable fermionic neural wavefunction for molecules

Comments

E-Print: [arXiv:2303.09949](https://arxiv.org/abs/2303.09949)[physics.comp-ph]

Authorship: L.G. and M.S. share first authorship.

Contribution

M.S., L.G., and P.G. conceived the project. M.S. conceived and implemented the ansatz, built the dataset and designed the experiments. L.G. gave input on the ansatz and worked on implementation. M.S. and L.G. performed the experiments. M.S. and L.G. wrote the manuscript with input, supervision and funding from P.G.

Bibliographic Information

Scherbela, Michael, Leon Gerard, and Philipp Grohs (2024). “Towards a transferable fermionic neural wavefunction for molecules.” In: *Nature Communications* 15.1, p. 120. DOI: [10.1038/s41467-023-44216-9](https://doi.org/10.1038/s41467-023-44216-9).

Towards a transferable fermionic neural wavefunction for molecules

Received: 22 March 2023

Accepted: 5 December 2023

Published online: 02 January 2024

 Check for updatesMichael Scherbela^{1,4}, Leon Gerard^{2,4} & Philipp Grohs^{1,2,3} 

Deep neural networks have become a highly accurate and powerful wavefunction ansatz in combination with variational Monte Carlo methods for solving the electronic Schrödinger equation. However, despite their success and favorable scaling, these methods are still computationally too costly for wide adoption. A significant obstacle is the requirement to optimize the wavefunction from scratch for each new system, thus requiring long optimization. In this work, we propose a neural network ansatz, which effectively maps uncorrelated, computationally cheap Hartree-Fock orbitals, to correlated, high-accuracy neural network orbitals. This ansatz is inherently capable of learning a single wavefunction across multiple compounds and geometries, as we demonstrate by successfully transferring a wavefunction model pre-trained on smaller fragments to larger compounds. Furthermore, we provide ample experimental evidence to support the idea that extensive pre-training of such a generalized wavefunction model across different compounds and geometries could lead to a foundation wavefunction model. Such a model could yield high-accuracy ab-initio energies using only minimal computational effort for fine-tuning and evaluation of observables.

Accurate predictions of quantum mechanical properties for molecules is of utmost importance for the development of new compounds, such as catalysts, or pharmaceuticals. For each molecule the solution to the Schrödinger equation yields the wavefunction and electron density, and thus in principle gives complete access to its chemical properties. However, due to the curse of dimensionality, computing accurate approximations to the Schrödinger equation quickly becomes computationally intractable with increasing number of particles. Recently, deep-learning-based Variational Monte Carlo (DL-VMC) methods have emerged as a high-accuracy approach with favorable scaling $\mathcal{O}(N^4)$ in the number of particles N . These methods use a deep neural network as ansatz for the high-dimensional wavefunction, and minimize the energy of this ansatz to obtain the ground-state wavefunction. Based on two major architectures for the treatment of molecules in first quantization, PauliNet¹ and FermiNet², several improvements and applications have emerged. On the one hand, enhancements of architecture, optimization and overall approach have led to substantial

improvements in accuracy or computational cost^{3–7}. On the other hand, these methods have been adapted to many different systems and observables: model systems of solids^{8,9}, real solids¹⁰, energies and properties of individual molecules^{1,2,5,11}, forces^{12,13}, excited states¹⁴ and potential energy surfaces^{13,15,16}. Furthermore, similar methods have been developed and successfully applied to Hamiltonians in second quantization^{17,18}.

We want to emphasize that DL-VMC is an ab-initio method, that does not require any input beyond the Hamiltonian, which is defined by the molecular geometry. This differentiates it from surrogate models, which are trained on results from ab-initio methods to either predict wavefunctions^{19,20} or observables²¹.

Despite the improvements in DL-VMC, it has not yet been widely adopted, in part due to the high computational cost. While DL-VMC offers favorable scaling, the method suffers from a large prefactor, caused by an expensive optimization with potentially slow convergence towards accurate approximations. Furthermore this

¹Faculty of Mathematics, University of Vienna, Vienna, Austria. ²Research Network Data Science, University of Vienna, Vienna, Austria. ³Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria. ⁴These authors contributed equally: Michael Scherbela, Leon Gerard. ✉ e-mail: philipp.grohs@univie.ac.at

optimization needs to be repeated for every new system, leading to prohibitively high computational cost for large-scale use. This can be partially overcome by sharing a single ansatz with identical parameters across different geometries of a compound, allowing more efficient computation of Potential Energy Surfaces (PES)^{13,15,16}. However, these approaches have been limited to different geometries of a single compound and do not allow successful transfer to new compounds. A key reason for this limitation is that current architectures explicitly depend on the number of orbitals (and thus electrons) in a molecule. Besides potential generalization issues, this prevents a transfer of weights between different compounds already by the fact that the shapes of weight matrices are different for compounds of different size.

In this work we propose a neural network ansatz, which does not depend explicitly on the number of particles, allowing to optimize wavefunctions across multiple non-periodic, gas-phase compounds with multiple different geometric conformations. We find, that our model exhibits strong generalization when transferring weights from small molecules to larger, similar molecules. In particular we find that our method achieves high accuracy for the important task of relative energies. Our approach is inspired by the success of foundation models in language²² or vision^{23,24}, where models are extensively pre-trained and then applied to new tasks—either without any subsequent training (referred to as zero-shot evaluation) or after small amount of training on the new task (referred to as fine-tuning). Zhang et al.²⁵ have shown that this paradigm can be successfully applied to wavefunctions, in their case for model Hamiltonians in second quantization.

In this work, we pre-train a first base-model for neural network wavefunctions in first quantization, and evaluate the pre-trained model by performing predictions on chemically similar molecules (in-distribution) and disparate molecules (out-of-distribution). We find that our ansatz outperforms conventional high-accuracy methods such as CCSD(T)-ccpVTZ and that fine-tuning our pre-trained model reaches this accuracy $\approx 20\times$ faster, than optimizing a new model from scratch. When analyzing the accuracy as a function of pre-training resources, we find that results systematically and substantially improve by scaling up either the model size, data size or number of pre-training steps. These results could pave the way towards a foundation wavefunction model, to obtain high-accuracy ab-initio results of quantum mechanical properties using only minimal computational effort for fine-tuning and evaluation of observables.

Additionally we compare our results to GLOBE, a concurrent work²⁶, which proposes reparameterization of the wavefunction based on machine-learned, localized molecular orbitals. We find that for the investigated setting of re-using pre-trained weights our method in comparison achieves lower (and thus more accurate) absolute energies, higher accuracy of relative energies and is better able to generalize across chemically different compounds.

We use the following notation throughout this work: All vectors, matrices and tensors are denoted in **bold letters**, including functions with vectorial output. The i -th electron position for $i \in \{1, \dots, n_{\text{el}}\}$ is denoted as $\mathbf{r}_i \in \mathbb{R}^3$. The set $\{\mathbf{r}_1, \dots, \mathbf{r}_{n_{\text{el}}}\}$ of all electrons with spin up is abbreviated with $\{\mathbf{r}_{\uparrow}\}$, the set $\{\mathbf{r}_{n_{\uparrow}+1}, \dots, \mathbf{r}_{n_{\text{el}}}\}$ of all spin-down electrons as $\{\mathbf{r}_{\downarrow}\}$. Similarly, the nuclear positions and charges of a molecule are denoted by $\mathbf{R}_I \in \mathbb{R}^3$ and $Z_I \in \mathbb{N}$, $I=1, \dots, N_{\text{atoms}}$, with the set of all nuclear positions and their corresponding charges denoted as $\{(\mathbf{R}, \mathbf{Z})\}$. Indices $i, k \in \{1, \dots, n_{\text{el}}\}$ correspond to electrons and orbitals respectively, whereas $I, J \in \{1, \dots, N_{\text{atoms}}\}$ correspond to nuclei. By $\langle \cdot, \cdot \rangle$ the dot product is denoted.

Results

In the following, we briefly outline our approach and how it extends existing work (A multi-compound wavefunction ansatz). We show the fundamental properties of our ansatz such as extensivity and equivariance with respect to the sign of reference orbitals. We demonstrate

the transferability of the ansatz when pre-training on small molecules and re-using it on larger, chemically similar molecules. We also compare its performance against GLOBE, a concurrent work²⁶. Lastly, we present a first wavefunction base model pre-trained on a large diverse dataset of 360 geometries and evaluate its downstream performance.

A multi-compound wavefunction ansatz

Existing high-accuracy ansätze for neural network wavefunctions ψ all exhibit the following structure:

$$\mathbf{h}_i = \mathbf{h}_{\theta}(\mathbf{r}_i, \{\mathbf{r}_{\uparrow}\}, \{\mathbf{r}_{\downarrow}\}, \{(\mathbf{R}, \mathbf{Z})\})$$

$$\mathbf{h}_i \in \mathbb{R}^{D_{\text{emb}}}, \quad i = 1 \dots n_{\text{el}} \quad (1)$$

$$\Phi_{ik}^d = \varphi_k^d(\mathbf{r}_i) \langle \mathbf{F}_k^d, \mathbf{h}_i \rangle$$

$$\varphi_k^d(\mathbf{r}_i) : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad \mathbf{F}_k^d \in \mathbb{R}^{D_{\text{emb}}}$$

$$i, k = 1 \dots n_{\text{el}}, \quad d = 1 \dots N_{\text{det}} \quad (2)$$

$$\psi = \sum_{d=1}^{N_{\text{det}}} \det [\Phi_{ik}^d]_{i,k=1 \dots n_{\text{el}}} \quad (3)$$

The neural network $\mathbf{h}_{\theta}$ in eq. (1) computes a D_{emb} -dimensional embedding $\mathbf{h}_i$ of electron i , by taking in information of all other particles, e.g., by using attention or message passing. Eq. (2) maps these high-dimensional embeddings onto $n_{\text{el}} \times N_{\text{det}}$ orbitals (indexed by k), using trainable backflow matrices $\mathbf{F}^d$ and typically trainable envelope functions $\varphi_k^d : \mathbb{R}^3 \rightarrow \mathbb{R}$. Eq. (3) evaluates the final wavefunction ψ as a sum of (Slater-)determinants of these orbitals, to ensure anti-symmetry with respect to permutation of electrons.

By considering the interaction of all particles, in particular with the sets $\{\mathbf{r}_{\uparrow}\}$ and $\{\mathbf{r}_{\downarrow}\}$, the functions in Eq. (2) account for the inter-particle correlation and therefore are able to better represent the true ground-state wavefunction. If the orbitals Φ_{ik}^d would only depend on $\mathbf{r}_i$ (instead of the many-body embedding $\mathbf{h}_i$), they would correspond to single-particle functions, e.g. Hartree-Fock orbitals. Existing methods, as proposed in Pfau et al.² or Hermann et al.¹, differ in the way the embedding $\mathbf{h}_i$ and the envelope functions φ_k^d are built. A popular choice for the embedding function $\mathbf{h}_{\theta}$ are continuous convolutions^{1,5,26} or an attention mechanism⁴. For the envelope functions Hermann et al.¹ proposed to use orbitals obtained from a Hartree-Fock calculation, whereas Pfau et al.² relied on exponentially decaying envelopes, (i.e., $\varphi_k(\mathbf{r}) = \exp(-\alpha_{kl}|\mathbf{r} - \mathbf{R}_l|)$ with trainable parameter $\alpha_{kl} \in \mathbb{R}$), to ensure the boundary conditions far away from the nuclei. In our architecture, we mainly focus on the matrix $\mathbf{F}^d = [\mathbf{F}_1^d, \dots, \mathbf{F}_{n_{\text{el}}}^d] \in \mathbb{R}^{n_{\text{el}} \times D_{\text{emb}}}$. While the mapping by $\mathbf{F}^d$ works well for the wavefunction of a single compound, it is fundamentally unsuited to represent wavefunctions for multiple different compounds at once, since its dimension $n_{\text{el}} \times D_{\text{emb}}$ depends explicitly on the number of electrons. There are several potential options, how this challenge could be overcome. A naïve approach would be to generate a fixed number of $N_{\text{orb}} \geq n_{\text{el}}$ orbitals and truncate the output to the required number of orbitals n_{el} , which may differ across molecules. While simple to implement, this approach is however fundamentally limited to molecules with $n_{\text{el}} \leq N_{\text{orb}}$. Another approach is to use separate matrices $\mathbf{F}_{\mathcal{G}}^d$ for each molecule or geometry $\mathcal{G}$, as was done in¹³, but also this approach can fundamentally not represent wavefunctions for molecules that are larger than the ones found in the training set. A third approach would be to not generate all orbitals in a single pass, but generate the orbitals sequentially in an auto-regressive manner, by conditioning each orbital on the previously generated orbitals. While this approach has been successful in other domains such as language processing, it suffers from inherently poor parallelization due to its sequential nature. A final approach—chosen in this work—is to replace the tensor $\mathbf{F}$ with a trainable function $\mathbf{f}_{\theta}^o(\mathbf{c}_{ik})$, which computes the

backflows based on some descriptor $\mathbf{c}_{Ik} \in \mathbb{R}^{N_{\text{basis}}}$ of the orbital k to be generated:

$$\varphi_{\theta}^d(\mathbf{r}_i, \mathbf{R}_I, \mathbf{c}_{Ik}) = \exp(-|\mathbf{r}_i - \mathbf{R}_I| g_{\theta}^{e,d}(\mathbf{c}_{Ik})) \quad (4)$$

$$\Phi_{ik}^d = \sum_{I=1}^{N_{\text{atoms}}} \varphi_{\theta}^d(\mathbf{r}_i, \mathbf{R}_I, \mathbf{c}_{Ik}) \langle \mathbf{f}_{\theta}^{o,d}(\mathbf{c}_{Ik}), \mathbf{h}_i \rangle \quad (5)$$

Similar to $\mathbf{f}_{\theta}^o$, the trainable function $g_{\theta}^{e,d}$ also maps some descriptors to scalar values for each orbital. While there are several potential descriptors $\mathbf{c}_{Ik}$ for orbitals, one particularly natural choice is to use outputs of computationally cheap, conventional quantum chemistry methods such as Density Functional Theory or Hartree-Fock. We compute orbital features based on the expansion coefficients of a Hartree-Fock calculation, by using orbital localization and a graph convolutional network (GCN), as outlined in the methods section “Obtaining orbital descriptors from Hartree-Fock”. We then map these features to orbitals Φ_{ik}^d , which we call in the following *transferable atomic orbitals (TAOs)*, using odd and even functions $\mathbf{f}_{\theta}^o$ and g_{θ}^e as illustrated in Fig. 1.

Properties of our ansatz

These TAOs fulfill many properties, which are desirable for a wavefunction ansatz:

- **Constant parameter count:** In principle, the number of parameters in the ansatz is independent of system size. While it might still be necessary to increase the parameter count to maintain uniform accuracy for systems of increasing size, TAOs have no explicit relationship between parameter count and system size. This is in contrast to previous approaches^{12,13} where the number of parameters grows explicitly with the number of particles, making it impossible to use a single ansatz across systems of different sizes. In particular, backflows and envelope exponents have typically been chosen as trainable parameters of shape $[N_{\text{orb}} \times N_{\text{det}}]$. In our ansatz the backflows $\mathbf{F}$ are instead computed by a single function $\mathbf{f}_{\theta}$ from multiple inputs $\mathbf{c}_{Ik}$.
- **Equivariant to sign of HF-orbital:** Orbitals of a HF-calculation are obtained as eigenvectors of a matrix and are thus determined only up to their sign (or their phase in the case of complex eigenvectors). We enforce that the functions $\mathbf{f}_{\theta}^o, g_{\theta}^e$ are odd and even with respect to $\mathbf{c}_{Ik}$. Therefore our orbitals Φ_{ik}^d are equivariant to a flip in the sign of the HF-orbitals used as inputs: $\Phi(-\mathbf{c}_{Ik}) = -\Phi(\mathbf{c}_{Ik})$. Therefore during supervised pre-training, the undetermined sign of the reference orbitals becomes irrelevant,

leading to faster convergence as demonstrated in “Equivariance with respect to HF-phase”.

- **Approximate locality:** When using localized HF-orbitals as input, the resulting TAOs are typically also localized. Localized HF-orbitals are orbitals which have non-zero orbital coefficients $\tilde{\alpha}_{Ik}$ only on some subset of atoms. Using our architecture outlined in “Obtaining orbital descriptors from Hartree-Fock”, this typically translates into local orbital features $\mathbf{c}_{Ik}$. Since we enforce the backflow $\mathbf{f}_{\theta}^o$ to be odd (and thus $\mathbf{f}_{\theta}^o(\mathbf{0}) = \mathbf{0}$), the resulting TAOs have zero contribution from atoms I with $\mathbf{c}_{Ik} = \mathbf{0}$. While the true wavefunction might not be fully decomposable into purely local contributions, many relevant chemical concepts –such as core electrons, bonds, lone pairs, or chemical groups– are intrinsically local concepts and locality has been successfully used as prior in many applications, for example in Neural Network Potentials^{27,28}. This hints at the prior of using local orbitals to compose many-body wavefunctions, which in turn can still contain non-local electron-electron correlations captured via the fully connected, non-local electron embedding $\mathbf{h}_i$.
- **High expressivity:** We empirically find that our ansatz is sufficiently expressive to model ground-state wavefunctions to high accuracy. We demonstrate this both empirically (c.f. SI 1, 1 mHa energy deviation against PsiFormer for NH_3) and theoretically (c.f. SI 2) in the supplementary information. This stands in contrast to previous approaches based on incorporating ab-initio orbitals¹, which could not reach chemical accuracy even for small molecules.

Size consistency of the ansatz

One design goal of the ansatz is to allow transfer of weights from small systems to larger systems. In particular, if a large system consists of many small previously seen fragments, one would hope to obtain an energy which corresponds approximately to the sum of the fragment energies. One simple test case, are chains of equally spaced Hydrogen atoms of increasing lengths. These systems have been studied extensively using high-accuracy methods²⁹, because they are small systems which already show strong correlation and are thus challenging to solve. We test our method by pre-training our ansatz on chains of length 6 and 10, and then evaluating the model (with and without subsequent fine-tuning) for chain lengths between 2 and 28. Figure 2a shows that our ansatz achieves very high zero-shot-accuracy in the interpolation regime ($N_{\text{atoms}} = 8$) and for extrapolation to slightly larger or slightly smaller chains ($N_{\text{atoms}} = 4, 12$). Even when extrapolating to systems of twice the size ($N_{\text{atom}} = 20$), our method still outperforms a Hartree-Fock calculation and eventually converges to an energy close to the Hartree-Fock solution.

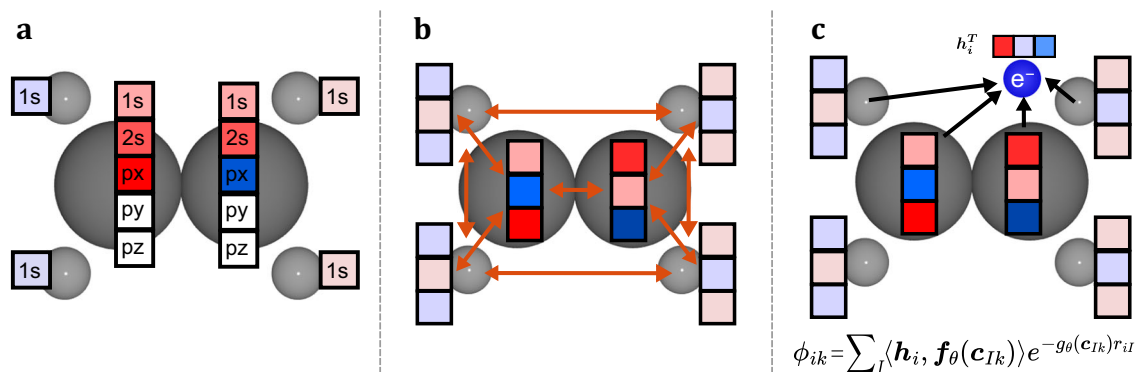


Fig. 1 | Illustration of the Transferable Atomic Orbitals, demonstrated on the C=C-bond of Ethene. **a** The input for each orbital are localized Hartree-Fock basis expansion coefficients $\tilde{\alpha}_I$, corresponding to every atom I . **b** We learn a representation $\mathbf{c}_I$ of the orbital on every atom using a Graph Neural Network, exchanging

information across atoms. **c** The orbital ϕ is evaluated for electron i by combining the electron-embedding $\mathbf{h}_i$ with the functions of the orbital representation $\mathbf{f}_{\theta}(\mathbf{c}_I)$ and $g_{\theta}(\mathbf{c}_I)$.

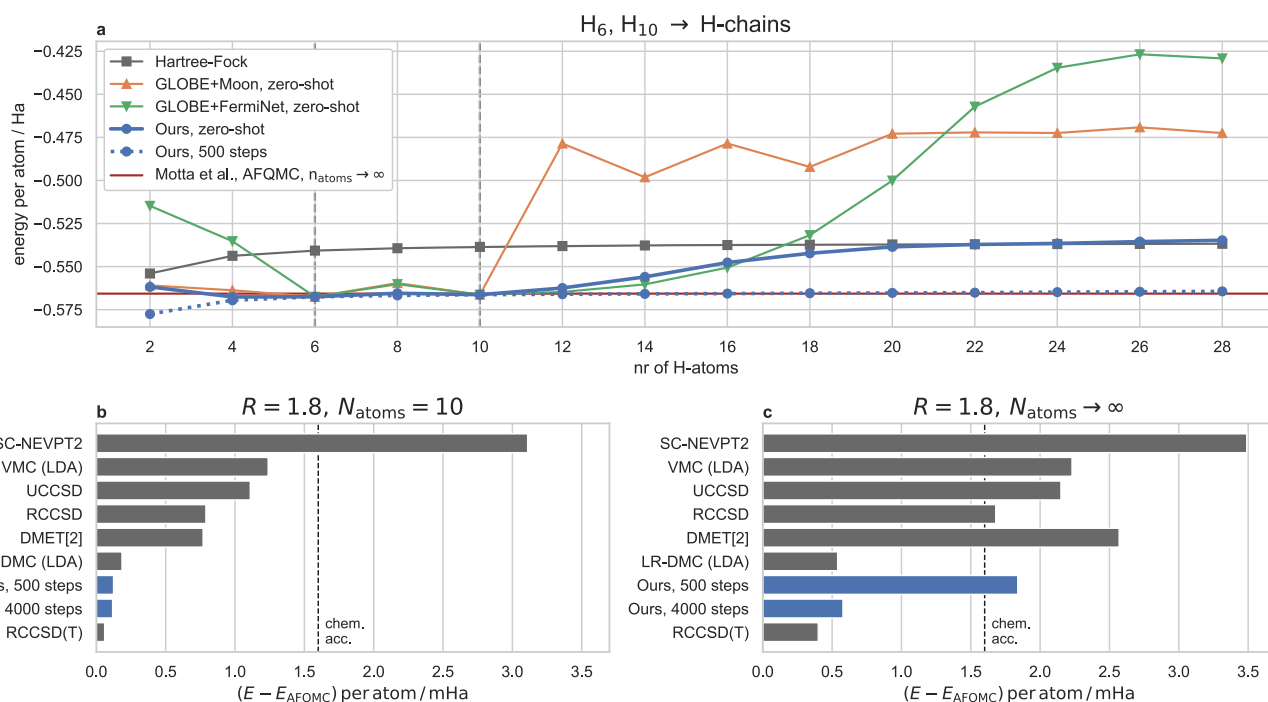


Fig. 2 | Transferability of the ansatz to chemically similar, larger systems, demonstrated on the example of hydrogen-chains. **a** Energy per atom as a function of chain-length. While GLOBE cannot successfully transfer to larger chains, our ansatz successfully predicts zero-shot energies (i.e. without fine-tuning) for up to 2x longer chains. **b, c** Comparison of our energies per atom after 500 and 4000 fine-tuning steps vs. high-accuracy methods from Motta et al.²⁹. The methods compared in this work include: SC-NEVPT2, the strongly contracted variant of the n_{el} electron valence state second-order perturbation theory; VMC (LDA), variational

Monte Carlo (local density approximation); UCCSD, couple cluster theory with full treatment of singles and doubles excitations; RCCSD and RCCSD(T), couple cluster theory with full treatment of singles and doubles and perturbative treatment of triple excitations using restricted Hartree-Fock as a reference state; DMET density-matrix embedding theory, LR-DMC (LDA) lattice-regularized diffusion Monte Carlo (local density approximation). **b** for the Hydrogen Chain with number of atoms $N_{\text{atoms}} = 10$ (**c**) for chain lengths extrapolated to the thermodynamic limit ($N_{\text{atoms}} \rightarrow \infty$).

To reach the accuracy of other correlated methods, we need a few fine-tuning steps for each new system. In Fig. 2b, c, we compare our results after 500 and 4000 fine-tuning steps against all high-accuracy methods from Motta et al.²⁹, which can obtain energies extrapolated to the thermodynamic limit (TDL). We compare for the two system sizes, investigated in ref. 29: 10 atoms in Fig. 2b, and extrapolation to $N_{\text{atoms}} = \infty$ in Fig. 2c. For H_{10} , our method is in near perfect agreement with their reference method AFQMC, deviating only by 0.1 mHa, nearly independent of the number of fine-tuning steps. This high-accuracy result is expected, since our model has also been pre-trained on chains of length 10 (albeit with different inter-atomic distances), and DL-VMC has previously been shown to achieve very high-accuracy on this system¹. When extrapolating to the TDL, our zero-shot energies are not competitive with high-accuracy methods, but instead yield energy errors comparable to Hartree-Fock, as seen in Fig. 2a. However, fine-tuning the ansatz for only 500 steps, yields energies that already outperform most methods studied in ref. 29 and fine-tuning for 4000 steps yields a deviation of 0.6 mHa/atom vs. AFQMC, on par with specialized methods such as LR-DMC.

This good performance stands in stark contrast to other approaches such as GLOBE+ FermiNet or GLOBE+Moon, studied in ref. 26: Both GLOBE-variants yield 5-6x higher errors in the interpolation regime and both converge to much higher energies for larger chains. While our approach yields Hartree-Fock-like energies for very long chains, GLOBE+FermiNet and GLOBE+Moon yield results that are outperformed even by assuming a chain of non-interacting H-atoms, which would yield an energy per atom of -0.5 Ha. For modest extrapolations ($N_{\text{atoms}} = 12$ to $N_{\text{atoms}} = 20$) our zero-shot results yield 3–20x lower errors than GLOBE+Moon.

Equivariance with respect to HF-phase

Due to using even and odd functions for the TAOs, our orbitals are equivariant with respect to a change of sign of the Hartree Fock orbitals. Therefore, a sign change of the HF-orbitals during HF-pre-training has no effect on the optimization of the wavefunction. One test case to assure this behavior is the rotation of a H_2O molecule, where we consider a set of 20 rotations of the same geometry, leading to a change of sign in the p-orbitals of the Oxygen atom (cf. Fig. 3). We evaluate our proposed architecture and compare it against a naïve approach, where we use a standard backflow matrix F , instead of a trainable, odd function f_0^o . In Fig. 3 we can see a clear spike in the HF-pre-training loss at the position of the sign flip for the standard backflow-type architecture, causing slower convergence during the subsequent variational optimization. After 16k optimization steps the effect diminishes and no substantial improvement on the accuracy can be observed. Although in this specific instance the orbital sign problem could also be overcome without our approach by correcting the phase of each orbital to align them across geometries, phase alignment is not possible in all circumstances. For example, there are geometry trajectories, where the Berry phase prevents such solutions³⁰.

Transfer to larger, chemically similar compounds

To test the generalization and transferability of our approach, we perform the following experiment: First, we train our ansatz on a dataset of multiple geometries of a single, small compound (e.g. 20 distorted geometries of Methane). For this training, we follow the usual procedure of supervised HF-pre-training and subsequent variational optimization as outlined in the methods section “Variational Monte Carlo”. After 64k variational optimization steps, we

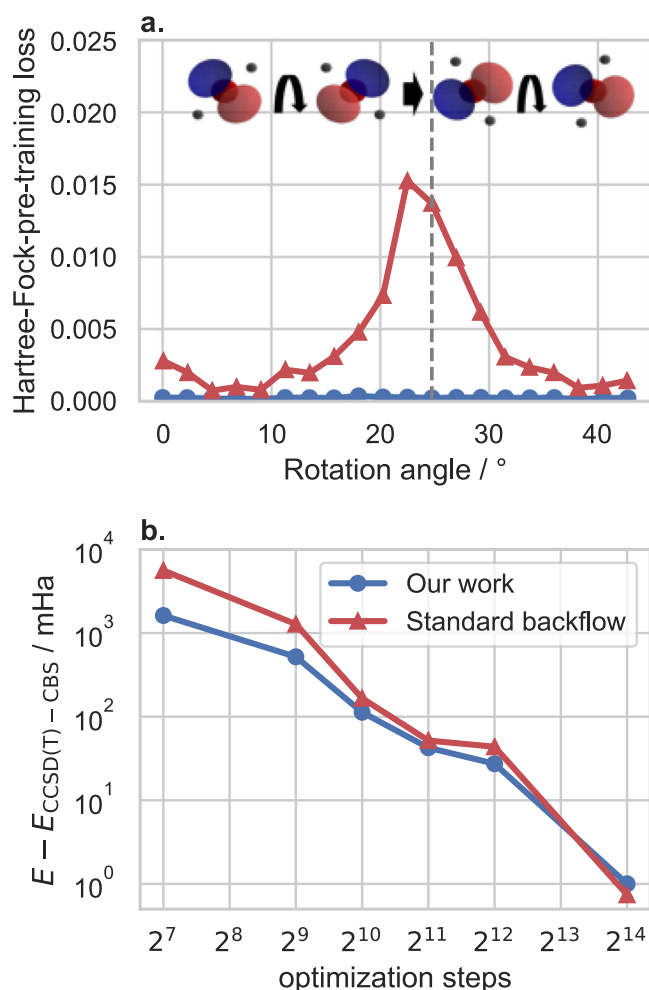


Fig. 3 | Accuracy when Hartree-Fock-pre-training against rotated H₂O molecules, which contain a change of sign in the Hartree-Fock-p-orbitals of the Oxygen atom (a). Comparing a shared optimization of a backflow-based neural network wavefunction (Standard backflow) against transferable atomic orbitals (Our work). **a** Hartree-Fock-pre-training loss of the last 100 Monte Carlo samples for 20 rotated geometries **(b):** Mean energy error vs. couple cluster reference calculations ($E_{\text{CCSD(T)}}$), averaged across all geometries.

then re-use the weights for different geometries of a larger compound (e.g. distorted geometries of Ethene). We fine-tune the model on this new geometry dataset for a variable number of steps and plot the resulting energy errors in Fig. 4. We do not require supervised HF-pre-training on the new, larger dataset. We perform this experiment for 3 pairs of test systems: Transferring from geometries of Hydrogen-chains with 6 atoms each, to chains with 10 atoms each, transferring from Methane to Ethene, and transferring from Ethene to Cyclobutadiene. These test systems are of interest, because they show strong correlation, despite being relatively small and computationally cheap systems. For example, even CCSD(T) overestimates the energy barriers of the Ethene- and Cyclobutadiene-PES by ≈ 10 mHa^{13,31}.

We compare our results to the earlier DeepErwin approach¹³, which only partially reused weights, and GLOBE, a concurrent work²⁶ which reuses all weights. To measure accuracy we compare two important metrics: First, the mean energy error (averaged across all geometries g of the test dataset) $\frac{1}{N} \sum_g (E_g - E_g^{\text{ref}})$, which reflects the method's accuracy for absolute energies (cf. Fig. 4a). Second, the deviation of the relative energy between the highest and lowest point of the PES, i.e. $\Delta E - \Delta E_{\text{ref}} = (E_{\text{max}} - E_{\text{min}}) - (E_{\text{max}}^{\text{ref}} - E_{\text{min}}^{\text{ref}})$, plotted in Fig. 4b. Since different studies use different batch-sizes and different

definitions of an epoch, we plot all results against the number of samples used for the energy estimation during variational optimization, which is very closely linked to computational cost.

Compared to other approaches, we find that our method yields substantially lower and more consistent energies. On the toy problem of H₆ to H₁₀ our approach and GLOBE reach the same accuracy, while DeepErwin converges to higher energies. For the real-world molecules Ethene (C₂H₄) and Cyclobutadiene (C₄H₄) our approach reaches substantially lower (and thus more accurate) energies and much more consistent potential energy surfaces. After 64 mio. fine-tuning samples, our mean absolute energies are 16 mHa and 17 mHa lower than GLOBE, and our relative energies are 39 mHa and 20 mHa closer to the reference calculation. When inspecting the resulting Potential Energy Surface for Ethene (Fig. 4c), we find that we obtain qualitatively similar results as DeepErwin and MRCI, but obtain energies that are ≈ 6 mHa lower (and thus more accurate). GLOBE on the other hand does not yield the correct PES for this electronically challenging problem, since it overestimates the energy barrier at 90° twist angle by ≈ 50 mHa. We observe similar results on the Cyclobutadiene geometries, where our approach yields relative energies that are in close agreement to the reference method, while the GLOBE-results overestimate the energy difference by ≈ 20 mHa.

Towards a first foundation model for neural network wavefunctions

While the experiments in the previous section demonstrate the ability to pre-train our model and fine-tune it on a new system, the resulting pre-trained models are of little practical use, since they are only pre-trained on a single compound each and can thus not be expected to generalize to chemically different systems. To obtain a more diverse pre-training dataset, we compiled a dataset of 360 distorted geometries, spread across 18 different compounds. The dataset effectively enumerates all chemically plausible molecules with up to 18 electrons containing the elements H, C, N, and O. For details on the data generation see “Dataset used for pre-training of multi-compound model”. We pre-train a base-model for 500,000 steps on this diverse dataset and subsequently evaluate its performance, when computing Potential Energy Surfaces. We evaluate its error against CCSD(T) (extrapolated to the complete basis set limit) both for compounds that were in the pre-training dataset (with different geometries), as well as for new, larger, out-of-distribution compounds which were not present in the pre-training dataset. We compare the results against a baseline model, which uses the same architecture, but is trained from scratch. Instead of re-using the pre-trained weights, this baseline initializes its weights using the default method of supervised HF-pre-training for each specific molecule².

Figure 5 shows that fine-tuning our pre-trained model yields substantially lower energies than the usual optimization from a HF-pre-trained model. For example, for new large compounds, it only takes 1k fine-tuning steps of the pre-trained model, to reach the same accuracy as CCSD(T) with a 3Z basis set. The non-pretrained model has a 60x higher energy error after 1k optimization steps, and requires 20x more steps to reach this accuracy. As expected, the gains from pre-training diminish for long subsequent optimization, but after 32k optimization steps, the pre-trained model still demonstrates 3x lower energy errors than the model being trained from scratch.

To assess the accuracy of our method for relative energies, we use the pre-trained model to compute a potential energy surface of a carbon dimer. Figure 6 compares our energies (with and without fine-tuning) against other state-of-the-art conventional and deep-learning-based methods. We compare against CCSD(T) extrapolated to the complete basis-set limit, an FCI-QMC study of

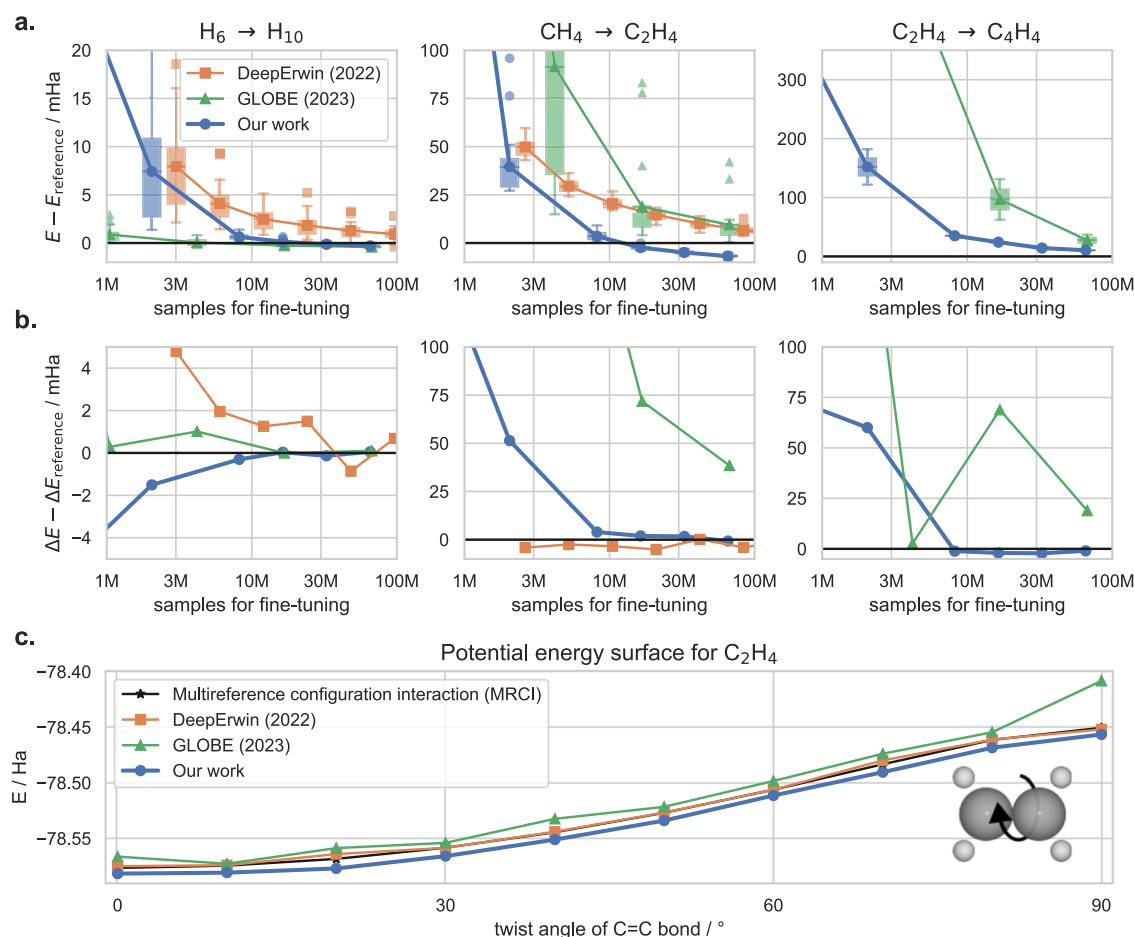


Fig. 4 | Accuracy when pre-training the model on small compounds and reusing it for larger compounds. Boxplots show the 25-75th percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 interquartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. **a** Mean energy vs reference energy

E_{ref} , averaged across all geometries of the test set. **b** Deviation of relative energies. **c** Final Potential Energy Surface (PES) for the Ethene molecule for each method. For the H-chains and C_2H_4 we use MRCI results from¹³ as reference energy, for Cyclobutadiene we use FermiNet results from¹⁶ as reference.

the C_2 -dimer by Booth et al.³² and PsiFormer⁴, the currently most accurate deep-learning based ansatz for absolute energies. We find that our approach without any fine-tuning steps correctly identifies the energy minimum at $d = 2.35$ bohr and even yields equilibrium energies that are lower than FCIQMC (cf. Fig. 6a). While the carbon dimer itself is not part of the pre-training dataset, several molecules with C=C bonds are, which explains the relatively high accuracy in this regime. When stretching the bond, our zero-shot energies overestimate the resulting energy by roughly 250 mHa (cf. Fig. 6b), clearly highlighting this failure case in a regime of lacking pre-training data. However, after just 1k fine-tuning steps of our pre-trained base-model, we obtain the qualitatively correct PES. In particular in the electronically most challenging regime around $d = 3$ bohr, FCIQMC and CCSD(T) both systematically overestimate the relative energy by 20-30 mHa compared to PsiFormer, whereas our method only overestimates the energy by ca. 10 mHa (cf. Fig. 6c). We note that our approach uses only 1k optimization steps per geometry, compared to 100k (and twice the batch size) for PsiFormer, thus requiring GPU-hours instead of GPU-days.

Scaling behavior

In many domains, increasing the amount of pre-training, has led to substantially better results, even without qualitative changes to the architecture³³. To investigate the scalability of our approach, we vary the three key choices, along which one could increase the scale of pre-

training: The size of the wavefunction model, the number of compounds and geometries present in the pre-training-dataset, and the number of pre-training steps. Starting from a large model, trained on 18×20 geometries, for 256k pre-training steps, we independently vary each parameter. We test 3 different architectures sizes, with decreasing layer width and depth for the networks f_θ , g_θ , and GCN_θ (Computational settings). We test 3 different training sets, with decreasing number of compounds in the training set, with 20 geometries each (Dataset used for pre-training of foundationmodel). Finally, we evaluate model-checkpoints at different amounts of pre-training, ranging from 64k steps to 512k steps. Figure 7 depicts the accuracy obtained by subsequently fine-tuning the resulting model for just 4000 steps on the evaluation set. In each case, increasing the scale of pre-training clearly improves evaluation results—both for the small in-distribution compounds, as well as the larger out-of-distribution compounds. We find a strong dependence of the accuracy on the model size and number of compounds in the pre-training dataset, and a weaker dependency on the number of pre-training steps. While our computational resources, currently prohibit us from training at larger scale, the results indicate that our approach may already be sufficient to train an accurate multi-compound, multi-geometry foundation model for wavefunctions.

Discussion

This work presents an ansatz for deep-learning-based VMC, which can in principle be applied to molecules of arbitrary size. We

demonstrate the favorable properties of our ansatz, such as extensivity, zero-shot prediction of wavefunctions for similar molecules (Size consistency of the ansatz), invariance to the phase of orbitals (Equivariance with respect to HF-phase) and fast fine-tuning for larger, new molecules (Transfer to larger, chemically similar compounds). Most importantly, “Towards a first foundation model for neural network wavefunctions” is, to our knowledge, the first successful demonstration of a wavefunction, which is transferable across compounds and has successfully been trained on a diverse dataset of compounds and geometries. We demonstrate that the dominating deep-learning paradigm of the last years—pre-

training on large data and fine-tuning on specific problems—can also be applied to the difficult problem of wavefunctions. While previous attempts^{13,26} have failed to obtain high-accuracy energies from pre-trained neural network wavefunctions, we find that our approach yields accurate energies and does so at a fraction of the cost needed without pre-training. A typical inference run (batch-size 2048, one compute node with 2 GPUs) for a molecule with 3 heavy atoms takes ~30 min for zero-shot evaluation, or 1.5h for 1k fine-tuning steps and subsequent evaluation. A CCSD(T)-4Z calculation on a compute-node with 128 CPUs took ~30 min. Given that the per-iteration cost of DL-VMC scales as $\mathcal{O}(n_{\text{el}}^4)$ vs. the $\mathcal{O}(n_{\text{el}}^7)$ cost of CCSD(T), we expect our model to become competitive once pre-trained and applied to sufficiently large molecules. We furthermore demonstrate in “Scaling behavior” that results can be improved systematically by scaling up any aspect of the pre-training: Model size, data-size, or pre-training-steps.

Despite these promising results, there are many open questions and limitations which should be addressed in future work. First, we find that our ansatz currently does not fully match the accuracy of state-of-the-art single-geometry DL-VMC ansätze. While our approach consistently outperforms conventional variational methods such as MRCI or CCSD at finite basis set, larger, computationally more expensive DL-VMC models can reach even lower energies. For example, PsiFormer optimized for 100k steps on the carbon dimer, reaches ~20 mHa lower absolute energies than our approach fine-tuned for 1k steps. Exchanging our message-passing-based electron-embedding, with recent attention based approaches⁴ should lead to higher accuracy. Furthermore we have made several deliberate design choices, which each trade-off expressivity (and thus potentially accuracy) for computational cost: We do not exchange information across orbitals and we base our orbitals on computationally cheap HF-calculations. Including attention or message passing across orbitals (e.g. similar to ref. 26), and substituting HF for a trainable, deep-learning-based model should further increase expressivity. While we currently use HF-orbitals due to their widespread use and low computational cost, our method does not rely on a specific orbital descriptor. We could substitute HF for a separate model such as PhisNet²⁰ or SchnOrb³⁴ to compute orbital descriptors $\mathbf{c}_{jk}$, leading to a fully end-to-end machine-learned wavefunction. Second, while we include useful physical priors such as locality, we do not yet currently use the invariance of the Hamiltonian with respect to rotations, inversions or spin-flip. E3-equivariant networks have been highly successful for neural network force-fields, but have not yet been applied to

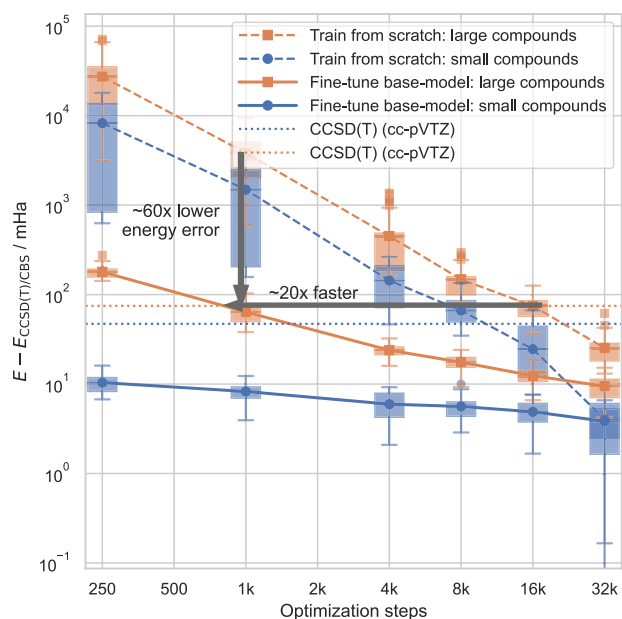


Fig. 5 | Fine-tuning of variationally pre-trained base-model (solid lines) vs. training a model from scratch (dashed lines) for 70 different geometries. Boxplots show the 25-75th percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 interquartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. Small compounds are in-distribution, with geometries similar to geometries in pre-training dataset. Larger compounds are out-of-distribution and are not present in the pre-training dataset.

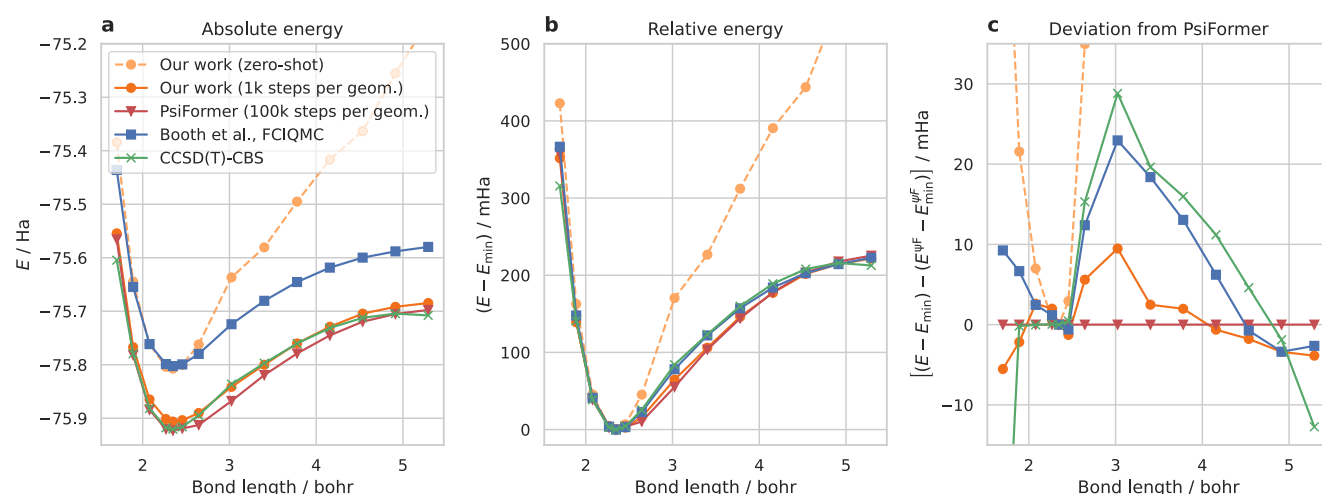


Fig. 6 | Potential energy surface of C_2 . **(a)** Absolute energies, **(b)** Energies of each method relative to the energy minimum at $d = 2.35$ bohr, **(c)** Deviation of relative energies from the relative energies obtained by PsiFormer (ψ^F).

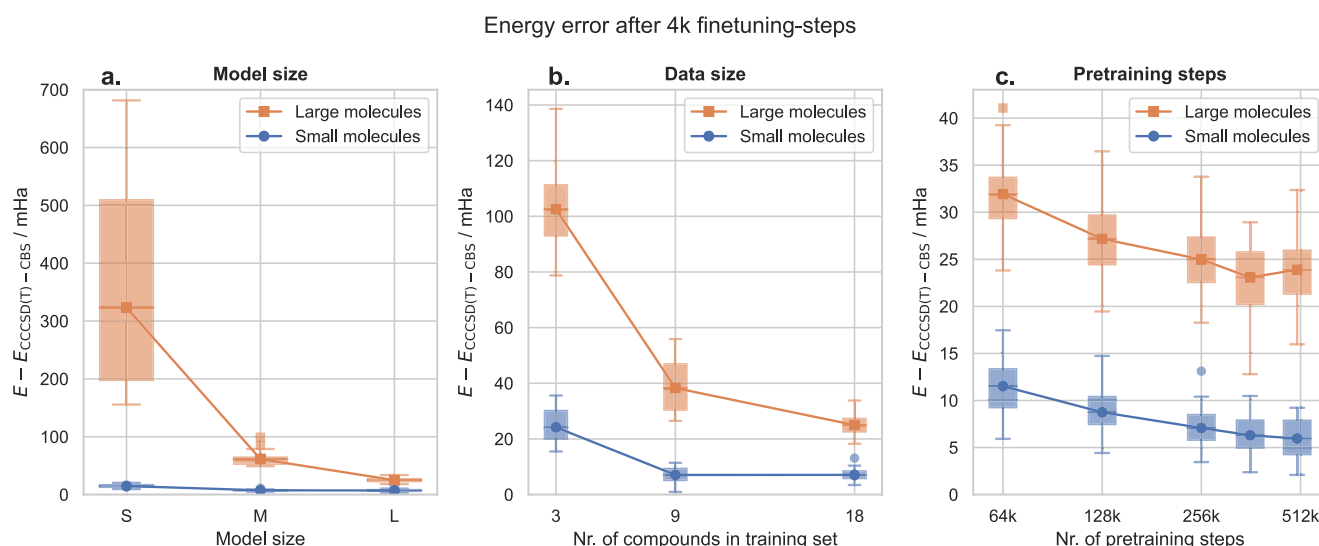


Fig. 7 | Error when fine-tuning the pre-trained model for 4000 steps on small in-distribution geometries and larger out-of-distribution geometries. Boxplots show the 25-75th percentile of energy deviations, connecting lines show mean energy deviations, whiskers span the non-outlier range (1.5 interquartile ranges above and below the boxes), energy deviations beyond the whiskers are plotted individually. **a** The energy error for increasing the model size of the transferable atomic orbitals. The small model uses no hidden layers and no graph convolutional

network. The medium-sized model uses one hidden layer of width 64 and 128 for g_θ and for f_θ respectively, and one iteration of the graph convolutional network. The large model uses two iterations of the graph convolutional network, f_θ and g_θ with hidden dimension 256 and 128. **b** The energy error when using a pre-trained model on a dataset with either 3, 9 or 18 compounds. **c** The energy error with increasing number of pretraining steps.

wavefunctions due to the hitherto unsolved problem of symmetry breaking¹⁵. Using HF-orbitals as symmetry breakers, could open a direct avenue towards E3-equivariant neural network wavefunctions. Third, while we use locality of our orbitals as a useful prior, we do not yet use it to reduce computational cost. By enforcing sparsity of the localized HF-coefficients, one could limit the evaluation of orbitals to a few participating atoms, instead of all atoms in the molecule. While the concurrent GLOBE approach enforces its orbitals to be localized at a single position, our approach naturally lends itself to force localization at a given number of atoms, allowing for a deliberate trade-off of accuracy vs. computational cost. Lastly, we observe that our method performs substantially better, when dedicating more computational resources to the pre-training, which makes it likely that future work will be able to scale up our approach. To facilitate this effort we open source our code, dataset as well as model parameters.

Methods

Variational Monte Carlo

Considering the Born-Oppenheimer approximation, a molecule with n_{el} electrons and N_{atoms} nuclei can be described by the time-independent Schrödinger equation

$$\hat{H}\psi = E\psi \quad (6)$$

with the Hamiltonian

$$\hat{H} = -\frac{1}{2} \sum_i \nabla_i^2 + \sum_{i,j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{I,J} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{i,I} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \quad (7)$$

By $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_\uparrow}, \dots, \mathbf{r}_{n_\uparrow+n_\downarrow}) \in \mathbb{R}^{3 \times n_{\text{el}}}$ we denote the set of electron positions divided into $n_\uparrow$ spin-up and $n_\downarrow = n_{\text{el}} - n_\uparrow$ spin-down electrons. The solution to the electronic Schrödinger equation ψ needs to fulfill the anti-symmetry property, i.e. $\psi(\mathcal{P}\mathbf{r}) = -\psi(\mathbf{r})$ for

any permutation $\mathcal{P}$ of two electrons of the same spin. Finding the groundstate wavefunction of a system corresponds to finding the solution to Eq. (6), with the lowest eigenvalue E_0 . Using the Rayleigh-Ritz principle, an approximate solution can be found through minimization of the loss

$$\mathcal{L}(\psi_\theta) = \mathbb{E}_{\mathbf{r} \sim \psi_\theta^2(\mathbf{r})} \left[\frac{(\hat{H}\psi_\theta)(\mathbf{r})}{\psi_\theta(\mathbf{r})} \right] \geq E_0, \quad (8)$$

using a parameterized trial wavefunction ψ_θ . The expectation value in Eq. (8) is computed by drawing samples $\mathbf{r}$ from the unnormalized probability distribution $\psi_\theta^2(\mathbf{r})$ using Markov Chain Monte Carlo (MCMC). The application of the Hamiltonian to the wavefunction can be computed using automatic differentiation and the loss is minimized using gradient based minimization. A full calculation typically consists of three steps:

- Supervised HF-pre-training:** Minimization of the difference between the neural network ansatz and a reference wavefunction (e.g. a Hartree-Fock calculation) $\|\psi_\theta - \psi^{\text{HF}}\|$. This is the only part of the procedure which requires reference data, and ensures that the initial wavefunction roughly resembles the true groundstate. While this step is in principle not required, it substantially improves the stability of the subsequent variational optimization.
- Variational optimization:** Minimization of the energy (Eq. (8)) by drawing samples from the wavefunction using MCMC, and optimizing the parameters θ of the ansatz using gradient based optimization.
- Evaluation:** Evaluation of the energy by evaluating Eq. (8) without updating the parameters θ , to obtain unbiased estimates of the energy.

To obtain a single wavefunction for a dataset of multiple geometries or compounds, only minimal changes are required. During supervised and variational optimization, for each gradient step we pick one geometry from the dataset. We pick geometries either in a

round-robin fashion, or based on the last computed energy variance for that geometry. We run the Metropolis Hastings algorithm³⁵ for that geometry to draw electron positions $\mathbf{r}$ and then evaluate energies and gradients. For each geometry we keep a distinct set of electron samples $\mathbf{r}$.

Obtaining orbital descriptors from Hartree-Fock

As discussed in “A multi-compound wavefunction ansatz”, our ansatz effectively maps uncorrelated, low-accuracy Hartree-Fock orbitals, to correlated, high-accuracy neural network orbitals. The first step in this approach is to obtain orbital descriptors $\mathbf{c}_k$ for each orbital k , based on a Hartree-Fock calculation.

The Hartree-Fock method uses a single determinant as ansatz, composed of single-particle orbitals ϕ_k^{HF} :

$$\psi^{\text{HF}}(\mathbf{r}_1, \dots, \mathbf{r}_{n_{\text{el}}}) = \det \left[\Phi_{ik}^{\text{HF}} \right]_{i,k=1 \dots n_{\text{el}}} \quad (9)$$

$$\Phi_{ik}^{\text{HF}} := \phi_k^{\text{HF}}(\mathbf{r}_i) \quad (10)$$

For molecules, these orbitals are typically expanded in atom-centered basis-functions $\mu(\mathbf{r})$, with N_{basis} functions centered on each atom l :

$$\phi_k^{\text{HF}}(\mathbf{r}) = \sum_{l=1}^{N_{\text{atoms}}} \sum_{b=1}^{N_{\text{basis}}} \alpha_{lk,b} \mu_b(\mathbf{r} - \mathbf{R}_l), \quad (11)$$

The coefficients $\alpha_{lk} \in \mathbb{R}^{N_{\text{basis}}}$ and the corresponding orbitals $\phi_k^{\text{HF}}(\mathbf{r})$ are obtained as solutions of an eigenvalue problem and are typically delocalized, i.e. they have non-zero contributions from many atoms. However, since $\det[U\Phi] = \det[U]\det[\Phi]$, the wavefunction is invariant under linear combination of orbitals by a matrix U with $\det[U]=1$. One can thus choose localized orbital expansion coefficients

$$\tilde{\alpha}_{lk,b} = \sum_{k'=1}^{N_{\text{orb}}} \alpha_{lk,b} U_{kk'} \quad (12)$$

corresponding to orbitals which are maximally localized according to some metric. We stress that such a transformation from canonical to localized orbitals is lossless: The localized orbitals represent exactly the same wavefunction as the canonical orbitals and thus the procedure involves no approximation. We localize orbitals purely to simplify the learning problem for the subsequent trainable functions $\mathbf{f}$, and $\mathbf{g}$, which map orbital descriptors to backflows and exponents. If the orbitals for typical molecules can be composed of recurring local motives (which empirically holds true), this substantially simplifies the generalization of $\mathbf{f}$ and $\mathbf{g}$ to larger molecules, since their inputs will mostly consist of orbital coefficients already seen in smaller molecules. Several different metrics and corresponding localization schemes, such as Foster-Boys³⁶ or Pipek-Mezey³⁷, have been proposed to find the optimal transformation matrix U and are easily available as computationally cheap post-processing options in quantum chemistry codes. We use the Foster-Boys method as implemented in pySCF³⁸.

Due to the fundamentally local nature of atom-wise orbital coefficients $\tilde{\alpha}_{lk}$, which can be insufficient to distinguish orbitals, we use a fully connected graph convolutional neural network (GCN) to add context about the surrounding atoms. We interpret each atom as a node (with node features $\tilde{\alpha}_{lk}$) and use the set of all 3D inter-atomic distance vectors $\{\mathbf{R}_{JJ'}\}$ as edge features:

$$\mathbf{c}_{lk} = \text{GCN}_{\theta,l} \left(\left\{ \tilde{\alpha}_{Jk} \right\}_{J=1 \dots N_{\text{atoms}}}, \left\{ \mathbf{R}_{JJ'} \right\} \right), \\ J, J' = 1 \dots N_{\text{atoms}}$$

We embed the edge features using a Kronecker product of Gaussian basis functions (of means $\boldsymbol{\mu} \in \mathbb{R}^{D_{\text{edge}}}$ and widths $\boldsymbol{\sigma} \in \mathbb{R}^{D_{\text{edge}}}$) of the inter-atomic distance $R_{JJ'}$ and the concatenation of the 3D-distance vector with the constant 1. The embedded edge features are then mapped to a high-dimensional feature space with a multi-layer perceptron (MLP):

$$\tilde{\mathbf{e}}_{JJ'} = \exp \left(-\frac{(\mathbf{R}_{JJ'} - \boldsymbol{\mu})^2}{2\boldsymbol{\sigma}^2} \right) \otimes [1|\mathbf{R}_{JJ'}] \quad (13)$$

$$\mathbf{e}_{JJ'} = \text{MLP}(\tilde{\mathbf{e}}_{JJ'}) \quad (14)$$

$$\mathbf{c}_{lk}^0 = \tilde{\alpha}_{lk} \\ \tilde{\mathbf{e}}_{JJ'} \in \mathbb{R}^{4D_{\text{edge}}}, \quad \mathbf{c}_{lk}^0 \in \mathbb{R}^{N_{\text{basis}}} \quad (15)$$

Each layer l of the GCN consist of the following update rules

$$\mathbf{u}_{lk}^l = \sum_j \mathbf{c}_{jk}^l \odot (\mathbf{W}_e^l \mathbf{e}_{lj}), \quad (16)$$

$$\mathbf{c}_{lk}^{l+1} = \sigma(\mathbf{W}_c^l \mathbf{c}_{lk}^l + \mathbf{W}_u^l \mathbf{u}_{lk}^l), \quad (17)$$

with trainable weight matrices $\mathbf{W}_e^l$, $\mathbf{W}_c^l$, $\mathbf{W}_u^l$ and the SiLU activation function σ ³⁹. After L iterations we use the final outputs as orbital features:

$$\mathbf{c}_{lk} := \mathbf{c}_{lk}^L \quad (18)$$

Mapping orbital descriptors to wavefunctions

To obtain entries Φ_{ik} of the Slater determinant, we combine a high-dimensional electron embedding $\mathbf{h}_i$ with a function of the orbital descriptor $\mathbf{c}_{lk}$:

$$\mathbf{h}_i = \mathbf{h}_{\theta}(\mathbf{r}_i, \{\mathbf{r}_{\uparrow}\}, \{\mathbf{r}_{\downarrow}\}, \{(\mathbf{R}, \mathbf{Z})\}) \quad (19)$$

$$\varphi_{\theta}^d(\mathbf{r}_i, \mathbf{R}_l, \mathbf{c}_{lk}) = \exp \left(-|\mathbf{r}_i - \mathbf{R}_l| g_{\theta}^{e,d}(\mathbf{c}_{lk}) \right) \quad (20)$$

$$\Phi_{ik}^d = \sum_{l=1}^{N_{\text{atoms}}} \varphi_{\theta}^d(\mathbf{r}_i, \mathbf{R}_l, \mathbf{c}_{lk}) \left\langle \mathbf{f}_{\theta}^{o,d}(\mathbf{c}_{lk}), \mathbf{h}_i \right\rangle \quad (21)$$

The functions GCN_{θ}^o , $\mathbf{f}_{\theta}^o$, and g_{θ}^e are trainable functions, which are enforced to be odd and even with respect to change in sign of their argument $\mathbf{c}$:

$$\text{Even } g_{\theta}^e : \\ g_{\theta}^e(\mathbf{c}) := g_{\theta}(\mathbf{c}) + g_{\theta}(-\mathbf{c}) \quad (22)$$

$$\text{Odd } \mathbf{f}_{\theta}^o : \\ \mathbf{f}_{\theta}^o(\mathbf{c}) := \mathbf{f}_{\theta}(\mathbf{c}) - \mathbf{f}_{\theta}(-\mathbf{c}) \quad (23)$$

$$\text{Odd } \text{GCN}_{\theta}^o : \\ \text{GCN}_{\theta}^o(\boldsymbol{\alpha}, \mathbf{R}) := \text{GCN}_{\theta}(\boldsymbol{\alpha}, \mathbf{R}) - \text{GCN}_{\theta}(-\boldsymbol{\alpha}, \mathbf{R}) \quad (24)$$

To obtain electron embeddings $\mathbf{h}_i$ we use the message-passing architecture outlined in⁵, which is invariant with respect to permutation of electrons of the same spin, or the permutation of ions. Note that during training, all samples in a batch come from the same geometry, and thus have the same values for $\mathbf{R}$, $\mathbf{Z}$, and $\tilde{\alpha}$. While the embedding

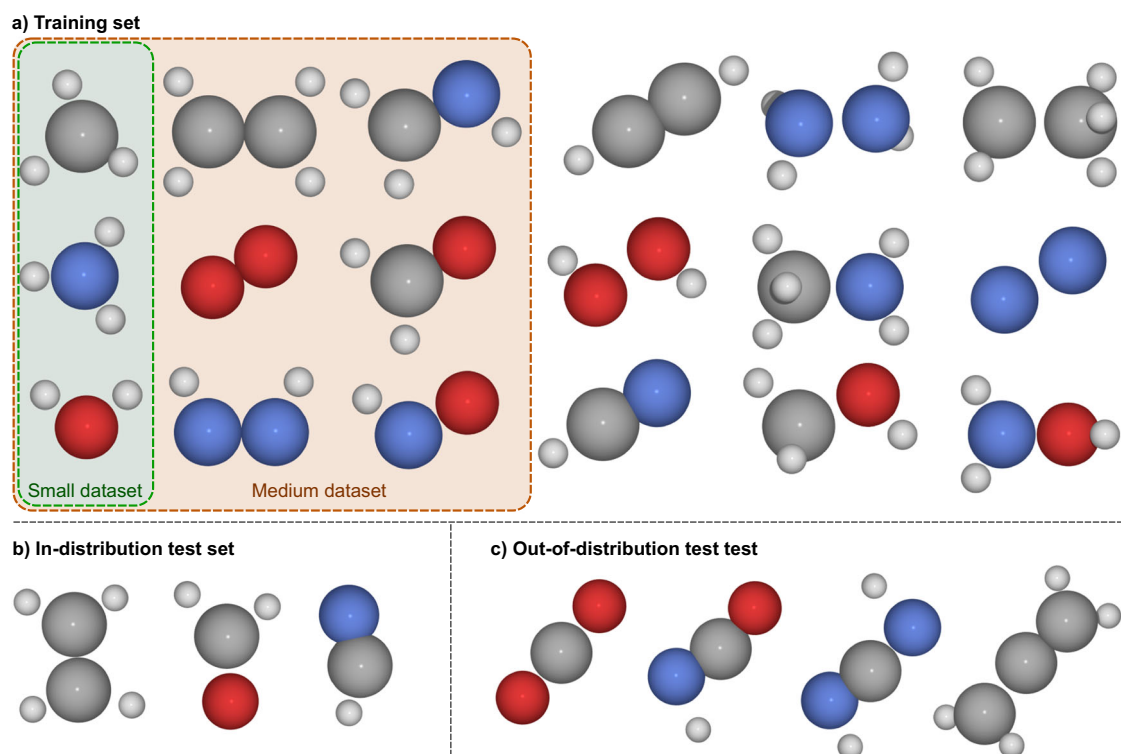


Fig. 8 | Compounds used for pre-training and evaluation of our model. Atom colors follow the usual convention of H = white, C = gray, N = blue, O = red. **a** The full training set, containing 18 compounds, each with 20 randomly distorted geometries. The small and medium sized training sets are subsets of this full training set,

containing 3 and 9 compounds respectively. **b** The in-distribution test set consists of 3 compounds with 10 distorted geometries each. **c** The out-of-distribution test set consists of 4 compounds with 10 distortions each.

network $\mathbf{h}_\theta^{\text{embed}}$, needs to be re-evaluated for every sample, the networks $\mathbf{GCN}_\theta$, $\mathbf{f}_\theta$, and $\mathbf{g}_\theta$ only need to be evaluated once per batch, substantially reducing their impact on computational cost.

Dataset used for pre-training of multi-compound model

We use RDKit⁴⁰ to generate all valid SMILES of molecules containing 1–3 atoms of the elements C, N, O. For each bond between atoms we allow single, double, and triple bonds. After saturating the molecules with Hydrogen, we perform force-field based geometry relaxation using RDKit. We obtain 18 compounds with 10–18 electrons, which we use for pre-training (cf. Fig. 8) and 35 compounds with 20–24 electrons, of which we use some for evaluation. Contrary to other datasets of small molecules such as GDB-7, our dataset also includes compounds which do not contain Carbon, such as the nitrogen dimer N_2 or hydrogen peroxide H_2O_2 . To obtain a more diverse dataset we perturb each equilibrium geometry by applying Gaussian noise to the 3D coordinates. Since this can generate nonphysical geometries, we keep only geometries in which the perturbed inter-atomic distances are between 90–140% of the unperturbed distances.

Reference energies

Reference energies for H_2O in Fig. 3 were computed using DL-VMC for 100,000 steps⁵. Reference energies for H_{10} and C_2H_4 in Fig. 4 were computed using MRCI-F12(Q)¹³. Reference energies for C_4H_4 in Fig. 4 were computed using DL-VMC¹⁶. To compute reference energies for our multi-compound dataset used in Fig. 5 and Fig. 7, we used pySCF³⁸ to perform CCSD(T) calculations using the cc-pCVXZ basis sets. We computed Hartree-Fock energies E_X^{HF} using basis-sets of valence $X = \{2, 3, 4\}$ and CCSD(T) energies $E_X^{\text{CCSD(T)}}$ using valence $X = \{2, 3\}$. To extrapolate to the complete-basis-set-

limit, we followed² and fitted the following functions with free parameters $E_{\text{CBS}}^{\text{HF}}, E_{\text{CBS}}^{\text{corr}}, a, b, c$:

$$\begin{aligned} E_X^{\text{HF}} &= E_{\text{CBS}}^{\text{HF}} + ae^{-bx} \\ E_X^{\text{corr}} &= E_X^{\text{HF}} - E_X^{\text{CCSD(T)}} = E_{\text{CBS}}^{\text{corr}} + cX^{-3} \\ E_{\text{CBS}}^{\text{CCSD(T)}} &= E_{\text{CBS}}^{\text{HF}} + E_{\text{CBS}}^{\text{corr}} \end{aligned}$$

We note that neither CCSD itself, nor the perturbative (T) treatment, nor the CBS extrapolation are variational methods. The computed reference energies are therefore not variational and may underestimate the true groundstate energy.

Computational settings

For a more detailed summary and explanation of the high-dimensional embedding structure we refer to the original work⁵. In all experiments we relied on the second order optimizer K-FAC^{41,42}. Key hyperparameters used in this work are summarized in Table 1. For the base model in “Towards a first foundation model for neural network wavefunctions” we increased the initial damping by 10x and ramped it down to 1×10^{-3} with an inverse scheduler. All runs reusing pre-trained weights, offset the learning rate scheduler by $o = 32,000$ steps, i.e. $\text{lr}(t) = \text{lr}_0(1 + (t + o)/6000)^{-1}$. This leads to a 5x lower initial learning rate. All pre-training runs in “Transfer to larger, chemically similar compounds” used 64,000 optimization steps. The base model in “Towards a first foundation model for neural network wavefunctions” used 512,000 optimization steps due to the larger and more diverse training corpus.

The small- and medium-sized model for our ablation study in Fig. 7 differ from the large model by the number of hidden layers for $\mathbf{f}_\theta$ and $\mathbf{g}_\theta$, the number of neurons per layer, and the number of iterations of the $\mathbf{GCN}_\theta$. The small model uses no hidden layers and

Table 1 | Hyperparameter settings used in this work

HF-pre-training	Pre-training basis set	6-31G + p-functions for H
	Pre-training steps per geometry	100-500
Embedding	Hidden dimension of $\mathbf{h}_i$	256
	Dimension of SchNet convolution	32
	N° iterations embedding	4
	Activation function	tanh
Transferable atomic orbitals	N° determinants N_{det}	4
	Basis set	6-31G + p-functions for H
	N° hidden layers $\mathbf{f}_{\theta}$	2
	Hidden dimension of $\mathbf{f}_{\theta}$	256
	N° hidden layers $\mathbf{g}_{\theta}$	2
	N° hidden dimension $\mathbf{g}_{\theta}$	128
	N° iterations GCN	2
	N° Gaussian basis functions	16
	Hidden edge embedding dimension D_{edge}	32
	Hidden node embedding dim.	16
	Activation function	SiLU
Markov Chain Monte Carlo	N° walkers	2048
	N° decorrelation steps	20
	Target acceptance prob.	50%
	Optimizer	KFAC
Variational pre-training	Damping	1×10^{-3}
	Norm constraint	3×10^{-3}
	Batch size	2048
	Initial learning rate lr_0	0.1
	Learning rate decay	$\text{lr}(t) = \text{lr}_0(1+t/6000)^{-1}$
	Optimization steps	64,000–512,000
Changes for fine-tuning	Learning rate decay	$\text{lr}(t) = \text{lr}_0(7+t/6000)^{-1}$
	Optimization steps	0–32,000

no graph convolutional network. The medium-sized model uses one hidden layer of width 64 for $\mathbf{g}_{\theta}$ and 128 for $\mathbf{f}_{\theta}$, and one iteration of the graph convolutional network. The small, medium and large models respectively have 0.8 mio, 1.2 mio. and 2.0 mio parameters.

Data availability

All geometry- and energy-data is available on GitHub under <https://github.com/mdsunivie/deeperwin>. Model weights are available on figshare under <https://doi.org/10.6084/m9.figshare.23585358.v1>⁴³. Source data are provided with this paper.

Code availability

All code is available on GitHub under <https://github.com/mdsunivie/deeperwin> and Zenodo (<https://doi.org/10.5281/zenodo.10081846>)⁴⁴.

References

- Hermann, J., Schätzle, Z. & Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.* **12**, 891–897 (2020).
- Pfau, D., Spencer, J. S., Matthews, A. G. D. G. & Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Res.* **2**, 033429 (2020).
- Spencer, J. S., Pfau, D., Botev, A. & Foulkes, W. M. C. Better, faster fermionic neural networks. *arXiv* <https://doi.org/10.48550/arXiv.2011.07125> (2020).
- von Glehn, I., Spencer, J. S. & Pfau, D. A self-attention ansatz for ab-initio quantum chemistry. In *The Eleventh International Conference on Learning Representations* 10853–10892 (ICLR, 2023).
- Gerard, L., Scherbela, M., Marquetand, P. & Grohs, P. Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need? In *Advances in Neural Information Processing Systems* 10282–10294 (NeurIPS, 2022).
- Towards the ground state of molecules via diffusion monte carlo on neural networks. *Nat. Commun.* **14**, 1860 (2023).
- Wilson, M., Gao, N., Wudarski, F., Rieffel, E. & Tubman, N. M. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo. *arXiv* <https://doi.org/10.48550/arXiv.2011.07125> (2021).
- Cassella, G. et al. Discovering quantum phase transitions with fermionic neural networks. *Phys. Rev. Lett.* **130**, 036401 (2023).
- Wilson, M. et al. Neural network ansatz for periodic wave functions and the homogeneous electron gas. *Phys. Rev. B* **107**, 235139 (2023).
- Li, X., Li, Z. & Chen, J. Ab initio calculation of real solids via neural network ansatz. *Nat. Commun.* **13**, 7895 (2022).
- Han, J., Zhang, L. & E, W. Solving many-electron Schrödinger equation using deep neural networks. *J. Comput. Phys.* **399**, 108929 (2019).
- Qian, Y., Fu, W., Ren, W. & Chen, J. Interatomic force from neural network based variational quantum Monte Carlo. *J. Chem. Phys.* **157**, 164104 (2022).
- Scherbela, M., Reisenhofer, R., Gerard, L., Marquetand, P. & Grohs, P. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nat. Comput. Sci.* **2**, 331–341 (2022).
- Entwistle, M. T., Schätzle, Z., Erdman, P. A., Hermann, J. & Noé, F. Electronic excited states in deep variational Monte Carlo. *Nat. Commun.* **14**, 274 (2023).
- Gao, N. & Günnemann, S. Ab-initio potential energy surfaces by pairing GNNs with neural wave functions. In *International Conference on Learning Representations* 10259–10281 (ICLR, 2022).
- Gao, N. & Günnemann, S. Sampling-free inference for ab-initio potential energy surface networks. In *The Eleventh International Conference on Learning Representations* 10896–10965 (ICLR, 2023).
- Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
- Kochkov, D. & Clark, B. K. Variational optimization in the AI era: computational graph states and supervised wave-function optimization. *arXiv* <https://doi.org/10.48550/arXiv.2011.07125> (2018).
- Schütt, K. T., Gastegger, M., Tkatchenko, A., Müller, K.-R. & Maurer, R. J. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nat. Commun.* **10**, 5024 (2019).
- Unke, O. et al. SE(3)-equivariant prediction of molecular wavefunctions and electronic densities. In *Advances in Neural Information Processing Systems* 14434–14447 (NeurIPS, 2021).
- Batatia, I. et al. The design space of E(3)-equivariant atom-centered interatomic potentials. *arXiv* <https://doi.org/10.48550/arXiv.2205.06643> (2022).
- Brown, T. et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*. (eds. Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H.) 1877–1901 (NeurIPS, 2020).

23. Radford, A. et al. Learning transferable visual models from natural language supervision. In Meila, M. & Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning Research* 8748–8763 (PMLR, 2021).
24. Yuan, L. et al. Florence: A new foundation model for computer vision. *arXiv* <https://doi.org/10.48550/arXiv.2011.07125> (2021).
25. Zhang, Y.-H. & Di Ventura, M. Transformer quantum state: a multi-purpose model for quantum many-body problems. *Phys. Rev. B* **107**, 075147 (2023).
26. Gao, N. & Günnemann, S. Generalizing neural wave functions. In Krause, A. et al. (eds.) *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research* 10708–10726 (PMLR, 2023).
27. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
28. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
29. Motta, M. et al. Towards the solution of the many-electron problem in real materials: equation of state of the hydrogen chain with state-of-the-art many-body methods. *Phys. Rev. X* **7**, 031059 (2017).
30. Westermayr, J. & Marquetand, P. Machine learning for electronically excited states of molecules. *Chem. Rev.* **121**, 9873–9926 (2021).
31. Lyakh, D. I., Musiał, M., Lotrich, V. F. & Bartlett, R. J. Multireference nature of chemistry: The coupled-cluster view. *Chemical Reviews* **112**, 182–243 (2012).
32. Booth, G. H., Cleland, D., Thom, A. J. W. & Alavi, A. Breaking the carbon dimer: The challenges of multiple bond dissociation with full configuration interaction quantum Monte Carlo methods. *J. Chem. Phys.* **135**, 084104 (2011).
33. Hoffmann, J. et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems* (eds. Koyejo, S. et al.) 30016–30030 (NeurIPS, 2022).
34. Gastegger, M., McSloy, A., Luya, M., Schütt, K. T. & Maurer, R. J. A deep neural network for molecular wave functions in quasi-atomic minimal basis representation. *J. Chem. Phys.* **153**, 044123 (2020).
35. Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).
36. Foster, J. M. & Boys, S. F. Canonical configurational interaction procedure. *Rev. Mod. Phys.* **32**, 300–302 (1960).
37. Pipek, J. & Mezey, P. G. A fast intrinsic localization procedure applicable for ab initio and semiempirical linear combination of atomic orbital wave functions. *J. Chem. Phys.* **90**, 4916–4926 (1989).
38. Sun, Q. et al. Recent developments in the PySCF program package. *J. Chem. Phys.* **153**, 024109 (2020).
39. Elfwing, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *arXiv* <https://doi.org/10.48550/arXiv.2011.07125> (2017).
40. Landrum, G. Rdkit: Open-Source Cheminformatics <https://github.com/rdkit/rdkit> (2009).
41. Martens, J. & Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2408–2417 (PMLR, 2015).
42. Botev, A. & Martens, J. KFAC-JAX <http://github.com/deepmind/kfac-jax> (2022).
43. Gerard, L., Scherbela, M. & Grohs, P. Pre-Trained Neural Wave-function Checkpoints for the GitHub Codebase DeepErwin https://figshare.com/articles/online_resource/Pre-trained_neural_wavefunction_checkpoints_for_the_GitHub_codebase_DeepErwin/23585358 (2023).
44. Scherbela, M., Gerard, L. & Grohs, P. DeepErwin <https://github.com/mdsunivie/deeperwin/blob/master/README.md> (2023).

Acknowledgements

We gratefully acknowledge financial support from the following grants: Austrian Science Fund FWF Project I 3403 (P.G.), WWTF-ICT19-041 (L.G.). The computational results have been achieved using the Vienna Scientific Cluster (VSC). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript. Additionally, we thank Nicholas Gao for providing his results and data, Ruud van Workum for initial work on the python implementation for multi-compound optimization and Jan Hermann for fruitful discussions.

Author contributions

M.S., L.G., and P.G. conceived the overall idea. M.S. conceived and implemented the ansatz, built the dataset and designed the experiments. L.G. gave input on the ansatz and worked on implementation. M.S. and L.G. performed the experiments. M.S. and L.G. wrote the manuscript with input, supervision and funding from P.G.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-023-44216-9>.

Correspondence and requests for materials should be addressed to Philipp Grohs.

Peer review information *Nature Communications* thanks Gero Friesecke and the other, anonymous, reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024

Supplementary Information Towards a Transferable Fermionic Neural Wavefunction for Molecules

Supplementary Note 1. Accuracy for single molecules

We compare in Fig. 1 our newly proposed orbital ansatz against other DL-VMC methods when optimizing a single geometry without initialization from a pre-trained base model. We find empirically that our method, using the built-in prior of local orbital coefficients, is able to reach chemical accuracy compared to a high-accurate PsiFormer calculation [1] with a final error of ~ 1 mHa. On the other hand PauliNet-like architectures, with ab-initio orbitals as initial prior, cannot reach this reference energy. We compare against two versions of PauliNet [2] implemented in [3]: A re-implementation of the original PauliNet (with minimally modified embedding) and an improved version using an additional backflow shift. When comparing to Gerard et al. [4], which uses the same electron embedding architecture as this work, we obtain a 0.7 mHa higher energy. This might be caused by the newly introduced orbitals, allowing to optimize across molecules but with a slight trade-off for single-molecule accuracy.

Supplementary Note 2. Expressiveness of our ansatz

In the following, we sketch out the theoretical expressiveness of our ansatz, relative to other DL-VMC ansätze. Conventional, non-transferable ansätze such as FermiNet or PsiFormer have the following structure

$$\phi_{ik}^{\text{FN}} = \prod_{I=1}^{N_{\text{nuc}}} \prod_{k=1}^{D_{\text{em}}} w_k h_i \exp(-\omega_k |\mathbf{r}_i|), \quad (1)$$

where w_k , h_i , $\omega_k \in \mathbb{R}$ are arbitrary trainable parameters. In comparison, our ansatz has the form

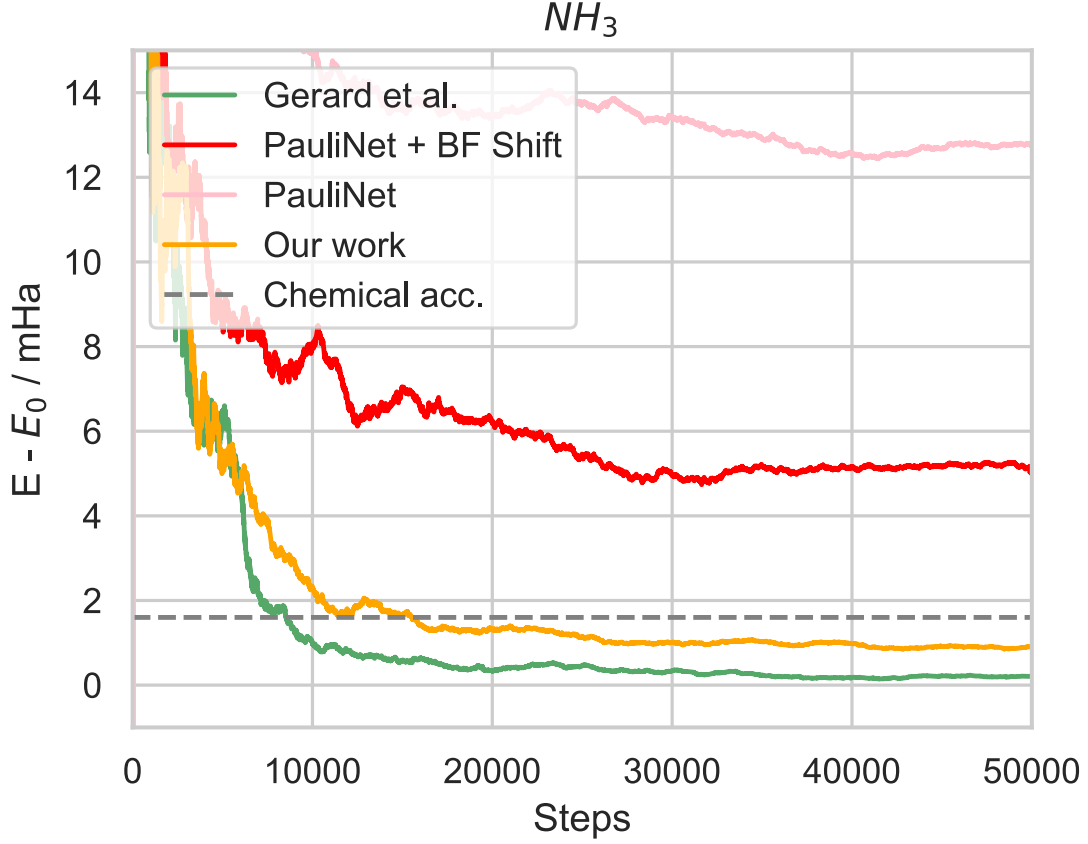
$$\phi_{ik}^{\text{TAO}} = \prod_{I=1}^{N_{\text{nuc}}} \prod_{k=1}^{D_{\text{em}}} f(\mathbf{c}_{Ik}) h_i \exp(-g(\mathbf{c}_{Ik}) |\mathbf{r}_i|), \quad (2)$$

leading to the following equivalence:

$$f(\mathbf{c}_{Ik}) = w_k \quad (3)$$

$$g(\mathbf{c}_{Ik}) = \omega_{Ik}. \quad (4)$$

Since neural networks are universal function approximators, this equivalence can be satisfied, for arbitrary choices of $\mathbf{w}$, $\boldsymbol{\omega}$, $\boldsymbol{\pi}$, as long as the inputs $\mathbf{c}_{Ik}$ are distinct for each index tuple (I, k) .



Supplementary Figure 1: Comparison of our work with HF-pre-training (i.e. no initialization from the base model) against other DL-VMC methods [2, 3, 4] for NH_3 . PsiFormer [1] is used as a baseline E_0 .

The orbital descriptors $\mathbf{c}_{Ik}$ are in turn functions of the localized Hartree-Fock coefficients $\tilde{c}_{Ik}$

$$\mathbf{c}_{Ik} = \text{GCN}_I(\{\tilde{c}_{Jk}\}_{J=1\dots N_{\text{nuc}}}, \{\mathbf{R}_{JJ'}\}_{J, J'=1\dots N_{\text{nuc}}}), \quad (5)$$

For single atoms, the Hartree-Fock orbital coefficients $\tilde{c}_k$ are always unique. In fact they are even orthogonal, since they are solutions to a generalized eigenvalue problem. Thus also $\mathbf{c}_k$ can be unique, and thus arbitrary backflows and exponents can be expressed. For a general molecule with no geometrical symmetry, unique orbital descriptors $\mathbf{c}_{Ik}$ can in principle also be achieved due to the expressiveness of the GCN. Since it can in principle learn any permutation equivariant function, it can output distinct coefficients $\mathbf{c}_{Ik}$ as long as all atoms are distinguishable either by geometry or by the orbital coefficients. This only leaves the case of symmetrical molecules, in which indistinguishable atoms exist, for example dimers. In this case non-transferable approaches such as FermiNet can express wavefunctions, which cannot be expressed by our ansatz. In particular,

these are wavefunctions which are not invariant to permutation of the nuclei. Since our ansatz is by design invariant to permutation of atoms (since the Hartree-Fock solution is equivariant to permutation and the sum over I ultimately leads to invariance), it cannot represent these wavefunctions. However, no physical observable can depend on the ordering of the atoms, this permutation invariance seems rather a desideratum than a limitation.

We note that the ability to express arbitrary antisymmetric wavefunctions is only a necessary, but not necessarily sufficient criterion for a good ansatz. Expressivity of the ansatz with a finite number of parameters, as well as stability during optimization can be of high importance in practice but are hard to assess from a purely theoretical point of view.

Supplementary references

- [1] Ingrid von Glehn, James S Spencer, and David Pfau. “A Self-Attention Ansatz for Ab-initio Quantum Chemistry”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [2] Jan Hermann, Zeno Schätzle, and Frank Noé. “Deep-Neural-Network Solution of the Electronic Schrödinger Equation”. In: *Nature Chemistry* 12.10 (Oct. 2020), pp. 891–897. ISSN: 1755-4349. OI: 10.1038/s41557-020-0544-y.
- [3] Michael Scherbela et al. “Solving the Electronic Schrödinger Equation for Multiple Nuclear Geometries with Weight-Sharing Deep Neural Networks”. In: *Nature Computational Science* 2.5 (May 2022), pp. 331–341. ISSN: 2662-8457. OI: 10.1038/s43588-022-00228-x.
- [4] Leon Gerard et al. “Gold-Standard Solutions to the Schrödinger Equation Using Deep Learning: How Much Physics Do We Need?” In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., Oct. 2022, pp. 10282–10294.

IV Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions

Comments

E-Print: `arXiv:2307.09337[physics.chem-ph]`

Conference: Poster presentation at NeurIPS 2023.

Authorship: L.G. and M.S. share first authorship.

Contribution

P.G., M.S. and L.G. conceived the project. M.S. implemented, adapted and trained the PhisNet-like orbital embedding model with input from L.G. L.G. developed the electron model with input from M.S. M.S. developed the normal-mode based sampling of nuclear geometries. L.G. extended the pre-training dataset. M.S. performed reference CCSD(T) calculations. L.G. and M.S. designed and performed the numerical experiments and wrote the manuscript with equal contribution. P.G. obtained funding and supervised the project.

Bibliographic Information

Scherbela, Michael, Leon Gerard, and Philipp Grohs (2023). “Variational Monte Carlo on a Budget - Fine-tuning pre-trained Neural Wavefunctions.” In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions

Michael Scherbela*

University of Vienna

michael.scherbela@univie.ac.at

Leon Gerard*

University of Vienna

leon.gerard@univie.ac.at

Philipp Grohs

University of Vienna

philipp.grohs@univie.ac.at

Abstract

Obtaining accurate solutions to the Schrödinger equation is the key challenge in computational quantum chemistry. Deep-learning-based Variational Monte Carlo (DL-VMC) has recently outperformed conventional approaches in terms of accuracy, but only at large computational cost. Whereas in many domains models are trained once and subsequently applied for inference, accurate DL-VMC so far requires a full optimization for every new problem instance, consuming thousands of GPUs even for small molecules. We instead propose a DL-VMC model which has been pre-trained using self-supervised wavefunction optimization on a large and chemically diverse set of molecules. Applying this model to new molecules without any optimization, yields wavefunctions and absolute energies that outperform established methods such as CCSD(T)-2Z. To obtain accurate relative energies, only few fine-tuning steps of this base model are required. We accomplish this with a fully end-to-end machine-learned model, consisting of an improved geometry embedding architecture and an existing SE(3)-equivariant model to represent molecular orbitals. Combining this architecture with continuous sampling of geometries, we improve zero-shot accuracy by two orders of magnitude compared to the state of the art. We extensively evaluate the accuracy, scalability and limitations of our base model on a wide variety of test systems.

1 Introduction

Solving the Schrödinger equation is of utmost importance for the prediction of quantum chemical properties in chemistry. The time-independent Schrödinger equation in the Born-Oppenheimer approximation [1] for a molecule with N_{nuc} nuclei and n_{el} electrons is an eigenvalue problem with Hamiltonian H :

$$H\psi = E\psi, \quad H = -\frac{1}{2} \sum_i \nabla_{\mathbf{r}_i}^2 + \sum_{i>j} \frac{1}{r_{ij}} + \sum_{I>J} \frac{Z_I Z_J}{R_{IJ}} - \sum_{i,I} \frac{Z_I}{r_{iI}}. \quad (1)$$

By $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_{N_{\text{nuc}}}) \in \mathbb{R}^{N_{\text{nuc}} \times 3}$ and $\mathbf{Z} = (Z_1, \dots, Z_{N_{\text{nuc}}}) \in \mathbb{N}^{N_{\text{nuc}}}$ we denote the nuclear positions and charges. The electron positions are denoted by $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_{\uparrow}}, \dots, \mathbf{r}_{n_{\uparrow}+n_{\downarrow}}) \in \mathbb{R}^{n_{\text{el}} \times 3}$ with $n_{\uparrow}$ spin-up electrons and $n_{\downarrow}$ spin-down electrons. The inter-particle difference and distance vectors are written as $\mathbf{R}_{IJ} = \mathbf{R}_I - \mathbf{R}_J$, $R_{IJ} = |\mathbf{R}_{IJ}|$, $\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$, $r_{iI} = |\mathbf{r}_{iI}|$, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $r_{ij} = |\mathbf{r}_{ij}|$ with $I, J = 1, \dots, N_{\text{nuc}}$ and $i, j = 1, \dots, n_{\text{el}}$. The eigenvalues E of Eq. 1 represent

*Equal contribution, author order random

the energy states of a molecule, whereas a special interest lies in finding the smallest eigenvalue E_0 , called the ground-state energy. The corresponding high-dimensional wavefunction $\psi : \mathbb{R}^{n_{\text{el}} \times 3} \rightarrow \mathbb{R}$ can be found via the Rayleigh-Ritz principle, by minimizing

$$\mathcal{L}(\psi_\theta) = \mathbb{E}_{\mathbf{r} \sim \psi_\theta^2(\mathbf{r})} \left[\frac{H\psi_\theta(\mathbf{r})}{\psi_\theta(\mathbf{r})} \right] \geq E_0. \quad (2)$$

Due to electrons being fermions, the solution must fulfill the anti-symmetry property, stating that the sign of the wavefunction must change for any permutation $\mathcal{P}$ of two electrons of the same spin: $\psi(\mathbf{r}) = -\psi(\mathcal{P}\mathbf{r})$. Having access to the solution ψ , allows in principle a complete description of the considered molecule. Unfortunately, only for one electron systems there exists an analytical solution and due to the curse of dimensionality, with increasing number of particles, obtaining an accurate approximation of the wavefunction becomes intractable already for medium-sized molecules. This is because many high-accuracy approximation methods scale poorly with n_{el} . For example CCSD(T) – coupled cluster with its single-, double-, and perturbative triple-excitations variant – is considered the gold-standard reference in computational chemistry, but its computational cost scales as $\mathcal{O}(n_{\text{el}}^7)$ [2]. Deep-learning-based Variational Monte Carlo (DL-VMC) has emerged as a promising alternative solution. A single step scales as $\mathcal{O}(n_{\text{el}}^4)$ and it has surpassed the accuracy of many conventional methods such as CCSD(T), when applied to small molecules [3]. In DL-VMC, the wavefunction is represented by a neural network with trainable parameters θ and optimized via Eq. 2 using gradient based optimization. Since the expectation value of Eq. 2 cannot be computed analytically, it is approximated by sampling the electron positions during optimization and evaluation with Monte Carlo methods like Metropolis-Hastings [4].

Related work FermiNet by Pfau et al. [3] and its variants have emerged as the leading architecture for DL-VMC in first quantization. It can reach highly accurate energies, but typically requires tens of thousands of optimization steps for convergence. Many improvements have been proposed to further increase accuracy [5–7] and accelerate convergence [8]. Furthermore, DL-VMC has been extended to properties beyond energies [9–11] and systems beyond molecules [12–16]. Despite the favorable scaling of DL-VMC, computational cost is still high, even for small molecules, often requiring thousands of GPUhs to find ψ for a single small molecule [17]. This is because unlike typical machine learning applications – which train an expensive model once, and subsequently achieve cheap inference – in DL-VMC the minimization of Eq. 2 is typically done from scratch for every new system.

A promising line of research to scale-up expensive ab-initio solvers such as DL-VMC or CCSD(T) has been to develop proxy methods, which can be trained on outputs of ab-initio methods to directly predict molecular properties [18, 19] or wavefunctions [20, 21] from the molecular geometry. While these proxy methods can often reproduce the underlying ab-initio method with high fidelity and scale to millions of atoms [22], they are fundamentally limited by the accuracy of their reference method and need many high-accuracy samples for training, reiterating the need for scaleable, high-accuracy reference methods.

To enable DL-VMC methods to efficiently compute wavefunctions for many molecules, several methods have been proposed to amortize the cost of optimization, by learning a single wavefunction across multiple systems. This has been demonstrated to work for different geometries of a single molecule [10, 23, 24], and recently two approaches have been proposed to learn wavefunctions across entirely different molecules, each with their own limitations. Gao et al. [17] reparameterized the orbitals of a wavefunction, by using chemistry-inspired heuristics to determine orbital positions, managing to efficiently generalize wavefunctions across different geometries of a single molecule. However, when learning a single wavefunction across different molecules, their results deteriorated and transfer to new molecules proved difficult. Scherbela et al. [25] do not require heuristic orbital positions, but instead use orbital descriptors of a low-accuracy conventional method to parameterize DL-VMC orbitals. However, while their wavefunction ansatz successfully transfers to new molecules, their method requires a separate, iterative Hartree-Fock (HF) calculation for every new geometry.

Our contribution This work presents the first end-to-end machine learning approach, which successfully learns a single wavefunction across many different molecules with high accuracy. Our contributions are:

- A transferable neural wavefunctions, which requires neither heuristic orbital positions, nor iterative HF calculations. We achieve this by building on the architecture by Scherbela et al. [25] and an orbital prediction model by Unke et al. [20].
- A simplified and improved electron embedding architecture, leveraging expressive nuclear features from our orbital prediction model and a message passing step between nuclei.
- A chemically diverse dataset with up to 100 molecules based on QM7-X [26], a data augmentation method based on normal mode distortions, and successful training of a neural wavefunction on these with continuously sampled geometries. Additionally, we improve the initialization of electrons around the molecule, reducing the computational overhead.
- As a final result, an accurate neural wavefunction, which shows for the first time zero-shot capabilities, i.e. high-accuracy energy predictions without additional optimization steps, on new systems. In particular it achieves better absolute energies than well established, high-accuracy gold-standard reference methods, such as CCSD(T)-3Z on unseen systems without any finetuning (cf. Fig. 2a).

Overview of the paper In Sec. 2 we outline our method and the procedure to optimize a transferable wavefunction across molecules. In Sec. 3 we thoroughly test the accuracy of our obtained wavefunction, by analyzing absolute energies (Sec. 3.1), relative energies (Sec. 3.2), and the impact of design choices in an ablation study (Sec. 3.3). Throughout this work, we compare against other high-accuracy methods, in particular results obtained by state-of-the-art DL-VMC methods and CCSD(T). Finally we analyze the scalability and limitations of our base model, by applying it to a large-scale dataset in Sec. 3.4, before a discussion and outlook for future research in Sec. 4.

2 Methods

Our approach is divided into two parts (cf. Fig. 1): On the one hand, a wavefunction ansatz, containing an electron embedding, orbital embedding and a Slater determinant. On the other hand a method for geometry sampling based on normal-mode distortions.

2.1 Our wavefunction ansatz

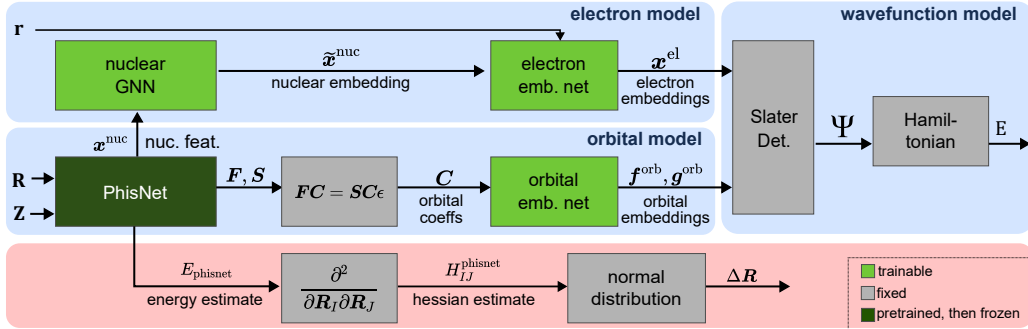


Figure 1: **Overview of our approach:** Wavefunction ansatz (top) and geometry sampling (bottom)

A single forward-pass for our wavefunction model

$$\psi = \sum_{d=1}^{N_{\text{det}}} \det[\phi_k^d(\mathbf{x}_i^{\text{el}})], \quad \phi_k^d(\mathbf{x}_i^{\text{el}}) = \sum_{I=1}^{N_{\text{nuc}}} \langle \mathbf{x}_i^{\text{el}}, \mathbf{f}_{Ikd}^{\text{orb}} \rangle e^{-r_{iI} g_{Ikd}^{\text{orb}}}, \quad i, k = 1, \dots, n_{\text{el}} \quad (3)$$

can again be divided into three blocks: An electron model acting on nuclear and electron coordinates, generating electron embeddings $\mathbf{x}^{\text{el}}$; an SE(3)-equivariant orbital model acting only on nuclear coordinates, generating orbital embeddings $\mathbf{g}^{\text{orb}}$ and $\mathbf{f}^{\text{orb}}$; a Slater determinant combining electron- and orbital-embeddings, and ensuring anti-symmetry of the wavefunction.

Message passing neural network Throughout this work we use message passing neural networks (MPNN), to operate on the graph of particles which are connected by edges containing information about their relative positions. The electron-electron, electron-nuclear and nuclear-nuclear edges are embedded with a multi-layer perceptron (MLP),

$$e_{ij}^{\text{el-el}} = \text{MLP}([r_{ij}, r_{ij}]) \quad e_{iI}^{\text{el-nuc}} = \text{MLP}([r_{iI}, r_{iI}]) \quad e_{IJ}^{\text{nuc-nuc}} = \text{MLP}([R_{IJ}, R_{IJ}]), \quad (4)$$

by using a concatenation ($[\cdot]$) of distance and difference vectors, and separate weights for each MLP. A single message passing step is decomposed into the following operations

$$\tilde{a}_i^{\text{rec}} = \text{MessagePassing}(a_i^{\text{rec}}, \{a_j^{\text{send}}\}, \{e_{ij}\}) \quad (5)$$

$$= \sigma \left(\text{Linear}(a_i^{\text{rec}}) + \text{Linear} \left(\sum_j \text{Linear}(a_j^{\text{send}}) \odot \text{Linear}(e_{ij}) \right) \right) \quad (6)$$

for a receiving particle $\tilde{a}_i^{\text{rec}}$ and the set of sending particles $\{a_j^{\text{send}}\}$, connected via their edges $\{e_{ij}\}$. By σ we denote the non-linear activation and with $\odot$ the element-wise multiplication along the feature dimension. An MPNN is obtained by stacking MessagePassing layers

$$\text{MPNN}(a_i, \{a_j\}, \{e_{ij}\}) = \text{MessagePassing}(\dots \text{MessagePassing}(a_i, \{a_j\}, \{e_{ij}\})) \quad (7)$$

In the following we use these message passing steps to model all inter-particle interactions.

SE(3)-equivariant orbital model The orbital model is a simplified version of PhisNet[20], a neural network predicting the overlap matrix S and the Fock matrix F , via nuclear embeddings x^{nuc} :

$$x^{\text{nuc}} = \text{phisnet}_{\theta}(R, Z) \quad S_{IJ} = s_{\theta}(x_I^{\text{nuc}}, x_J^{\text{nuc}}) \quad F_{IJ} = f_{\theta}(x_I^{\text{nuc}}, x_J^{\text{nuc}}). \quad (8)$$

Here $x^{\text{nuc}} \in \mathbb{R}^{N_{\text{nuc}} \times (L+1)^2 \times N_{\text{channels}}}$, and S_{IJ} and F_{IJ} are each in $\mathbb{R}^{N_{\text{basis}} \times N_{\text{basis}}}$. The basis-set size of the predicted orbitals is denoted by N_{basis} and the feature dimension of the nuclear embeddings by N_{channels} . The full overlap- and Fock-matrices are assembled from the corresponding blocks S_{IJ} and F_{IJ} , leading to matrices of shape $[N_{\text{nuc}} N_{\text{basis}} \times N_{\text{nuc}} N_{\text{basis}}]$. Each layer of PhisNet is SE(3)-equivariant, ensuring that any 3D-rotation or inversion of the input coordinates R , leads to an equivalent rotation of its outputs. This is done by splitting any feature vector into representations of varying harmonic degree $l = 0, \dots, L$, each with components $m = -l, \dots, l$. A detailed description of SE3-equivariant networks in general, as well as PhisNet in particular can be found in [20]. A list of changes and simplifications we made to PhisNet can be found in Appendix C.

The orbital embeddings (corresponding to orbital expansion coefficients in a conventional quantum chemistry calculation) are obtained by solving the generalized eigenvalue problem:

$$FC_k = SC_{k \in k} \quad \hat{x}_{Ik}^{\text{orb}} = \text{reshape}(C_k, [N_{\text{nuc}}, N_{\text{basis}}])_I \quad (9)$$

It has been shown empirically that it is beneficial to obtain the orbital coefficients $\hat{x}_{Ik}^{\text{orb}}$ as solutions to this generalized eigenvalue problem, rather than predicting them directly [27] as functions of R and Z . This is because the orbital coefficients are neither unique, nor do they share the molecule’s symmetry. The matrices F and S on the other hand are unique and transform equivariantly under E(3)-transformations of the molecule, leading to a well defined learning problem.

Following [25], we do not use the orbital energies ϵ_k and obtain the backflow factors f and exponents g , by first localizing the resulting orbitals using the Foster-Boys localization [28] (cf. Appendix B) and subsequently using an MPNN and MLP acting on the orbital embeddings.

$$\tilde{x}_{Ik}^{\text{orb}} = \sum_{n=1}^{N_{\text{orb}}} U_{kn}^{\text{loc}} \hat{x}_{In}^{\text{orb}}, \quad x_{Ik}^{\text{orb}} = \text{MPNN}(\tilde{x}_{Ik}^{\text{orb}}, \{\tilde{x}_{Jk}^{\text{orb}}\}, \{e_{IJ}^{\text{nuc-nuc}}\}) \quad (10)$$

$$f_{Ik}^{\text{orb}} = \text{MLP}(x_{Ik}^{\text{orb}}), \quad f_{Ik}^{\text{orb}} \in \mathbb{R}^{N_{\text{nuc}} \times N_{\text{orb}} \times N_{\text{det}} \times N_{\text{emb}}} \quad (11)$$

$$g_{Ik}^{\text{orb}} = \text{MLP}(x_{Ik}^{\text{orb}}), \quad g_{Ik}^{\text{orb}} \in \mathbb{R}^{N_{\text{nuc}} \times N_{\text{orb}} \times N_{\text{det}}} \quad (12)$$

Electron model The electron embedding is a message passing neural network. To incorporate the geometric information of the molecule considered, we leverage the equivariant prediction of the PhisNet nuclear embeddings x^{nuc} , by first performing a message passing step between the nuclear embeddings

$$\hat{x}_I^{\text{nuc}} = \text{MLP}(x_I^{\text{nuc}}) \quad \tilde{x}_I^{\text{nuc}} = \text{MessagePassing}(\hat{x}_I^{\text{nuc}}, \{\hat{x}_J^{\text{nuc}}\}, \{e_{IJ}^{\text{nuc-nuc}}\})$$

and then using these features to initialize the electron embeddings

$$\mathbf{x}_i^{\text{el},0} = \text{MessagePassing}(\mathbf{0}, \{\tilde{\mathbf{x}}_J^{\text{nuc}}\}, \{\mathbf{e}_{iJ}^{\text{el-nuc}}\}),$$

by using a zero vector $\mathbf{0}$ for the initial receiving electrons. This differentiates our electron model from previously proposed methods [3, 6, 8], leading to a better generalization when optimized across molecules (cf. Sec. 3.3). The final step of the embedding is a multi-iteration message passing between electron embeddings to capture the necessary electron-electron interaction

$$\mathbf{x}_i^{\text{el}} = \text{MPNN}(\mathbf{x}_i^{\text{el},0}, \{\mathbf{x}_j^{\text{el},0}\}, \{\mathbf{e}_{ij}^{\text{el-el}}\}),$$

resulting in a N_{emb} -dimensional representation for each electron $\mathbf{x}_i^{\text{el}} \in \mathbb{R}^{N_{\text{emb}}}$, $i = 1, \dots, n_{\text{el}}$.

Overall E(3)-equivariance Like existing approaches [3, 8, 17] our overall wavefunction does not enforce E(3)-symmetry. This is because the wavefunction can have lower symmetry than the molecule [23], for example in the case of the excited states of a hydrogen atom. We therefore choose all parts of the network that act purely on nuclear coordinates (i.e. the PhisNet model and the energy/hessian estimate) to be equivariant under E(3)-transformations. All parts of the network acting on electron coordinates (in particular the electron model) break this symmetry by depending explicitly on the cartesian coordinates of the electrons. The architecture is thus only invariant under translations, but not under rotations or inversions. We bias the model towards approximately invariant energies using data augmentation as discussed in Sec. 2.3.

2.2 Sampling

Markov Chain Monte Carlo (MCMC) sampling of electron positions We use MCMC to draw samples $\mathbf{r}$ from the probability distribution $\psi(\mathbf{r})^2$, to evaluate the expectation value of Eq. 2. One notable difference compared to other works is our initialization $\mathbf{r}^0$ of the Markov Chain. In the limit of infinite steps, the samples are distributed according to ψ^2 , but for a finite number of steps the obtained samples strongly depend on $\mathbf{r}^0$. This issue is typically addressed by a "burn-in", where MCMC is run for a fixed number of steps (without using the resulting samples) to ensure that $\mathbf{r}$ has diffused to state of high probability. Previous work has initialized $\mathbf{r}^0$ using a Gaussian distribution of the electrons around the nuclei. We find that this initialization is far from the desired distribution ψ^2 and thus requires $\approx 10^5$ MCMC steps to reach the equilibrium distribution. We instead initialize $\mathbf{r}^0$ by samples drawn from an exponential distribution around the nuclei, which much better approximates the correct distribution and thus equilibrates substantially faster (cf. Appendix A). We find that exponential initialization reduces the required number of burn-in steps by ca. 50%, reducing the computational cost of a 500-step zero-shot evaluation by ca. 5%.

Normal mode sampling of geometries Since DL-VMC is an ab-initio method, we do not require a labeled dataset of reference energies, but to obtain a transferable wavefunction, which generalized well to new systems, a diverse dataset of molecular geometries $\mathbf{R}$ is required. Starting with an initial set of geometries $\mathbf{R}^0$, we update $\mathbf{R}$ on the fly, by perturbing each geometry every 20 optimization steps by adding random noise $\Delta\mathbf{R}$ to the nuclear coordinates. Using uncorrelated, isotropic random noise for $\Delta\mathbf{R}$ would yield many non-physical geometries $\mathbf{R}'$, since the stiffness of different degrees of freedom can vary by orders of magnitude. Intuitively we want to make large perturbations along directions in which the energy changes slowly, and vice-versa. We achieve this by sampling $\Delta\mathbf{R}$ from a correlated normal distribution

$$\Delta\mathbf{R} \sim \mathcal{N}(\alpha(\mathbf{R}^0 - \mathbf{R}), \beta\mathbf{H}_{\text{phis}}^{-1}) \quad \mathbf{R}' = \mathbf{R} + \Delta\mathbf{R}. \quad (13)$$

The bias term $\alpha(\mathbf{R}^0 - \mathbf{R})$ ensures that geometries stay sufficiently close to their starting point $\mathbf{R}^0$. The covariance matrix is chosen proportional to the pseudo-inverse of the hessian of the energy E^{phis} , which is predicted from the scalar component of the nuclear embeddings using a pre-trained MLP. By using $\mathbf{H}_{\text{phis}}^{-1}$ as covariance matrix, we take large steps along soft directions and small steps along stiff directions, thus avoiding unphysical geometries with very high energies.

$$E^{\text{phis}} = \sum_{I=1}^{N_{\text{nuc}}} \text{MLP}(\mathbf{x}_I^{\text{nuc}}), \quad H_{I\zeta, J\xi}^{\text{phis}} = \frac{\partial^2 E^{\text{phis}}}{\partial \mathbf{R}_{I\zeta} \partial \mathbf{R}_{J\xi}}, \quad I, J = 1 \dots N_{\text{nuc}} \quad \zeta, \xi = 1 \dots 3 \quad (14)$$

After distorting the nuclear coordinates $\mathbf{R}$, we also adjust the electron positions $\mathbf{r}$, using the space-warp coordinate transform described in [29], which effectively shifts the electrons by a weighted average of the shift of their neighbouring nuclei. In addition to this distortion of the molecule, we also apply a random global rotation to all coordinates, to obtain a more diverse dataset.

2.3 Optimization

To obtain orbital descriptors (and energies to calculate the hessian of the energy) we pre-train PhisNet against the Fock matrix, the overlap matrix, the energy and the forces of Hartree-Fock calculations in a minimal basis set across 47k molecules. Further details of the loss function, dataset, and the adaptations to PhisNet can be found in Appendix C. For all subsequent experiments, we freeze the parameters of PhisNet. A full DL-VMC calculation to obtain a ground-state energy prediction can be divided into three consecutive steps:

1. **Supervised optimization using PhisNet:** Initially, the neural-network orbitals (cf. Eq. 3) are optimized to minimize the residual against orbitals obtained from PhisNet. It ensures that the initial wavefunction roughly resembles the true ground-state and is omitted when fine-tuning an already optimized base model.
2. **Variational optimization:** Minimization of the energy (cf. Eq. 2) by drawing samples from ψ^2 using MCMC and updating the wavefunction parameters using the KFAC optimizer [30].
3. **Evaluation:** For inference of the ground-state energy, we sample electron positions using MCMC, and evaluate the energy using Eq. 2 without updating θ .

To train a multi-geometry transferable wavefunction we further divide the variational optimization of a neural wavefunction into two steps:

1. **Pre-training:** A single wavefunction model is trained across many molecules and geometries. In every gradient step we only consider a single geometry per batch. The next geometry to optimize is chosen based on the energy variance as proposed by [10]. To sample continuously the space of molecular geometries we distort each geometry every 20 optimization steps as described in Sec. 2.2. We refer to evaluations of this model on new systems as "zero-shot".
2. **Fine-tuning:** A small number of additional variational optimization steps is done using geometries of interest, starting from the weights of a pre-trained base model. This procedure yields a model that is specialized to the molecule at hand and typically yields more accurate energies on the specific problem than the raw pre-trained model.

3 Results

We pre-train our wavefunction model on a dataset of 98 molecules (699 conformers) for 256k optimization steps using the architecture and training procedure outlined in Sec. 2. Below we demonstrate the performance of this model, for zero-shot evaluations and after subsequent fine-tuning.

3.1 Accuracy of pre-trained model for absolute energies

To test the transfer capabilities of the model, we evaluate it on test-sets, which each contain 4 randomly chosen and perturbed molecules, grouped by molecule size (measured as the number of non-Hydrogen atoms). To avoid train/test leakage, we excluded all molecules that are part of these test sets from the training set (cf. Appendix D). Although the model has only been trained on molecules containing up to 4 heavy atoms, we evaluate its performance across the full range up to 7 heavy atoms. We find that for molecules containing up to 6 heavy atoms our method outperforms CCSD(T) with a 2Z basis set and outperforms CCSD(T) with a 4Z basis set after only 4k fine-tuning steps. This is a large improvement over the state of the art: Zero-shot evaluations by Gao et al. [17] did not manage to outperform a Hartree-Fock baseline, even on the toy system of Hydrogen-chains. Similarly, Scherbela et al. [25] achieve high accuracy after fine-tuning, but result that are worse than Hartree-Fock in a zero-shot setting. In contrast, our improvements to their method increase zero-shot accuracy by more than 2 orders of magnitude.

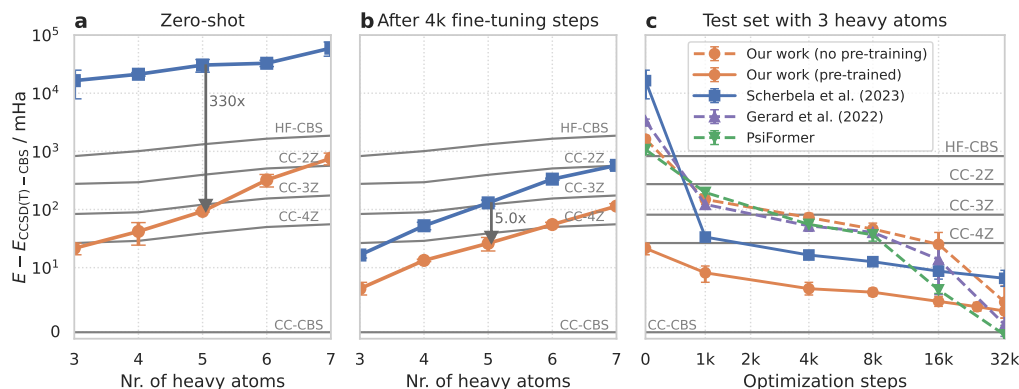


Figure 2: **Absolute energies:** Energies relative to CCSD(T)-CBS (complete basis set limit) when re-using the pre-trained model on molecules of varying size without optimization (a) and after fine-tuning (b). (c) depicts energy for the test set containing 3 heavy atoms as a function of optimization steps and compares against SOTA methods. Solid lines are with pre-training, dashed lines without. Gray lines correspond to conventional methods: Hartree-Fock in the complete basis set limit (HF-CBS), and CCSD(T) with correlation consistent basis basis sets of double to quadruple valence (CC-nZ).

We furthermore find that fine-tuning pre-trained wavefunctions can be more cost effective than well established conventional methods. For a typical molecule containing 5 heavy atoms, we require 9.5 node-hours for 4k fine-tuning steps, achieving accuracy that surpasses CCSD(T)-4Z. In contrast, CCSD(T)-4Z requires 10.5 node-hours and 16k steps of PsiFormer (achieving the same accuracy), require 53 node-hours. All run-times are listed in Tab. 3 of Appendix I.

To further test the accuracy of our method, we evaluate for 3 heavy atoms the performance with increasing fine-tuning steps (cf. Fig. 2c). We compare our work (with and without fine-tuning) against CCSD(T) and reference calculations done with state-of-the-art DL-VMC methods [6, 8]. For up to 16k optimization steps, our pre-trained model yields energy errors that are 1-2 orders of magnitude lower than other reference methods. After longer optimization the method by Gerard et al. [8] and PsiFormer [6] surpass our predictions. We hypothesize that this is not to blame on pre-training, but that our orbital prediction framework, which allows us to optimize across molecule, yields less expressive wavefunctions than a fully trainable backflow. This is demonstrated by the fact that a pre-trained and non-pre-trained model converge to the same energy in the limit of long optimization.

Like the absolute energies, also the variance of the local energies (another measure of wavefunction accuracy) is substantially improved by our method and results are depicted in Appendix E.

3.2 Accuracy for relative energies

While Sec. 3.1 demonstrates high accuracy for absolute energies, we find that relative energies (being the small difference between two large absolute energies) can be unsatisfactory in the zero-shot regime, and a small number of fine-tuning steps is required to reach quantitatively correct relative energies. Fig. 3 demonstrates this issue on 4 distinct systems, each highlighting a different challenge for our model. For each system, we evaluate our pre-trained base model without any system specific optimization (zero-shot), and after 4000 fine-tuning steps. The fine-tuning optimization is done separately for each of the 4 systems, analogously to Sec. 2.3, yielding 4 distinct wavefunctions that each represent the ground-state wavefunctions of all considered geometries per system. Results after more fine-tuning steps can be found in Appendix F.

Bicyclobutane conformers Fig. 3a depicts the energies of 5 conformers of bicyclobutane relative to the energy of its initial structure. The system is of interest, because CCSD(T) severely underestimates the energy of the dis-TS conformer by ≈ 60 mHa [31]. While our zero-shot results yield the correct sign for the relative energies, they are quantitatively far off from the gold-standard DMC reference calculation [31], in particular for the dis-TS geometry, where we overestimate the relative energy

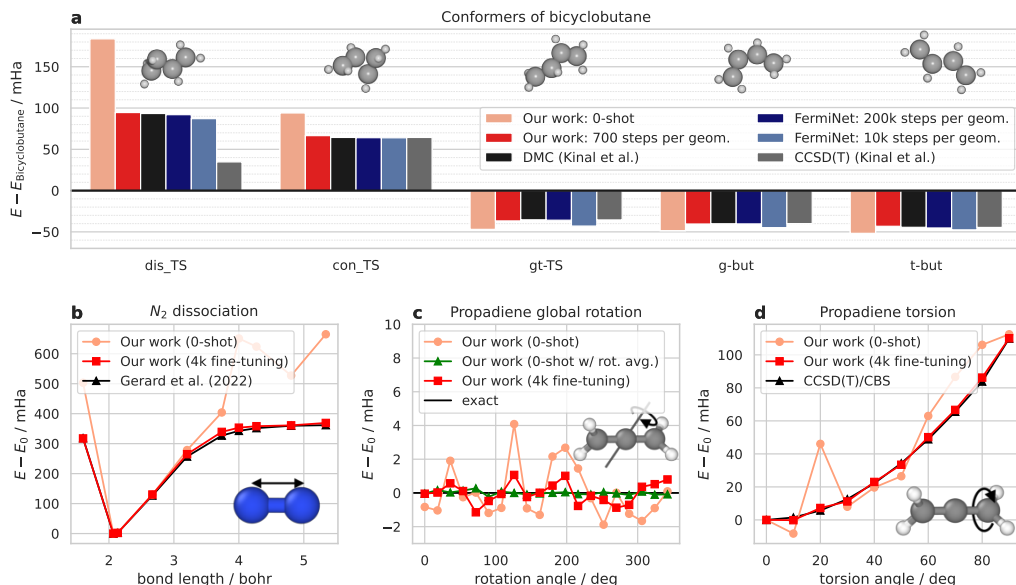


Figure 3: **Challenging relative energies:** Relative energies obtained with and without fine-tuning on 4 distinct, challenging systems, compared against high-accuracy reference methods. a) Relative energy of bicyclobutane conformers vs. the energy of bicyclobutane b) Potential energy surface (PES) of N_2 , c) global rotation of propadiene d) relative energy of twisted vs. untwisted propadiene.

by 90 mHa. However when fine-tuning our model for only 700 steps per geometry (4k total), we obtain relative energies that are in close agreement with DMC (max. deviation 2.1 mHa). Comparing to FermiNet [5] we find that our results are more accurate than a FermiNet calculation after 10k steps (requiring twice our batch size; max. deviation 7.5 mHa), and slightly less accurate than a FermiNet calculation optimized for 200k steps (max. deviation 1.4 mHa). As opposed to CCSD(T) our model does not suffer from systematic errors, even for the challenging dis-TS geometry. A table of all relative energies can be found in Appendix F.

Nitrogen dissociation When evaluating the energy of an N_2 molecule at various bond-lengths we obtain high zero-shot accuracy near the equilibrium geometry ($d = 2.1$ bohr; in training set), but substantially lower accuracy at smaller or large bond lengths (cf. Fig. 3b). This is due to the lack of dissociated atoms in the pre-training dataset and again mostly remedied by finetuning. In the most challenging regime around $d = 3.7$, even with fine-tuning we obtain energies that are 12 mHa above high-accuracy results obtained by Gerard et al. [8], indicating the need for longer fine-tuning.

Global rotation of propadiene Energies of molecules are invariant under global translation or rotation of all particle positions. The wavefunction however is not invariant and neither is our wavefunction ansatz, which can lead to different energies for rotated copies of a molecule. When evaluating the energy of our model on 20 copies of propadiene (C_3H_4) rotated around a random axis, we find typical energy variations of ± 1 mHa, but also a individual outliers, deviating by up to 5 mHa. This highlights a dilemma facing all existing DL-VMC models: On the one hand, constraining the wavefunctions to be fully invariant under rotation (or even just invariant under symmetries of the Hartree-Fock orbitals) is too restrictive to express arbitrary ground-state wavefunctions [23]. On the other hand, our approach of biasing the model towards rotation-invariant energies by data augmentation, appears to be helpful but not to be fully sufficient. When evaluating an earlier checkpoint of our base model (trained for 170k epochs instead of 256k), we observe mean energy variations of ± 3 mHa and outliers of up to 30 mHa, indicating the positive impact of prolonged training with data augmentation. We again find 4000 fine-tuning steps split across all geometries are sufficient to reduce errors below chemical accuracy of 1.6 mHa.

In addition to augmenting data during training time, we can also augment the data during inference time, to obtain fully rotation-invariant energies, from an approximately rotation invariant model. We

achieve this by randomly rotating the molecule at every inference step and averaging across all rotated geometries. Since the Monte-Carlo estimate is already an average across many different samples anyways, this comes at no additional computational cost. Fig. 3c shows that with rotation averaging we obtain energies that are fully invariant, up to Monte-Carlo noise.

Twisted propadiene Twisting one of the C=C bonds of propadiene leads to a transition state with an energy difference of 110 mHa. Evaluating the energy without fine-tuning on an equidistant grid of torsion angles, we obtain the correct barrier height (112 mHa), but also deviations of up to 40 mHa. During pre-training we purposefully sampled twisted molecules, but only included equilibrium geometries, transition geometries, and one intermediate twist (cf. Appendix D). This seems sufficient for correct zero-shot barrier heights, but insufficient for high accuracy along the full path. Short fine-tuning (4k steps distributed across all 10 geometries) yields excellent agreement with CCSD(T): ~ 0.2 mHa discrepancy for the barrier height and a maximum deviation of 2 mHa along the path.

3.3 Ablation studies

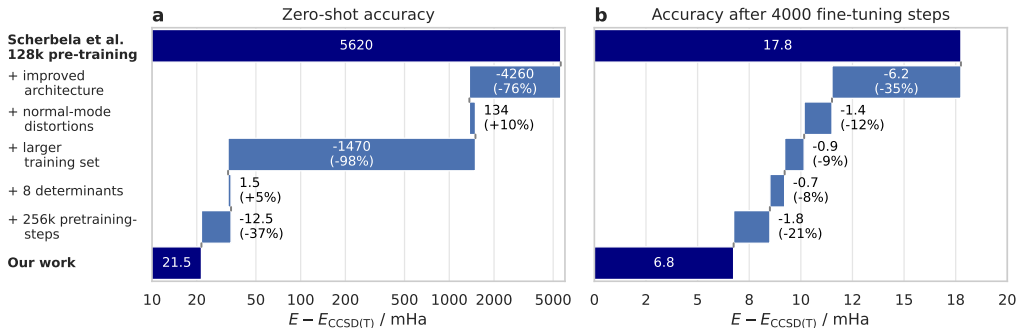


Figure 4: **Ablation study:** Breakdown of absolute energy difference between Scherbela et al. [25] and our work, when evaluating the respective base models on the test set consisting of 3 heavy atoms. Panel a shows accuracy gains for zero-shot evaluation, panel b the accuracy after 4k fine-tuning steps. First and last row depict their and our absolute energy respectively, intermediate bars show the subsequent improvements of various changes in mHa (and %). Note that panel a. is plotted on a logarithmic scale due to the large energy difference.

To analyze the relative importance of our changes, we break down the accuracy gap between our work and the prior work by Scherbela et al. [25] in Fig. 4. We start with their model checkpoint trained for 128k epochs on their dataset consisting of 18 different compounds. Next, we train a model using our improved architecture with their dataset and methodology, already reducing the zero-shot error by 76% and fine-tuning error by 35%. The next model additionally uses normal-mode distortions to augment the training dataset, decreasing the fine-tuning error by another 12%. Additionally increasing the training set size to our dataset containing 98 molecules and the corresponding torsional conformers, reduces the zero-shot error by 98% and yields modest improvements in the fine-tuning case. Increasing the number of determinants from 4 to 8 and increasing the number of pre-training steps from 128k to 256k improve fine-tuning accuracy by 8% and 21% respectively. The two largest contributions overall are the improved architecture – yielding substantial gains both in zero-shot and fine-tuning – as well as the larger training set, which is crucial for high zero-shot accuracy. This is consistent with [25], which found that both model size and training set size do improve performance, while pre-training duration shows diminishing returns after 256k steps. Since we pre-train the PhisNet model against Hartree-Fock references and do not update its parameters during optimization, the PhisNet model by itself contributes no substantial accuracy improvement. It impacts energy predictions indirectly, by providing pre-trained nuclear embeddings x^{nuc} and enabling fast geometry distortions and rotations during training and inference, which in turn improve accuracy.

3.4 Large-scale experiment

To showcase the scalability of our approach, we evaluate zero-shot predictions of the absolute energies for a subset of 250 molecules from the QM7 dataset [32], containing molecules with 14-58 electrons.

Since CCSD(T) calculations would be prohibitively expensive for a dataset of this size, in Fig. 5a we compare against density functional theory (DFT) reference calculations (PBE0+MBD, [26]). Fig. 5b breaks down the energy differences compared to DFT, by molecule size. For molecules with less than 5 heavy atoms we obtain energies that are lower than DFT by 10-120 mHa. Because our ansatz is variational (in contrast to DFT), our lower energies translate to a more accurate prediction of the true ground-state energy, confirming again our high zero-shot accuracy. With increasing system size the accuracy deteriorates, due to increasing extrapolation and out-of-distribution predictions, consistent with our results in Sec. 3.1. One reason for the large energy differences in larger molecules is the increasing inter-atomic distance within the molecules with increasing number of atoms (cf. Fig. 5c). While the largest inter-atomic distance observed in the training set is 11 bohr, the evaluation set contains distances up to 17 bohr. This issue could be overcome by employing a distance cutoff, as it is already applied in supervised machine-learned potential energy surface predictions [19, 33] or has just recently been incorporated into a neural wavefunction [17] via an exponential decay with increasing inter-particle distance. Another potential solution is to include larger molecules or separated molecule fragments in the pre-training molecule dataset, reducing the extrapolation regime.

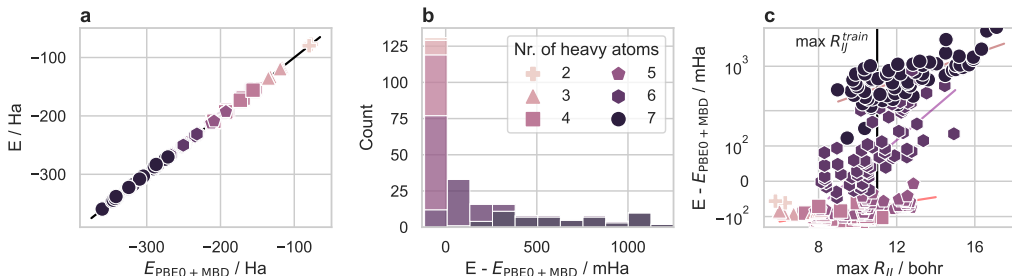


Figure 5: **Zero-shot on QM7:** a) Our zero-shot energies vs. DFT [26] b) Histogram of energy residuals (truncated at 1.25 Ha for clarity) c) Residuals vs. largest inter-molecular distance R_{IJ} .

4 Discussion

We have presented to our knowledge the first ab-initio wavefunction model, which achieves high-accuracy zero-shot energies on new systems (Sec. 3.1 and Sec. 3.4). Our pre-trained wavefunction yields more accurate total energies than CCSD(T)-2Z across all molecule sizes and outperforms CCSD(T)-3Z on molecules containing up to 5 heavy atoms, despite having been trained only on molecules containing up to 4 heavy atoms. We find that relative energies of our model are qualitatively correct without fine-tuning, but need on the order of 4000 fine-tuning steps to reach chemical accuracy of 1.6 mHa (Sec. 3.2). This is a substantial improvement over previous work, which so far has fallen in two categories: High-accuracy ansätze (such as [3, 6]) that cannot generalize across molecules and thus need ca. 10x more compute to reach the same accuracy, or methods that can generalize ([17, 25]), but yield orders of magnitude lower accuracy in the zero- or few-shot regime. We demonstrate in Sec. 3.3 that these improvements are primarily driven by an improved architecture (containing a more expressive electron embedding and an ML-based orbital model), improved geometry sampling, and a larger training dataset.

While results are encouraging in the zero- and few-shot regime, open questions for further research abound. The most pressing issues are currently limited zero-shot accuracy for relative energies, and potentially limited expressiveness of the ansatz in the regime of very long optimization. Zero-shot accuracy could be further improved by training on an even larger dataset, further improved geometry sampling (in particular of torsion angles), and an interaction cut-off to avoid previously unseen particle pairs for new large molecules. Furthermore SE(3)-symmetry of the wavefunction should be explored further, since currently only the orbital part of our architecture is SE(3)-equivariant. We experimented with a fully equivariant architecture, but found the resulting wavefunctions to not be expressive enough. To improve overall accuracy, attention based embeddings [6, 34] could be pursued. Additionally, we currently freeze the weights of the orbital embedding to simplify the architecture and avoid back-propagation through the iterative orbital localization procedure. Optimizing these weights in addition to the electron embedding will lead to a more expressive ansatz.

Acknowledgements

We gratefully acknowledge financial support from the following grants: Austrian Science Fund FWF Project I 3403 (P.G.), WWTF-ICT19-041 (L.G.). The computational results have been achieved using the Vienna Scientific Cluster (VSC). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

References

- [1] M. Born and R. Oppenheimer. “Zur Quantentheorie der Molekeln”. In: *Annalen der Physik* 389.20 (1927), pp. 457–484. DOI: <https://doi.org/10.1002/andp.19273892002>.
- [2] László Gyevi-Nagy, Mihály Kállay, and Péter R. Nagy. “Accurate Reduced-Cost CCSD(T) Energies: Parallel Implementation, Benchmarks, and Large-Scale Applications”. In: *Journal of Chemical Theory and Computation* 17.2 (Feb. 2021), pp. 860–878. DOI: 10.1021/acs.jctc.0c01077.
- [3] David Pfau et al. “Ab Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks”. In: *Phys. Rev. Res.* 2.3 (Sept. 2020), p. 033429. DOI: 10.1103/PhysRevResearch.2.033429.
- [4] W. K. Hastings. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications”. In: *Biometrika* 57.1 (Apr. 1970), pp. 97–109. DOI: 10.1093/biomet/57.1.97.
- [5] James S. Spencer et al. “Better, Faster Fermionic Neural Networks”. In: (2020).
- [6] Ingrid von Glehn, James S. Spencer, and David Pfau. *A Self-Attention Ansatz for Ab-initio Quantum Chemistry*. Nov. 2022. DOI: 10.48550/arXiv.2211.13672.
- [7] Weili Ren et al. “Towards the ground state of molecules via diffusion Monte Carlo on neural networks”. In: *Nature Communications* 14.1 (Apr. 2023), p. 1860. DOI: 10.1038/s41467-023-37609-3.
- [8] Leon Gerard et al. “Gold-Standard Solutions to the Schrödinger Equation Using Deep Learning: How Much Physics Do We Need?” In: *Advances in Neural Information Processing Systems*. Oct. 2022.
- [9] M. T. Entwistle et al. “Electronic excited states in deep variational Monte Carlo”. In: *Nature Communications* 14.1 (Jan. 2023), p. 274. DOI: 10.1038/s41467-022-35534-5.
- [10] Michael Scherbela et al. “Solving the Electronic Schrödinger Equation for Multiple Nuclear Geometries with Weight-Sharing Deep Neural Networks”. In: *Nature Computational Science* 2.5 (May 2022), pp. 331–341. DOI: 10.1038/s43588-022-00228-x.
- [11] Yubing Qian et al. “Interatomic Force from Neural Network Based Variational Quantum Monte Carlo”. In: *The Journal of Chemical Physics* 157.16 (Oct. 2022), p. 164104. DOI: 10.1063/5.0112344.
- [12] Gino Cassella et al. “Discovering Quantum Phase Transitions with Fermionic Neural Networks”. In: *Physical Review Letters* 130.3 (Jan. 2023), p. 036401. DOI: 10.1103/PhysRevLett.130.036401.
- [13] Xiang Li, Zhe Li, and Ji Chen. “Ab initio calculation of real solids via neural network ansatz”. In: *Nature Communications* 13.1 (Dec. 2022), p. 7895. DOI: 10.1038/s41467-022-35627-1.
- [14] Giuseppe Carleo and Matthias Troyer. “Solving the Quantum Many-Body Problem with Artificial Neural Networks”. In: *Science* 355.6325 (Feb. 2017), pp. 602–606. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aag2302.
- [15] Giuseppe Carleo et al. “NetKet: A Machine Learning Toolkit for Many-Body Quantum Systems”. In: *SoftwareX* 10 (July 2019), p. 100311. DOI: 10.1016/j.softx.2019.100311.
- [16] Yuan-Hang Zhang and Massimiliano Di Ventra. “Transformer Quantum State: A Multipurpose Model for Quantum Many-Body Problems”. In: *Physical Review B* 107.7 (Feb. 2023), p. 075147. DOI: 10.1103/PhysRevB.107.075147.
- [17] Nicholas Gao and Stephan Günnemann. *Generalizing Neural Wave Functions*. Feb. 2023. DOI: 10.48550/arXiv.2302.04168.
- [18] Simon Batzner et al. “E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials”. In: *Nature Communications* 13.1 (May 2022), p. 2453. DOI: 10.1038/s41467-022-29939-5.
- [19] Ilyes Batatia et al. “MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields”. In: *Advances in Neural Information Processing Systems*. 2022.
- [20] Oliver Unke et al. “SE(3)-Equivariant Prediction of Molecular Wavefunctions and Electronic Densities”. In: *Advances in Neural Information Processing Systems*. 2021.
- [21] M. Gastegger et al. “A Deep Neural Network for Molecular Wave Functions in Quasi-Atomic Minimal Basis Representation”. In: *The Journal of Chemical Physics* 153.4 (July 2020), p. 044123. DOI: 10.1063/5.0012911.

- [22] Albert Musaelian et al. "Scaling the Leading Accuracy of Deep Equivariant Models to Biomolecular Simulations of Realistic Size". In: (Apr. 2023). DOI: 10.48550/arXiv.2304.10061. arXiv: 2304.10061.
- [23] Nicholas Gao and Stephan Günnemann. "Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions". In: *arXiv:2110.05064 [physics]* (Nov. 2021). arXiv: 2110.05064 [physics].
- [24] Nicholas Gao and Stephan Günnemann. "Sampling-free Inference for Ab-Initio Potential Energy Surface Networks". In: (2022). DOI: 10.48550/ARXIV.2205.14962.
- [25] Michael Scherbela, Leon Gerard, and Philipp Grohs. "Towards a Foundation Model for Neural Network Wavefunctions". In: *arXiv.org* (Mar. 2023).
- [26] Johannes Hoja et al. "QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules". In: *Scientific Data* 8.1 (Feb. 2021), p. 43. DOI: 10.1038/s41597-021-00812-2.
- [27] Ruud van Workum, Joao Malhado, and Philipp Marquetand. *CASNet: Learning Complete Active Space Orbitals Using Message Passing Neural Networks*. June 2023. DOI: 10.26434/chemrxiv-2023-1wj87.
- [28] János Pipek and Paul G. Mezey. "A Fast Intrinsic Localization Procedure Applicable for Ab Initio and Semiempirical Linear Combination of Atomic Orbital Wave Functions". In: *The Journal of Chemical Physics* 90.9 (May 1989), pp. 4916–4926. DOI: 10.1063/1.456588.
- [29] Claudia Filippi and C. J. Umrigar. "Correlated Sampling in Quantum Monte Carlo: A Route to Forces". In: *Physical Review B* 61.24 (June 2000), R16291–R16294. DOI: 10.1103/PhysRevB.61.R16291.
- [30] James Martens and Roger Grosse. "Optimizing neural networks with kronecker-factored approximate curvature". In: *International conference on machine learning*. PMLR. 2015, pp. 2408–2417.
- [31] Armağan Kinal and Piotr Piecuch. "Computational Investigation of the Conrotatory and Disrotatory Isomerization Channels of Bicyclo[1.1.0]butane to Buta-1,3-diene: A Completely Renormalized Coupled-Cluster Study". In: *The Journal of Physical Chemistry A* 111.4 (Feb. 2007). Publisher: American Chemical Society, pp. 734–742. ISSN: 1089-5639. DOI: 10.1021/jp065721k. URL: <https://doi.org/10.1021/jp065721k>.
- [32] Matthias Rupp et al. "Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning". In: *Phys. Rev. Lett.* 108 (5 Jan. 2012), p. 058301. DOI: 10.1103/PhysRevLett.108.058301.
- [33] Kristof T. Schütt et al. "SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions". In: (June 2017). DOI: 10.48550/arXiv.1706.08566.
- [34] Gabriel Pescia et al. *Message-Passing Neural Quantum States for the Homogeneous Electron Gas*. May 2023. DOI: 10.48550/arXiv.2305.07240.
- [35] Qiming Sun et al. "Recent Developments in the PySCF Program Package". In: *The Journal of Chemical Physics* 153.2 (July 2020), p. 024109. DOI: 10.1063/5.0006074.
- [36] Mario Geiger and Tess Smidt. *E3nn: Euclidean Neural Networks*. July 2022. arXiv: 2207.09453 [cs].
- [37] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: arXiv:1412.6980 (Jan. 2017). DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs].
- [38] Greg Landrum. *RDKit: Open-source cheminformatics*. <https://github.com/rdkit/rdkit>. GitHub repository. 2009.
- [39] Frank Neese et al. "The ORCA Quantum Chemistry Program Package". In: *The Journal of Chemical Physics* 152.22 (June 2020), p. 224108. DOI: 10.1063/5.0004608.
- [40] David Pfau James S. Spencer and FermiNet Contributors. *FermiNet*. 2020. URL: <http://github.com/deepmind/ferminet>.
- [41] Aleksandar Botev and James Martens. *KFAC-JAX*. Version 0.0.1. 2022. URL: <http://github.com/deepmind/kfac-jax>.

Supplementary material for Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunction

A Electron MCMC initialization

To investigate the impact of the initial distribution of electron positions on the equilibration of the Markov Chain, we run two evaluations for a glycine molecule, using a pre-trained wavefunction. We perform no initial burn-in and use every 50th sample for energy evaluation. If the chain was perfectly equilibrated right after initialization, all sampled energies would fluctuate around the mean energy. However as Fig. 6b shows, it takes several thousand steps for the sampled energies to converge to the correct mean. This is particularly pronounced with Gaussian initialization of electron positions, which is the default in state-of-the-art DL-VMC codes such as FermiNet [3]. Using an exponential distribution of the initial electron positions much more closely resembles the correct electron density ψ^2 (cf. Fig. 6a) and thus reaches equilibrium substantially faster.

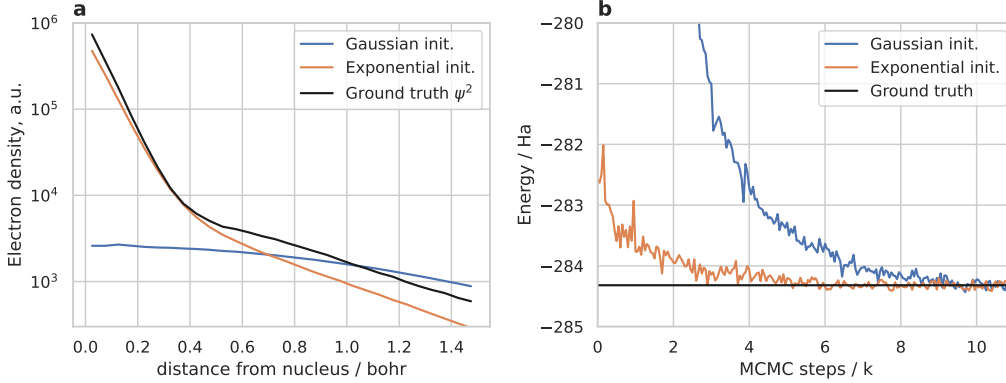


Figure 6: **Effect of electron initialization:** Initializing the electron positions using an exponential distribution instead of Gaussian, better fits the actual density (a), and thus leads to faster equilibration of observables during evaluation (b).

B Orbital localization

Our model uses orbital embeddings $\mathbf{x}^{\text{orb}}$ as inputs to parameterize the backflows $\mathbf{f}^{\text{orb}}$, and exponents $\mathbf{g}^{\text{orb}}$ of the orbitals. These orbital embeddings were introduced by Scherbela et al. [25] in the form of molecular orbital expansion coefficients, obtained from a self consistent Hartree-Fock calculation. In this setting, the coefficients $\mathbf{x}^{\text{orb}}$ are not uniquely defined, but only up to a linear transformation U with determinant ± 1

$$\mathbf{x}_{I_k}^{\text{orb}} = \sum_{n=1}^{N_{\text{orb}}} U_{kn} \hat{\mathbf{x}}_{I_k}^{\text{orb}}, \quad U \in \mathbb{R}^{N_{\text{orb}} \times N_{\text{orb}}}, \quad \det U = \pm 1. \quad (15)$$

This stems from the fact that the corresponding Hartree-Fock wavefunction is invariant under such a transformation. Consequently there is free choice, which linear combination of embeddings $\mathbf{x}^{\text{orb}}$ to choose from without any loss of information. We follow the approach of [25], by choosing U such that the corresponding Hartree-Fock orbitals are maximally localized according to the Foster-Boys

metric, i.e. minimize the spatial variance $\mathcal{L}$:

$$\phi_k(\mathbf{r}, U) = \sum_{I=1}^{N_{\text{nuc}}} \sum_{\mu=1}^{N_{\text{basis}}} b_{I\mu}(\mathbf{r}) U_{kn} \hat{x}_{In\mu}^{\text{orb}} \quad (16)$$

$$\mathcal{L}(U) = \sum_k \int \phi_k^2(\mathbf{r}, U) \mathbf{r}^2 d\mathbf{r} - \left(\int \phi_k^2(\mathbf{r}, U) \mathbf{r} d\mathbf{r} \right)^2 \quad (17)$$

Here $b_{I\mu}(\mathbf{r})$ denotes μ -th basis function of the Hartree-Fock expansion, centered on the I -th nucleus. In practice the integrals of Eq. 17 do not have to be evaluated explicitly, but can instead be computed via the overlap matrix $\mathbf{S}$. The minimization of $\mathcal{L}$ is typically done iteratively, requires on the order of 10 steps, and is readily implemented in many open-source quantum chemistry codes such as pySCF [35].

C Adaption of PhisNet

We heavily rely on PhisNet by Unke et al. [20] to obtain orbital descriptors without the need for a separate SCF calculation. Compared to their original work, we made several simplifications, which are motivated by the fact that we do not predict final high-accuracy orbitals in a large basis set, but only use PhisNet as a feature extractor by predicting orbitals in a minimal basis-set:

- **Layer Norm** We found deep variants of PhisNet to be unstable to train and mitigated the issue by adding an (equivariant) layer norm after each PhisNet module.
- **Simplified Fock matrix prediction** The original PhisNet implementation uses a final interaction between the node embeddings, before predicting the elements of the Fock matrix. We found this interaction to be superfluous for our purposes and left it out for simplicity.
- **Separate energy head** The original PhisNet computes energies via the eigenvalues obtained by diagonalization of the Fock matrix. We instead predict energies using a separate head on top of the scalar features of the node embeddings.
- **Smaller network** We changed the hyperparameters to obtain a smaller and faster version of PhisNet which obtained sufficient accuracy for our purposes. We used 2 layers (instead of 5) and $L_{\text{max}} = 2$ (instead of 4). This reduces the number of parameters from 17M to 3M.
- **Diverse training set** While the original work optimized separate models for each molecule (e.g. by training on different geometries of a molecular dynamics simulation), we optimize a single model to predict $\mathbf{F}$, $\mathbf{S}$, $\mathbf{E}$, and $\nabla \mathbf{E}$ across a dataset of 47k geometries sampled from QM7-X [26].
- **JAX re-implementation** We re-implemented PhisNet in JAX, using the e3nn library [36] to construct the SE(3)-equivariant operations.

We train the PhisNet-model on a dataset of 47k molecules from QM7X [26], using the Adam optimizer [37] on the following loss

$$\mathcal{L} = \sum_n (E^{\text{phis}}(\mathbf{R}^n, \mathbf{Z}^n) - E^{\text{ref},n})^2 + \quad (18)$$

$$+ \sum_{nI\zeta} \left(\frac{\partial}{\partial R_{I\zeta}^n} E^{\text{phis}}(\mathbf{R}^n, \mathbf{Z}^n) - G_{I\zeta}^{\text{ref},n} \right)^2 + \quad (19)$$

$$+ \sum_{nIJ\mu\nu} (F_{IJ\mu\nu}^{\text{phis}}(\mathbf{R}^n, \mathbf{Z}^n) - F_{IJ\mu\nu}^{\text{ref},n})^2 + \quad (20)$$

$$+ \sum_{nIJ\mu\nu} (S_{IJ\mu\nu}^{\text{phis}}(\mathbf{R}^n, \mathbf{Z}^n) - S_{IJ\mu\nu}^{\text{ref},n})^2. \quad (21)$$

Here E denotes energies, G denotes gradients of energies, F Fock matrices, and S overlap matrices. The indices I, J run over nuclei, the indices μ, ν over basis functions, and the index n over samples in a batch.

D Molecule datasets

Bicyclobutane For the Bicyclobutane to 1,3-butadiene transition we use the geometries from Kinal et al. [31] and compare against the reference energies stated in Spencer et al. [5].

N₂ For the N₂ potential energy surface with various bond-lengths we used the geometries including reference calculations from [8].

Propadiene The global rotation of 360° degrees for propadiene is performed on the geometry which is part of the test set for 3 heavy atoms. For the torsion experiment we used the equilibrium geometry and rotated the torsion angle by 90° degrees in steps of 10° degrees.

Zero-shot and fine-tuning dataset The results on zero-shot and few-shot predictions for increasing number of heavy atoms are performed on random subsets of molecules. For 5-7 heavy atoms we sample 4 unique and distorted molecules from QM7-X [26]. For 4 heavy atoms we use all geometries from the Bicyclobutane dataset. For 3 heavy atoms we use the ablation dataset.

Ablation dataset For the ablation study, we use one geometry per molecule from the out-of-distribution test set from Scherbela et al. [25], leading to a set of four distinct molecules. We ensure that these molecules are not part of the training set.

Large scale experiment For the large scale experiment we used a stratified random sample of 250 molecules from QM7 [32]. It contains all molecules with up to 4 heavy atoms, and additionally 65 randomly chosen molecules for 5, 6 and 7 heavy atoms each.

Pre-training dataset for transferable neural wavefunctions To train our pre-trained wavefunctions we use two datasets, consisting of 18 and 98 disparate molecules. For part of the ablation we use the dataset proposed in [25] and an extended version with 80 additional molecules. The additional compounds are a combination of all valid SMILES generated with RDKit [38] with 3 heavy atoms, allowing only Nitrogen, Oxygen and Carbon with single-, double- or triple-bonds, and all molecules up to four heavy atoms from QM7-X [26] (excluding molecule containing Fluorine). To prevent a train-test leakage, we remove Bicyclobutane (including all conformations) and the four molecules from the ablations dataset. Since the normal-mode-distortions by design do not generate strongly distorted geometries, we augment the 98-molecule-dataset with rotated dihedral angles. To generate a subset of all possible dihedral angles for a heavy-atom bond we first generate samples with equidistant angles for all possible dihedral angles and compute Hartree-Fock energies with a minimal basis-set. We include the equilibrium geometry and all extrema of the potential energy surface with respect to the rotation of a single dihedral angle if the energy of the extrema is significantly different to already included geometries of the same molecule. Additionally, we include the transition geometry towards the respective extrema and again only include energetic diverse states. Finally, to make sure that certain molecules are not underrepresented in the dataset we make sure that all molecules have at least 5 geometries that get distorted during pre-training by adding copies of the equilibrium geometry. Overall this yields 699 initial geometries R^0 for pre-training.

E Energy variance

Fig. 7 depicts the energy variance obtained by various models. Analogously to the the energies presented in Fig. 2 of the main text, we also find that our pre-trained model achieves substantially lower energy variance compared to previous generalizing wavefunctions. Notably on the test set with 3 heavy atoms, the variance of our zero-shot model is on par with the variance of a PsiFormer model trained for 16k steps.

F Relative energies

To better depict the accuracy of the models’ relative energies for the test systems in Fig. 3 of the main text, we depict the difference of relative energies between our model and a references method in Fig. 8. Furthermore we list numerical values vor all energies of Fig. 3a in Tab. 1. We find that on these

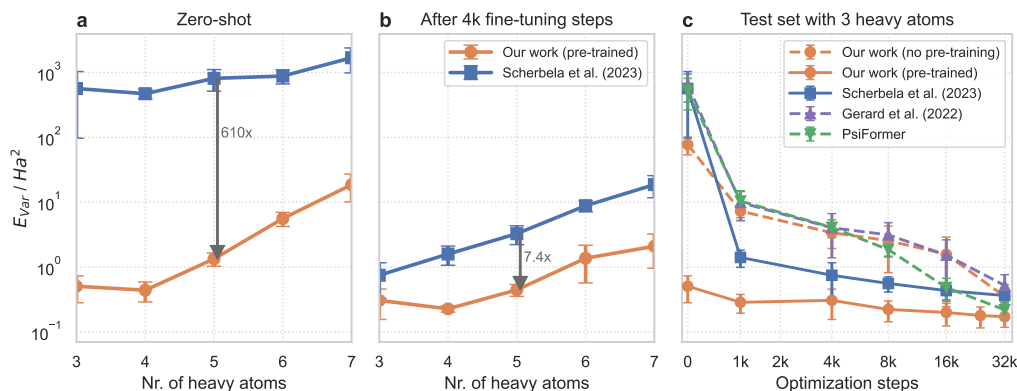


Figure 7: **Variance of energies:** Variance when re-using the pre-trained model on molecules of varying size without optimization (a) and after fine-tuning (b). For the test set containing 3 heavy atoms as a function of optimization steps, comparing against current state-of-the-art methods. Solid lines correspond to pre-trained models, dashed lines to models without pre-training (c).

challenging systems our model yields relative energy errors of up to 15 mHa when only fine-tuned for 400 steps per geometry. When fine-tuning our pre-trained model for 3200 steps per geometry, we obtain relative energies that are accurate within 2-3 mHa.

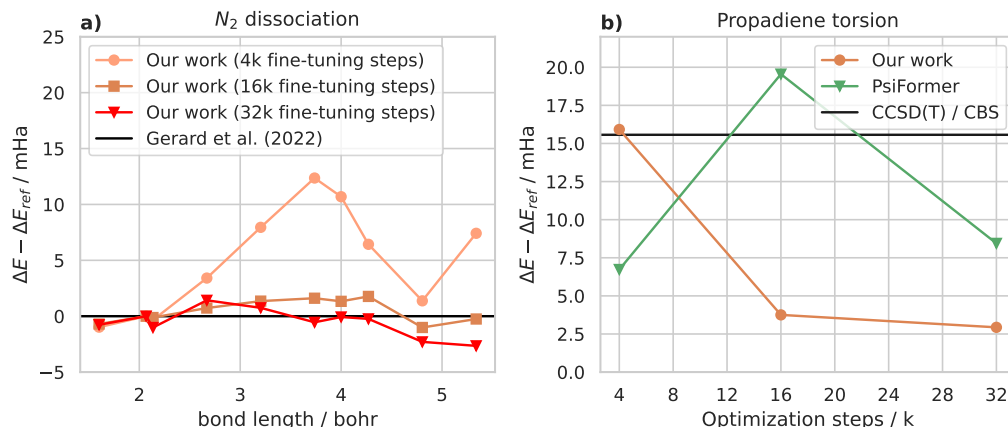


Figure 8: **Challenging relative energies:** The difference between relative energies of a reference method vs our work for two challenging systems. **a) Nitrogen dimer:** Difference of relative energies to the equilibrium geometry for our approach to reference calculation from Gerard et al. for N_2 , **a) Propadiene:** Difference of relative energies of the transition barrier of twisted vs. untwisted propadiene. As reference method we choose PsiFormer with 64k optimization steps. This figure complements Fig. 3b,d of the main text with results for additional fine-tuning steps.

G Reference energies

CCSD(T) All CCSD(T) energies – except explicitly stated otherwise – were obtained using ORCA [39] starting from a restricted Hartree-Fock calculation. We use correlation consistent basis sets of the cc-pCVXZ family, with X in {2, 3, 4}. To extrapolate to the complete basis set limit (CBS), we use

structure	CCSD(T) [31]	DMC [31]	FermiNet 200k [5]	FermiNet 10k [5]	Our work zero-shot	Our work 700 per geom
con_TS	64.4	64.4	64.1	63.9	94.0	66.6
dis_TS	34.7	93.4	92.0	87.1	183.8	94.5
g-but	-40.0	-40.2	-40.3	-44.9	-48.5	-40.4
gt-TS	-35.5	-35.4	-35.9	-42.9	-46.9	-36.7
t-but	-44.6	-44.5	-45.3	-47.5	-51.8	-43.2

Table 1: Energies relative to the energy of bicyclobutane in mHa, including the zero-point vibrational energy correction from Kinal et al. [31]. This data complements Fig. 3a in the main text.

the approach outlined in [3] and fit the following functions with free parameters $E_{\text{CBS}}^{\text{HF}}$, $E_{\text{CBS}}^{\text{corr}}$, a , b , c :

$$\begin{aligned}
E_X^{\text{HF}} &= E_{\text{CBS}}^{\text{HF}} + ae^{-bX} \\
E_X^{\text{corr}} &:= E_X^{\text{HF}} - E_X^{\text{CCSD(T)}} = E_{\text{CBS}}^{\text{corr}} + cX^{-3} \\
E_{\text{CBS}}^{\text{CCSD(T)}} &= E_{\text{CBS}}^{\text{HF}} + E_{\text{CBS}}^{\text{corr}}
\end{aligned}$$

We stress that although CCSD(T)-energies are often considered as "gold-standard", they do not necessarily represent the actual ground-state energy. There are many cases, where CCSD(T) either overestimates the true ground-state energy, or even underestimates it, because CCSD(T) does not yield upper bounds to the true ground-state energy.

PsiFormer For Fig. 2 we used the open-source FermiNet codebase [40]. The codebase didn’t allow for inference calculation, therefore a slight fix was applied. All calculations were performed with the small settings as proposed in von Glehn et al. [6].

H Hyperparameters

A detailed description of the hyperparameter used in this work can be found below (cf. Tab. 2). For the mapping of the orbital descriptors to the electron embeddings to build the orbitals we rely on the hyperparameter from [25]. For optimization we rely on the second-order method KFAC [30] and use their Python implementation [41]. During the continuous sampling of the geometries we allow each geometry to perform a maximum of 20 steps of normal-mode distortion from the initial geometry and reset to the original one once the threshold is reached.

I Runtime and computational resources

Tab. 3 lists the run-times for our method and a comparable conventional reference method (CCSD(T)-4Z) for a typical molecule with 5 heavy atoms. We find that our fine-tuning our model for 4k steps yields lower absolute energies than CCSD(T)-4Z (cf. Fig. 2) at similar computational cost. This is in contrast to many earlier works that achieve highly accurate energies, but at computational cost that far surpasses the cost of conventional methods for small molecules.

Overall we used $\approx 5\text{k GPUhs}$ (A100) for development and training of our base models, and another 5k GPUhs (A40) on evaluations and fine-tuning. Additionally we required $\approx 20\text{k CPUhs}$ for CCSD(T) reference calculations.

Method	Runtime / h
Our work, zero-shot	1.0
Our work, 4k fine-tuning	9.5
PsiFormer, 16k steps	53.0
CCSD(T), 4Z basis	10.5

Table 3: **Runtime** of various quantum chemistry methods for C_4OH_6 . All timings are in node hours, being 2 A100-hours for our work and PsiFormer, and 256 CPU-hours for CCSD(T).

J Code and data availability

All code, configuration files, geometries, datasets and obtained energies are available on GitHub under <https://github.com/mdsunivie/deeperwin>. Model weights are available on figshare under <https://doi.org/10.6084/m9.figshare.23585358.v1>.

Electron Embedding	Hidden dimension N_{emb}	256
	N ^o iterations	4
Nuclear Embedding	Hidden dimension $\tilde{x}^{nuc}$	64
	N ^o layer MLP	1
Message passing	Activation function	SiLU
	N ^o layer edge embedding	3
	Dimension edge embedding	64
	Dimension linear layer	32
Markov Chain Monte Carlo	N ^o walkers	2048
	N ^o decorrelation steps	50
	Target acceptance prob.	50%
PhisNet [20]	Pre-trained against basis set	STO-6G
	N ^o iterations	2
	Harmonic degree L	2
	N ^o radial basis functions	128
	Hidden dimension of x^{nuc}	128
	Distance cutoff (bohr)	30
Transferable atomic orbitals [25]	N ^o determinants N_{det}	8
	N ^o hidden layers f^{orb}	2
	Hidden dimension of f^{orb}	256
	N ^o hidden layers g^{orb}	2
	Hidden dimension g^{orb}	128
	N ^o iterations MPNN	2
	N ^o radial basis functions	16
	Hidden edge embedding dimension	32
	Hidden node embedding dimension	16
	Activation function	SiLU
Variational pre-training	Optimizer	KFAC
	Batch size	2048
	Norm constraint	3×10^{-3}
	Initial damping d_0	1
	Minimal damping d_{min}	0.001
	Damping rate decay	$d(t) = d_0 \exp(-t/20000)$
	Initial learning rate lr_0	0.1
	Learning rate decay	$lr(t) = lr_0(1 + t/6000)^{-1}$
Changes for fine-tuning	Optimization steps	128,000 - 256,000
Sampling geometries	Learning rate decay	$lr(t) = lr_0(7 + t/6000)^{-1}$
	Optimization steps	0 - 32,000
Sampling geometries	Distortion energy β	0.005 Ha
	Max age	20
	Bias towards original geometry α	0.2

Table 2: Hyperparameter settings used in this work

Appendix

Author contributions	133
--------------------------------	-----

Author Contributions

Leon Gerard and Michael Scherbela

January 2024

Leon Gerard and Michael Scherbela had the fortune of researching and publishing together on various aspects of solving the Schrödinger Equation using Deep Learning methods. Assigning credit in a joint research project is always difficult, but the listings below give a rough overview of the individual contributions to the various papers.

Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks

Authors: Michael Scherbela[†], Rafael Reisenhofer[†], Leon Gerard[†], Philipp Marquetand and Philipp Grohs

P.G., P.M. and R.R. conceived the project. M.S., L.G. and R.R. developed the detailed method. M.S. and L.G. wrote the Python code with contributions from R.R. The numerical experiments were designed and performed by M.S., L.G. and P.M. with support from R.R. R.R., M.S. and L.G. wrote the manuscript with input from P.G. and P.M. P.G. supervised the project. R.R. and P.G. obtained funding.

Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need?

Authors: Leon Gerard[†], Michael Scherbela[†], Philipp Marquetand and Philipp Grohs

P.G., P.M., M.S. and L.G. conceived the project. M.S. and L.G. developed the network architecture with equal contributions and performed the numerical experiments. M.S. developed the local invariant input features and the orbital envelope initialization. L.G. designed and ran experiments regarding the impact of physical prior knowledge. M.S. and L.G. wrote the manuscript with equal

[†] Equal contribution

contribution, with input from P.G. P.M. performed reference calculations. P.G. obtained funding and supervised the project.

Towards a transferable fermionic neural wavefunction for molecules

Authors: Michael Scherbela[†], Leon Gerard[†] and Philipp Grohs

M.S., L.G., and P.G. conceived the project. M.S. conceived and implemented the ansatz, built the dataset and designed the experiments. L.G. gave input on the ansatz and worked on implementation. M.S. and L.G. performed the experiments. M.S. and L.G. wrote the manuscript with input, supervision and funding from P.G.

Variational Monte Carlo on a Budget – Fine-tuning pre-trained Neural Wavefunctions

Authors: Michael Scherbela[†], Leon Gerard[†] and Philipp Grohs

P.G., M.S. and L.G. conceived the project. M.S. implemented, adapted and trained the PhisNet-like orbital embedding model with input from L.G. L.G. developed the electron model with input from M.S. M.S. developed the normal-mode based sampling of nuclear geometries. L.G. extended the pre-training dataset. M.S. performed reference CCSD(T) calculations. L.G. and M.S. designed and performed the numerical experiments and wrote the manuscript with equal contribution. P.G. obtained funding and supervised the project.

[†] Equal contribution