TUM School of Computation, Information and Technology
Technische Universität München

Generalized Neural Wave Functions

Nicholas David Gao

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr. Alessio Gagliardi

Prüfende der Dissertation:

1. Prof. Dr. Stephan Günnemann
2. Prof. Dr. Philipp Grohs

Die Dissertation wurde am 16.06.2025 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 26.10.2025 angenommen.

Abstract

Solving the electronic Schrödinger equation is at the heart of contemporary quantum chemistry to understand and simulate the behavior of electrons, atoms, and molecules. Its solution, the electronic wave function, allows deriving quantum mechanical properties and observables of atoms, molecules, and materials. Unfortunately, exact solutions to the Schrödinger equation are generally unavailable and only exist for the simplest systems, whereas approximate solutions require significant computational resources for high accuracy. The advent of neural network-based approximations of the electronic wave function within the variational Monte Carlo framework promised to change this by pairing unprecedented accuracy with theoretically favorable scaling in the system size. Such neural wave functions explicitly model the electronic wave function, i.e., the mapping from electronic coordinates to an amplitude. Despite the lower theoretical complexity, in practice, the computational cost remains a severe limitation in their application. This thesis presents a novel approach to amortize the cost of highly accurate neural wave functions over a series of molecular structures by training a single neural network to represent the electronic ground-state wave function of different molecules. We name this *generalized neural wave functions*. We develop generalized neural wave functions in several steps of increasing complexity and accuracy. First, we fix the nuclear charges and only vary their positions. We demonstrate that such a model can accurately represent the potential energy surface for all conformers at no loss of accuracy while reducing the computational cost an order of magnitude. Next, we generalize our approach to stable biochemical systems and, for the first time, demonstrate that a single function can represent the ground state of various molecules simultaneously. After showing feasibility, we extend it to the general case of arbitrary molecular and electronic configurations. Lastly, we present a machine-learning-driven solution to avoid numerical integrations at inference time, enabling high-resolution, high-accuracy potential energy surface scans. In summary, we present a novel approach for solving the electronic Schrödinger equation for many molecules, simultaneously reducing cost at high accuracy.

Zusammenfassung

Die Lösung der elektronischen Schrödinger-Gleichung steht im Mittelpunkt der modernen Quantenchemie, um das Verhalten von Elektronen, Atomen und Molekülen zu verstehen und zu simulieren. Ihre Lösung, die Elektronenwellenfunktion, ermöglicht die Ableitung quantenmechanischer Eigenschaften und Observablen von Atomen, Molekülen und Materialien. Leider sind exakte Lösungen der Schrödinger-Gleichung im Allgemeinen nicht verfügbar und existieren nur für die einfachsten Systeme, während angenäherte Lösungen erhebliche Rechenressourcen für eine hohe Genauigkeit erfordern. Das Aufkommen neuronaler Netzwerke zur Approximation der Elektronenwellenfunktion im Rahmen der Variations-Monte-Carlo-Methode versprach eine Veränderung, indem es beispiellose Genauigkeit mit theoretisch vorteilhafter Skalierung in der Systemgröße kombiniert. Solche neuronalen Wellenfunktionen modellieren die Elektronenwellenfunktion explizit, d. h. die Abbildung von Elektronenkoordinaten auf eine Amplitude. Trotz der geringeren theoretischen Komplexität bleibt in der Praxis der Rechenaufwand eine erhebliche Einschränkung ihrer Anwendung. In dieser Arbeit präsentieren wir einen neuartigen Ansatz, die Kosten hochgenauer neuronaler Wellenfunktionen über eine Reihe von Molekülstrukturen hinweg zu amortisieren, indem ein einziges neuronales Netzwerk trainiert wird, um die elektronische Grundzustandswellenfunktion verschiedener Moleküle zu repräsentieren. Wir nennen diese Methode *generalisierte neuronale Wellenfunktionen*. Wir entwickeln generalisierte neuronale Wellenfunktionen in mehreren Schritten mit zunehmender Komplexität und Genauigkeit. Zunächst fixieren wir die Menge der Kernladungen und variieren nur ihre Positionen. Wir zeigen, dass ein solches Modell die potenzielle Energiefläche aller Konformere präzise darstellen kann, ohne an Genauigkeit zu verlieren, und gleichzeitig die Rechenkosten um eine Größenordnung reduziert. Anschließend verallgemeinern wir unseren Ansatz auf stabile biochemische Systeme und zeigen erstmals, dass eine einzige Funktion den Grundzustand verschiedener Moleküle gleichzeitig repräsentieren kann. Nachdem wir die Machbarkeit demonstriert haben, erweitern wir ihn auf den allgemeinen Fall beliebiger molekularer und elektronischer Konfigurationen. Schließlich präsentieren wir eine maschinell lernbasierte Lösung, um numerische Integrationen zur Inferenzzeit zu vermeiden, wodurch hochauflösende und hochgenaue Scans der potenziellen Energiefläche ermöglicht werden. Zusammenfassend präsentieren wir einen neuartigen Ansatz zur Lösung der elektronischen Schrödinger-Gleichung für viele Moleküle gleichzeitig und reduzieren dabei die Kosten bei hoher Genauigkeit.

Acknowledgements

The last four years have been an incredible opportunity to develop both my personality and professional skills, and this would not have been possible without the fantastic people who accompanied me on this journey.

I would like to thank my incredible colleagues at the chair. Beyond advancing my research with your expertise and insightful discussions, you also enriched this journey on a personal level. You make DAML special. While I cherish every moment with each and every lab member, I would especially like to thank my close friends: Lukas Gosch for our personality-shaping discussions during our mountaineering adventures, Dominik Fuchsgruber for my weekly meme supply keeping me sane, Leo Schwinn for his guidance, and Eike Eberhard for constantly listening to my rambling about QMC. A special thanks to Jan Schuchardt for reading every single one of my works and improving them with his thorough feedback. Furthermore, I am grateful to Arthur Kosmala, Johannes Gasteiger, Amine Ketata, Eike Eberhard, Johanna Sommer, Bertrand Charpentier, and Tom Wollschläger for being such talented co-authors.

Naturally, my thanks extend to the students I had the pleasure of supervising: Arthur, Chendi, Baris, Gianluca, Amine, Eike, and Till. You greatly aided my research, and I learned a lot from you.

I am incredibly grateful to have pursued my PhD under Stephan Günnemann's supervision. You guided me into research, helped me find my research field, supported my ideas, and enabled me to follow my research ambitions.

I also want to express my gratitude to the many external researchers with whom I had the joy of interacting with. I am deeply thankful for my colleagues at NASA Ames Research Center, especially Max Wilson, who kickstarted my scientific journey by guiding me during my internship – without him, I would not be where I am today. Thanks also to the amazing colleagues at Microsoft: Jonas Köhler, Daniel Zügner, Jan Hermann, and Frank Noe.

Last but not least, I would also like to thank my friends, my parents, and my brother, Alexander, for their unconditional support.

Contents

Part I: Introduction and Methodology

1 Introduction

Recent years have seen a surge in the interest in *in-silico* design, i.e., the computational design of molecules and materials (Lin et al., 2012; Chanussot et al., 2021; Barroso-Luque et al., 2024). By avoiding expensive lab experiments, in-silico design promises to reduce the time and cost of developing new drugs, catalysts, and materials (Terstappen and Reggiani, 2001). This is accomplished by simulating the dynamics of atoms and molecules in different chemical environments. These simulations are primarily driven by the Potential Energy Surface (PES), a function mapping the nuclear positions and charges to the ground-state energy of the system. Access to the PES enables predicting various properties, such as geometric conformers, free energy calculations, and reaction pathways. Unfortunately, evaluating the PES at any point requires solving the time-independent Schrödinger equation, a computationally expensive task.

The Schrödinger equation is a partial differential equation describing nuclei and electrons' quantum mechanical behavior. For a fixed structure, its solution, the wave function, enables the calculation of all observables, such as the energy, dipole moment, and electron density. Despite being known for almost a century, analytical solutions are only available for the simplest systems. Further, an exact solution is known to be NP-hard (Troyer and Wiese, 2005). Consequently, a zoo of approximate methods has been developed which can be broadly categorized into three classes of decreasing accuracy: *ab-initio* methods which solve the Schrödinger equation directly from first principle, *semi-empirical* methods which are generally similar to ab-initio methods but rely on empirical parameters, and *empirical* methods which avoid solving the Schrödinger altogether by fitting an empirical potential to reference calculations. Ab-initio methods include mean-field methods like Hartree-Fock (Hartree, 1928; Fock, 1930), to post-Hartree-Fock methods like coupled cluster theory (Čížek, 1966), and Quantum Monte Carlo (QMC) (Ceperley et al., 1977). Density functional theory is the prominent representative of semi-empirical methods (Hohenberg and Kohn, 1964; Kohn and Sham, 1965), whereas empirical methods include empirically fitted molecular force fields (Halgren, 1996). As these methods differ in accuracy and computational cost, finding the best trade-off is the central challenge of quantum chemistry. For instance, while density functional theory is widely used due to its favorable computational cost, it is known to have systematic errors (Cohen et al., 2008). Conversely, accurate multireference methods are computationally expensive, scale poorly with system size, and require expert knowledge to apply correctly (Szabo and Ostlund, 2012).

The field of numerical solutions has recently been disrupted by neural network-based approaches (Carleo and Troyer, 2017). While historically touted as a low-accuracy

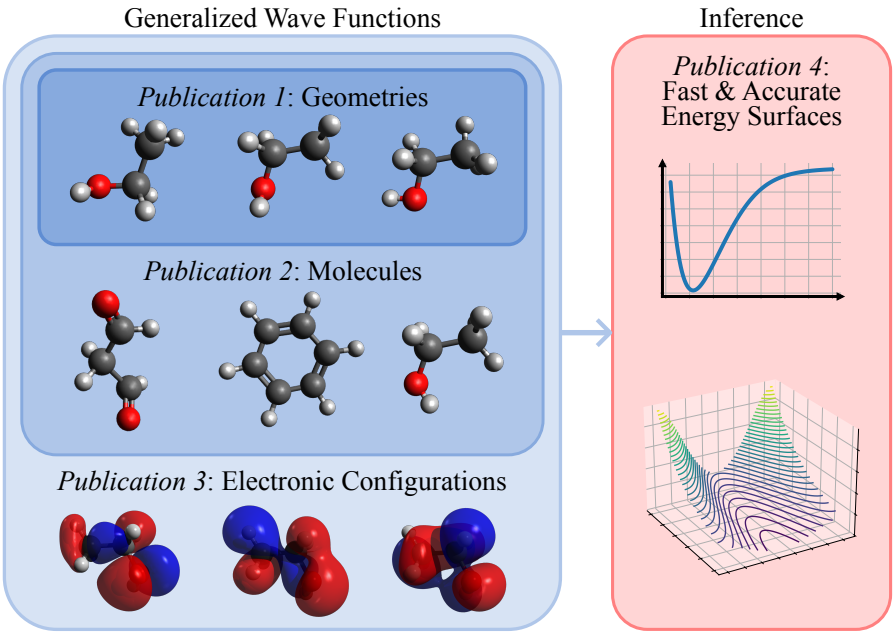


Figure 1.1: Contributions of this thesis. In *Publication 1*, we tackle generalized wave functions for different conformers of the same molecule. In *Publication 2*, we generalize over different stable biochemical molecules. In *Publication 3*, we extend our approach to arbitrary molecular structures and electronic configurations. Finally, in *Publication 4*, we propose a fast and accurate inference method.

method, Variational Monte Carlo (VMC) (Ceperley et al., 1977), a flavor of QMC, has accomplished unprecedented accuracy by approximating the electronic wave function with deep neural networks (Pfau et al., 2019; Hermann et al., 2020). In neural network-based VMC, one selects a parametrized trial neural-network wave function and optimizes its parameters to minimize the energy until convergence. These neural network-based methods promise lower computational complexity, higher accuracy, simple application, and benefits from the advancements made in deep learning (Krizhevsky et al., 2012).

Traditionally, given any of the above methods, the PES can be obtained by scanning the nuclear coordinates and solving the electronic Schrödinger equation for each geometry separately. By fitting an empirical potential to a set of reference calculations, one can accelerate this by interpolating between them (Jiang and Guo, 2013; Schütt et al., 2017). This standard approach has severe limitations as it requires solving the Schrödinger equation for many systems, growing the computational cost linearly with the number of geometries. Due to the exponential nature of growing search space, this approach is prohibitive for large molecules or high-dimensional PESs.

In this thesis, we propose a novel approach to obtaining highly accurate PESs for molecular systems via our *generalized neural wave functions*. Concretely, we propose to leverage the flexible deep learning paradigm to learn a wave function that generalizes across molecules and conformers within the VMC framework. Such a wave function can be used to

evaluate the PES at any geometry without resolving the electronic Schrödinger equation. We approach this problem in several steps, outlined by the four core publications of this thesis, which we visualize in Figure 1.1. First, we tackle the simplified problem of learning the PES for a single molecule, i.e., a fixed set of nuclei that only differ in their positions. Second, we generalize this approach to generalize over different stable biochemical molecules. Third, we propose a general solution for arbitrary molecular structures and electronic configurations. Finally, we propose a sampling-free inference method that enables the evaluation of the PES orders of magnitude faster than relying on traditional numerical integration.

1.1 Structure and Outline

The thesis is structured in three parts: Part I introduces the problem, provides the necessary background, and outlines our contributions. There, we start in Chapter 2 with an introduction to classic quantum chemistry, the electronic Schrödinger equation, variational methods, Hartree Fock, the variational Monte Carlo algorithm, and the concept of the potential energy surface. In Chapter 3, we discuss the application of machine learning to quantum chemistry. Specifically, we introduce the concept of equivariant deep learning central to the following sections on machine-learned force fields, empirical approximations of the PES, and neural-network wave functions in *ab-initio* quantum chemistry. Finally, in Chapter 4, we introduce the main contribution of this thesis, generalized neural wave functions. We outline the desiderata for a generalized wave function and our works approaching this goal. In Part II, we present the four core publications of this thesis with a separate chapter for each. Finally, in Part III, we retrospectively review the impact of our works and conclude with an outlook on future work. Before diving into the main content, the next section, Section 1.2, provides an index of the publications that are part of this thesis, followed by a list of additional publications by the author in Section 1.3. Lastly, we provide a notation guide in Section 1.4.

1.2 Index of Publications

This thesis comprises four "core publications", listed below. All are published at top-tier conferences in the field of machine learning. Publication 1 has been selected as a spotlight, and Publication 3 as an oral presentation at their respective conferences.

Publication 1: Nicholas Gao and Stephan Günnemann. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, 2022

Publication 2: Nicholas Gao and Stephan Günnemann. Generalizing Neural Wave Functions. In *International Conference on Machine Learning*, 2023a

Publication 3: Nicholas Gao and Stephan Günnemann. Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a

Publication 4: Nicholas Gao and Stephan Günnemann. Sampling-free Inference for Ab-Initio Potential Energy Surface Networks. In *The Eleventh International Conference on Learning Representations*, 2023b

1.3 Other Publications by the Author

Below, we list additional publications by the author that were published during the PhD but are not part of this thesis.

Publication 5: Max Wilson, Nicholas Gao, Filip Wudarski, Eleanor Rieffel, and Norm M. Tubman. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo, 2021

Publication 6: Nicola Franco, Tom Wollschläger, Nicholas Gao, Jeanette Miriam Lorenz, and Stephan Günnemann. Quantum Robustness Verification: A Hybrid Quantum-Classical Neural Network Certification Algorithm. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 142–153, 2022

Publication 7: Max Wilson, Saverio Moroni, Markus Holzmann, Nicholas Gao, Filip Wudarski, Tejs Vegge, and Arghya Bhowmik. Neural network ansatz for periodic wave functions and the homogeneous electron gas. *Physical Review B*, 107(23): 235139, 2023

Publication 8: Arthur Kosmala, Johannes Gasteiger, Nicholas Gao, and Stephan Günnemann. Ewald-based Long-Range Message Passing for Molecular Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17544–17563. PMLR, 2023

Publication 9: Tom Wollschläger, Nicholas Gao, Bertrand Charpentier, Mohamed Amine Ketata, and Stephan Günnemann. Uncertainty Estimation for Molecules: Desiderata and Methods. In *Proceedings of the 40th International Conference on Machine Learning*, pages 37133–37156. PMLR, 2023

Publication 10: Xuan Zhang, 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. Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems, 2023a

Publication 11: Mohamed Amine Ketata, Nicholas Gao, Johanna Sommer, Tom Wollschläger, and Stephan Günnemann. Lift Your Molecules: Molecular Graph Generation in Latent Euclidean Space. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024

Publication 12: Nicholas Gao and Stephan Günnemann. On Representing Electronic Wave Functions with Sign Equivariant Neural Networks. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024b

Publication 13: Nicholas Gao, Eike Eberhard, and Stephan Günnemann. Learning Equivariant Non-Local Electron Density Functionals. In *The Thirteenth International Conference on Learning Representations*, 2025

Publication 14: Mohamed Amine Ketata, Nicholas Gao, Johanna Sommer, Tom Wollschläger, and Stephan Günnemann. Lift Your Molecules: Molecular Graph Generation in Latent Euclidean Space. In *The Thirteenth International Conference on Learning Representations*, 2025

Publication 15: Artur P. Toshev, Teodor Kalinov, Nicholas Gao, Stephan Günnemann, and Nikolaus A. Adams. On Learning Quasi-Lagrangian Turbulence. In *ICLR 2025 Workshop on Machine Learning Multiscale Processes*, 2025

1.4 Notation

This work generally follows standard machine learning notation and uses bold symbols to denote vectors (lower case) $\boldsymbol{v}$ and matrices (upper case) $\boldsymbol{A}$. Their respective elements are denoted by the same symbol without boldface, e.g., v_i and A_{ij}. Due to their special role in this thesis, we denote the set of N_e electrons as $\vec{\mathbf{r}} \in \mathbb{R}^{N_\mathrm{e} \times 3}$ and a single electron as $\vec{r} \in \mathbb{R}^3$. Similarly, we denote the set of N_n nuclei as $\vec{\mathbf{R}} \in \mathbb{R}^{N_\mathrm{n} \times 3}$ and a single nucleus as $\vec{R} \in \mathbb{R}^3$. Where appropriate, we use the Braket notation to denote inner products, $\langle f|g\rangle = \int f(x)g(x)\mathrm{d}x$ in infinite Hilbert spaces (Dirac, 1939). In Braket, operators act on the Ket, e.g., $\nabla_x |f\rangle = \nabla_x f$ denotes the gradient of the function f with respect to x. If ambiguous, we use parentheses if we are interested in the value of an operator at a specific point, e.g., $[\nabla_x f](x)$. We use $\mathrm{J}_f(x), \mathrm{H}_f(x)$ to denote the Jacobian and Hessian of a function f at x. We further denote the partial of a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, i.e., the function with some arguments fixed, via $f(\cdot|y) : \mathcal{X} \to \mathcal{Z}$ for $y \in \mathcal{Y}$. The explicit dependence on the variational parameters is frequently omitted and implicitly assumed.

2 Quantum Chemistry

The goal of quantum chemistry is to solve the electronic Schrödinger equation. To this end, a few simplifications are typically made: (1) One looks at the time-independent Schrödinger equation, i.e., one neglects the system's dynamics and only considers the ground state as it strongly indicates a molecule's stability, structure, and properties (Schrödinger, 1926). (2) One considers the nuclei fixed relative to the electrons due to their order-of-magnitude different speeds. This is called the Born-Oppenheimer approximation (Born and Heisenberg, 1985). The resulting time-independent Schrödinger equation in the Born-Oppenheimer approximation is defined by the positions $\vec{\mathbf{R}} \in \mathbb{R}^{N_\mathrm{n} \times 3}$ and charges $\mathbf{Z} \in \mathbb{N}_+^{N_\mathrm{n}}$ of the nuclei. It is a second-order partial differential equation for the wave function, $\Psi : \left\{\mathbb{R}^3 \times \{\uparrow, \downarrow\}\right\}^{N_\mathrm{e}} \to \mathbb{R}$, of the N_e electrons:

$$\hat{H} \left|\Psi\right\rangle = E \left|\Psi\right\rangle \tag{2.1}$$

where $\hat{H} : L^2 \to L^2$ is the Hamiltonian operator and E is the energy. In atomic units, the Hamiltonian operator $\hat{H}$, that maps square-integrable functions to square-integrable functions $L^2 = \{f \mid \int |f(x)|^2 \mathrm{d}x < \infty\}$,

$$\hat{H} := -\frac{1}{2} \underbrace{\sum_{i=1}^{N_\mathrm{e}} \sum_{k=1}^{3} \frac{\partial^2}{\partial \vec{r}_{ik}^2}}_{\nabla^2} + \underbrace{\sum_{j>i}^{N_\mathrm{e}} \frac{1}{\|\vec{r}_i - \vec{r}_j\|} - \sum_{i=1}^{N_\mathrm{e}} \sum_{m=1}^{N_\mathrm{n}} \frac{Z_m}{\|\vec{r}_i - \vec{R}_m\|} + \sum_{n>m}^{N_\mathrm{n}} \frac{Z_m Z_n}{\|\vec{R}_m - \vec{R}_n\|}}_{V(\vec{\mathbf{r}})} \tag{2.2}$$

is given as a sum of the kinetic energy operator, ∇^2, and the potential energy, $V(\vec{\mathbf{r}})$. The potential energy is the sum of the electron-electron repulsion, the electron-nucleus attraction, and the nucleus-nucleus repulsion. The electron-nuclei and nuclei-nuclei interactions are the so-called external potential and vary with the number N_n, positions $\vec{\mathbf{R}}$, and charges $\mathbf{Z}$ of the nuclei. Thus, different molecules and conformations correspond to Hamiltonians that differ in their external potential. As $\hat{H}$ is hermitian, the eigenfunctions Ψ satisfying Equation 2.1 have a constant phase and, thus, can be chosen to be real-valued.

As electrons are indistinguishable fermions, each has a half-integer spin $\uparrow$ / $\downarrow$, and the wave function Ψ must be antisymmetric with respect to the exchange of any same-spin two electrons (Dirac and Fowler, 1997):

$$\forall \tau^\uparrow \in S_{N_\uparrow}, \tau^\downarrow \in S_{N_\downarrow} : \Psi\left(\tau^\uparrow(\vec{\mathbf{r}}^\uparrow), \tau^\downarrow(\vec{\mathbf{r}}^\downarrow)\right) = \mathrm{sgn}(\tau^\uparrow)\mathrm{sgn}(\tau^\downarrow)\Psi(\vec{\mathbf{r}}) \tag{2.3}$$

where S_N denotes the symmetric group of N elements and sgn the sign of a permutation, i.e., the parity of the number of pairwise exchanges required to obtain the permutation.

As the spin does not occur in the Hamiltonian, the spin can be fixed a priori (Szabo and Ostlund, 2012). Further, thanks to the Aufbau principle, the spin configuration of the ground state is known for most molecules (Bohr, 1921). Thus, we will frequently omit the spin variable in the wave function and treat the wave function as $\Psi : \mathbb{R}^{N_\mathrm{e} \times 3} \to \mathbb{R}$ with the first $N_\uparrow$ electrons being spin-up and the remaining $N_\downarrow = N_\mathrm{e} - N_\uparrow$ being spin-down.

In linear algebra, Equation 2.1 is an eigenvalue equation where we seek the eigenvalues E and eigenfunctions Ψ. However, not all are equally important as most of quantum chemistry concerns the lowest eigenvalue E_0, the ground-state energy, and its corresponding eigenfunction Ψ_0, the ground-state wave function. The ground state provides information about a molecule's most stable state and electronic structure, such as bond lengths, angles, and electronic density. Higher eigenvalues correspond to excited states, which inform electronic properties such as light absorption and emission.

2.1 Variational Wave Function Ansatz

An important concept in solving the Schrödinger equation is the variational principle. It states that the energy of any wave function Ψ upper bounds the ground-state energy E_0. Let $\Psi = \sum_i \alpha_i \Psi_i, \alpha_i \in \mathbb{R}$ be a linear combination of the eigenfunctions Ψ_i, one can define the so-called variational energy as (Heitler and London, 1927)

$$E\left[\Psi\right] := \frac{\langle\Psi|\,\hat{H}\,|\Psi\rangle}{\langle\Psi|\Psi\rangle}. \tag{2.4}$$

This variational energy upper bounds the ground-state energy E_0:

$$\begin{aligned} E\left[\Psi\right] =& \frac{\langle\Psi|\,\hat{H}\,|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \sum_{i,j} \alpha_i \alpha_j \frac{\langle\Psi_i|\,\hat{H}\,|\Psi_j\rangle}{\langle\Psi_i|\Psi_j\rangle} \\ =& \frac{1}{\langle\Psi|\Psi\rangle} \sum_{i,j} \alpha_i \alpha_j \langle\Psi_i|\,\hat{H}\,|\Psi_j\rangle = \frac{1}{\langle\Psi|\Psi\rangle} \sum_{i,j} \alpha_i \alpha_j \langle\Psi_i|\,E_j\,|\Psi_j\rangle \\ =& \frac{1}{\langle\Psi|\Psi\rangle} \sum_{i,j} \alpha_i \alpha_j E_j \delta_{ij} = \frac{1}{\langle\Psi|\Psi\rangle} \sum_i \alpha_i^2 E_i \\ =& \frac{1}{\sum_i \alpha_i^2} \sum_i \alpha_i^2 E_i \geq \frac{1}{\sum_i \alpha_i^2} \sum_i \alpha_i^2 E_0 = E_0. \end{aligned} \tag{2.5}$$

This variational principle is the foundation of many quantum chemistry methods. It allows for constructing approximate wave functions and their variational optimization towards the ground state E_0 by minimizing the energy $E[\Psi]$. A common approach to implement this by using a parametrized wave function Ψ_θ with parameters θ where one solves the optimization problem

$$\theta^* = \arg\min_\theta E\left[\Psi_\theta\right]. \tag{2.6}$$

The accuracy of such a method typically depends on the flexibility of the wave function ansatz and the optimization algorithm.

2.2 Hartree Fock

The simplest non-trivial wave function Ansatz is likely the Hartree-Fock (HF) method (Hartree, 1928; Fock, 1930). In HF, the wave function is defined as a product of two so-called Slater determinants, one for the spin-up electrons and one for the spin-down electrons (Slater, 1929):

$$\begin{aligned}\Psi_{\mathrm{HF}}(\vec{\mathbf{r}}) &= \det \begin{pmatrix} \phi_1\left(\vec{r}_1\right) & \cdots & \phi_1\left(\vec{r}_{N_\uparrow}\right) \\ \vdots & \ddots & \vdots \\ \phi_{N_\uparrow}\left(\vec{r}_1\right) & \cdots & \phi_{N_\uparrow}\left(\vec{r}_{N_\uparrow}\right) \end{pmatrix} \det \begin{pmatrix} \phi_{N_\uparrow+1}\left(\vec{r}_{N_\uparrow+1}\right) & \cdots & \phi_{N_\uparrow+1}\left(\vec{r}_{N_\mathrm{e}}\right) \\ \vdots & \ddots & \vdots \\ \phi_{N_\mathrm{e}}\left(\vec{r}_{N_\uparrow+1}\right) & \cdots & \phi_{N_\mathrm{e}}\left(\vec{r}_{N_\mathrm{e}}\right) \end{pmatrix} \\ &= \det \Phi^{\uparrow}(\vec{\mathbf{r}}^{\uparrow}) \det \Phi^{\downarrow}(\vec{\mathbf{r}}^{\downarrow}).\end{aligned} \tag{2.7}$$

Here, $\phi_i : \mathbb{R}^3 \to \mathbb{R}$ are so-called the *molecular orbitals* (MOs) (Slater, 1929). The separate determinants ensure that the antisymmetry only holds with respect to the exchange of electrons with the same spin.

Each MO is defined as a linear combination of a fixed set of basis functions, the so-called *atomic orbitals*, $\chi_\mu : \mathbb{R}^3 \to \mathbb{R}, \mu \in \{1, \ldots, N_{\mathrm{bas}}\}$:

$$\phi_j(\vec{r}) = \sum_\mu C_{\mu j} \chi_\mu(\vec{r}) \tag{2.8}$$

with the MO coefficients $\boldsymbol{C} \in \mathbb{R}^{N_{\mathrm{bas}} \times N_{\mathrm{bas}}}$. While this discretization in atomic orbitals is a significant constraint on the flexibility of the wave function, it allows for the analytical computation of the integrals in the Hamiltonian (Equation 2.2) and the optimization of the MO coefficients $\boldsymbol{C}$ in $O({N_{\mathrm{bas}}}^4)$ via the Roothaan-Hall equations (Roothaan, 1951; Hall, 1951).

The HF method generally includes two types of errors, the *basis set error* and the *mean-field error*. The first arises from the finite basis set and can be systematically reduced by increasing the number of basis functions N_{bas}. The second arises from the mean-field approximation, where the electrons are assumed to move independently of each other, i.e., ϕ_j being single-electron functions. This approximation neglects electron correlation, the interaction between electrons due to their Coulomb repulsion. The correlation energy is the difference between the exact energy and the *complete basis set* (CBS) HF energy limit $E_{\mathrm{HF\ (CBS)}}$ (Dunning, 1989):

$$E_{\mathrm{corr}} = E_0 - E_{\mathrm{HF\ (CBS)}} \tag{2.9}$$

and is the target of post-HF methods (Moller and Plesset, 1934; Lyakh et al., 2012).

While HF is frequently qualitatively correct, especially for closed-shell systems, it is quantitatively inaccurate. Nonetheless, it is frequently the starting point for more accurate post-HF methods like coupled cluster theory (Čížek, 1966) or quantum Monte Carlo (QMC) (Ceperley et al., 1977) as the mean-field energy constitutes $> 99\%$ of the total energy in most cases (Szabo and Ostlund, 2012).

2.3 Variational Quantum Monte Carlo

The post-HF method this thesis focuses on is Variational Quantum Monte Carlo (VMC) (Ceperley et al., 1977), which we illustrate in Figure 2.1. As the name suggests, VMC combines the variational principle with Monte Carlo sampling to optimize the wave function parameters θ in Equation 2.6. Concretely, one can rewrite the energy in Equation 2.4 as an expectation value over the probability density $\rho_\Psi(\vec{\mathbf{r}}) = \frac{\Psi(\vec{\mathbf{r}})^2}{\langle\Psi|\Psi\rangle}$:

$$\begin{aligned} E\left[\Psi\right] = \frac{\langle\Psi|\,\hat{H}\,|\Psi\rangle}{\langle\Psi|\Psi\rangle} &= \frac{1}{\langle\Psi|\Psi\rangle}\int \Psi(\vec{\mathbf{r}})[\hat{H}\Psi](\vec{\mathbf{r}})\mathrm{d}\vec{\mathbf{r}} \\ &= \frac{1}{\langle\Psi|\Psi\rangle}\int \Psi(\vec{\mathbf{r}})\frac{\Psi(\vec{\mathbf{r}})}{\Psi(\vec{\mathbf{r}})}[\hat{H}\Psi](\vec{\mathbf{r}})\mathrm{d}\vec{\mathbf{r}} \\ &= \int \frac{\Psi(\vec{\mathbf{r}})^2}{\langle\Psi|\Psi\rangle}\Psi(\vec{\mathbf{r}})^{-1}[\hat{H}\Psi](\vec{\mathbf{r}})\mathrm{d}\vec{\mathbf{r}} \\ &= \int \Psi(\vec{\mathbf{r}})^{-1}[\hat{H}\Psi](\vec{\mathbf{r}})\mathrm{d}\rho_\Psi(\vec{\mathbf{r}}) \\ &= \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\underbrace{\Psi(\vec{\mathbf{r}})^{-1}[\hat{H}\Psi](\vec{\mathbf{r}})}_{E_L(\vec{\mathbf{r}})}\right] \end{aligned} \tag{2.10}$$

where $E_L : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$ is the so-called *local energy*. We further describe the local energy computation in Section 2.3.1. The expectation value in Equation 2.10 can be approximated by sampling N_walker configurations $\vec{\mathbf{r}}_i \sim \rho_\Psi$ from the probability density ρ_Ψ:

$$E\left[\Psi\right] \approx \frac{1}{N_\mathrm{walker}}\sum_{i=1}^{N_\mathrm{walker}} E_L(\vec{\mathbf{r}}_i). \tag{2.11}$$

Conveniently, to compute the energy in Equation 2.11, the wave function does not need to be normalized; one only needs samples from the probability distribution ρ_Ψ. These are typically obtained by techniques that allow sampling from the unnormalized probability distribution, Ψ^2. Most commonly, Metropolis-Hastings algorithm (Metropolis et al., 1953) or its derivates like Langevin Monte Carlo (Besag, 2004) are used. We cover these in Section 2.3.2.

As VMC puts no additional constraints on the wave function, closed-form minimization of Equation 2.10 is typically infeasible. Instead, one relies on gradient-based optimization algorithms to minimize the energy with respect to the wave function parameters θ:

$$\theta^{t+1} = \theta^t - \eta\boldsymbol{P}^{-1}\nabla_{\theta^t}E\left[\Psi\right] \tag{2.12}$$

where $\boldsymbol{P} \in \mathbb{R}^{N_\mathrm{param}\times N_\mathrm{param}}$ is the preconditioner and $\eta \in \mathbb{R}_+$ is the learning rate. For instance, the identity $\boldsymbol{P} = I$ yields the standard stochastic gradient descent algorithm and the Hessian $\boldsymbol{P} = \mathrm{H}_{E[\Psi]}$ the Newton method. In practice, one uses quasi-Newton methods

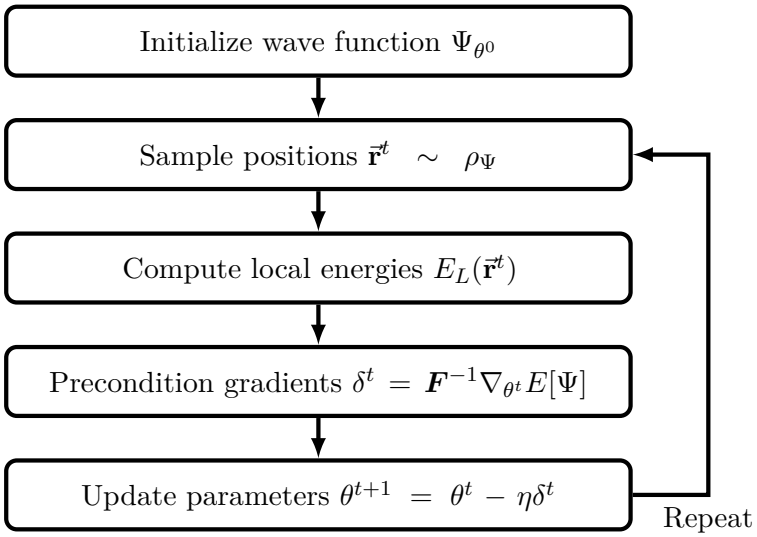


Figure 2.1: The Variational Quantum Monte Carlo (VMC) algorithm. One starts with an initial wave function Ψ_{θ^0}. Then, one iteratively refines the wave function's parameters be minimizing the energy that is approximated via importance sampling with samples from $\Psi^2_{\theta^t}$.

like Stochastic Reconfiguration (SR) (Sorella, 1998) and its derivatives like the Kronecker-factorized approximate curvature (KFAC) (Martens and Grosse, 2015) where $\boldsymbol{P}$ is an approximation of the Fisher information matrix $\boldsymbol{F} = \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\left(\nabla_\theta \ln \rho_\Psi(\vec{\mathbf{r}})\right)\left(\nabla_\theta \ln \rho_\Psi(\vec{\mathbf{r}})\right)^T\right]$. We cover these in Section 2.3.3. The gradient of the energy with respect to the wave function parameters θ can be computed as

$$\begin{aligned}
\nabla_\theta E[\Psi] &= \nabla_\theta \int \rho_\Psi(\vec{\mathbf{r}}) E_L(\vec{\mathbf{r}}) \mathrm{d}\vec{\mathbf{r}} = \int \underbrace{\nabla_\theta \rho_\Psi(\vec{\mathbf{r}})}_{(1)} E_L(\vec{\mathbf{r}}) \mathrm{d}\vec{\mathbf{r}} + \underbrace{\int \rho_\Psi(\vec{\mathbf{r}}) \nabla_\theta E_L(\vec{\mathbf{r}}) \mathrm{d}\vec{\mathbf{r}}}_{(2)} \\
&= \int 2\rho_\Psi(\vec{\mathbf{r}}) \left(\nabla_\theta \ln |\Psi(\vec{\mathbf{r}})| - \mathbb{E}_{\vec{\mathbf{r}}'\sim\rho_\Psi}\left[\nabla_\theta \ln |\Psi(\vec{\mathbf{r}}')|\right]\right) E_L(\vec{\mathbf{r}}) \mathrm{d}\vec{\mathbf{r}} \\
&= 2\mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\left(E_L(\vec{\mathbf{r}}) - E[\Psi]\right) \nabla_\theta \ln |\Psi(\vec{\mathbf{r}})|\right]
\end{aligned} \tag{2.13}$$

which we obtain if we use the identity $\nabla_\theta \Psi(\vec{\mathbf{r}}) = \Psi(\vec{\mathbf{r}}) \nabla_\theta \ln |\Psi(\vec{\mathbf{r}})|$ on (1)

$$\begin{aligned}
\nabla_\theta \rho_\Psi(\vec{\mathbf{r}}) &= \nabla_\theta \frac{\Psi^2(\vec{\mathbf{r}})}{\langle\Psi|\Psi\rangle} \\
&= \frac{1}{\langle\Psi|\Psi\rangle} \nabla_\theta \Psi^2(\vec{\mathbf{r}}) - \frac{\Psi^2(\vec{\mathbf{r}})}{\langle\Psi|\Psi\rangle^2} \nabla_\theta \langle\Psi|\Psi\rangle \\
&= 2\left(\frac{\Psi^2(\vec{\mathbf{r}})}{\langle\Psi|\Psi\rangle} \nabla_\theta \ln |\Psi(\vec{\mathbf{r}})| - \frac{\Psi^2(\vec{\mathbf{r}})}{\langle\Psi|\Psi\rangle^2} \langle\Psi|\Psi\nabla_\theta \ln |\Psi|\rangle\right) \\
&= 2\rho_\Psi(\vec{\mathbf{r}}) \left(\nabla_\theta \ln |\Psi(\vec{\mathbf{r}})| - \mathbb{E}_{\vec{\mathbf{r}}'\sim\rho_\Psi}\left[\nabla_\theta \ln |\Psi(\vec{\mathbf{r}}')|\right]\right).
\end{aligned} \tag{2.14}$$

For (2), we can first evaluate the gradient of the local energy where we use that the Hamiltonian $\hat{H}$ is independent of θ and commutes with ∇_θ:

$$\begin{aligned}\nabla_\theta E_L(\vec{\mathbf{r}}) &= \nabla_\theta \left(\Psi(\vec{\mathbf{r}})^{-1}\left[\hat{H}\Psi\right](\vec{\mathbf{r}})\right) \\ &= \Psi(\vec{\mathbf{r}})^{-1}\left[\hat{H}\nabla_\theta\Psi\right](\vec{\mathbf{r}}) - \Psi(\vec{\mathbf{r}})^{-2}\nabla_\theta\Psi(\vec{\mathbf{r}})\left[\hat{H}\Psi\right](\vec{\mathbf{r}}) \\ &= \Psi(\vec{\mathbf{r}})^{-1}\left[\hat{H}\left(\Psi\nabla_\theta \ln|\Psi|\right)\right](\vec{\mathbf{r}}) - \Psi(\vec{\mathbf{r}})^{-1}\nabla_\theta \ln|\Psi(\vec{\mathbf{r}})|\left[\hat{H}\Psi\right](\vec{\mathbf{r}}),\end{aligned} \tag{2.15}$$

$$\langle\rho_\Psi|\nabla_\theta E_L\rangle = \frac{\langle\Psi^2|\nabla_\theta E_L\rangle}{\langle\Psi|\Psi\rangle} = \frac{\left(\langle\Psi|\hat{H}|\Psi\nabla_\theta \ln|\Psi|\rangle - \langle\Psi\nabla_\theta \ln|\Psi||\hat{H}|\Psi\rangle\right)}{\langle\Psi|\Psi\rangle} = 0. \tag{2.16}$$

In Equation 2.16, we used the fact that the Hamiltonian $\hat{H}$ is hermitian, i.e., $\langle f|\hat{H}|g\rangle = \langle g|\hat{H}|f\rangle$. This efficient computation of the energy gradient is a key advantage of VMC as it allows for the flexible parametrization of the wave function and does not require differentiation through the local energy computation, avoiding third-order differentials (Ceperley et al., 1977).

Instead of minimizing the energy, one could also optimize the variance of the local energy $\mathrm{Var}[E_L] = \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\left(E_L(\vec{\mathbf{r}}) - \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[E_L(\vec{\mathbf{r}})\right]\right)^2\right]$ which is 0 for all eigenfunctions of the Hamiltonian (Umrigar et al., 1988). While minimizing the variance of the local energy used to be the standard procedure in traditional VMC (Foulkes et al., 2001), it requires computing derivatives through the local energy computation (Bueckert et al., 1992) rendering it costly in neural-network VMC. Further, it has been shown that variance minimization may lead to incorrect energies or higher energies, which energy minimization avoids (Cuzzocrea et al., 2020). Thus, energy minimization has become the standard in neural-network VMC (Hermann et al., 2020) with various modifications to improve convergence (Neklyudov et al., 2023; Su and Liu, 2024).

As VMC only requires differentiable wave functions for optimization, only the physical constraints of antisymmetry and square integrability must be enforced. Thus, to enhance the flexibility beyond the simple Slater wave function in Equation 2.7, one typically defined as a product of an antisymmetric part, typically realized as a linear combination of Slater determinants, and a symmetric part, the so-called *Jastrow factor* $\mathcal{J} : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$ (Jastrow, 1955):

$$\Psi(\vec{\mathbf{r}}) = \exp\left(\mathcal{J}(\vec{\mathbf{r}})\right) \cdot \sum_{k=1}^{K} \det\Phi_k^{\uparrow}(\vec{\mathbf{r}}^{\uparrow}) \det\Phi_k^{\downarrow}(\vec{\mathbf{r}}^{\downarrow}). \tag{2.17}$$

The orbital functions, may further be augmented to depend on all electron positions, i.e., $\Phi_k^\alpha(\vec{\mathbf{r}}^\alpha)_{ij} = \phi_i^\alpha(\vec{r}_j^{\,\alpha}|\vec{\mathbf{r}}^\alpha, \vec{\mathbf{r}}^{\tilde{\alpha}})$ where $\alpha \in \{\uparrow,\downarrow\}$ and $\tilde{\alpha} = \downarrow$ if $\alpha = \uparrow$ and vice versa (Feynman and Cohen, 1956; Luo and Clark, 2019). Though, other functional forms are very well possible, e.g., Pfaffians (Bajdich et al., 2006, 2008), antisymmetric geminal powers (AGP) (Casula and Sorella, 2003; Casula et al., 2004), pairwise antisymmetric function (Han et al., 2019), or Vandermonde determinants (Acevedo et al., 2020).

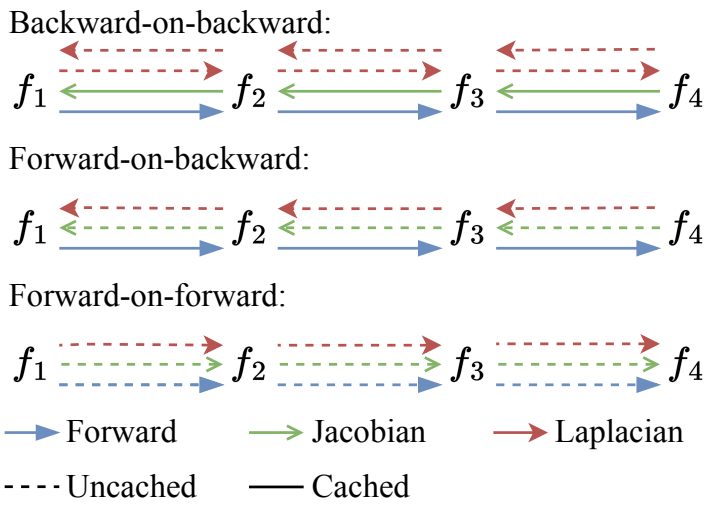


Figure 2.2: Comparison of different ways to compute the Laplacian of a composite function $f_1 \circ \ldots \circ f_4$. Solid lines denote quantities that must be cached for backpropagation, while dashed lines denote quantities that can be freed after computation.

2.3.1 Local Energy Computation

Computing the local energy $E_L(\vec{\mathbf{r}})$ is one of VMC's most computationally expensive parts. While one could naively compute $E_L(\vec{\mathbf{r}})$ directly via $\Psi(\vec{\mathbf{r}})^{-1}(\vec{\mathbf{r}})[\hat{H}\Psi](\vec{\mathbf{r}})$, this is sensitive to numerical instabilities. Instead, we can reformulate the local energy in terms of the logarithm of the wave function:

$$\begin{aligned} E_L(\vec{\mathbf{r}}) &= \Psi^{-1}(\vec{\mathbf{r}})[\hat{H}\Psi](\vec{\mathbf{r}}) \\ &= -\frac{1}{2}\Psi^{-1}(\vec{\mathbf{r}})[\nabla^2\Psi](\vec{\mathbf{r}}) + V(\vec{\mathbf{r}}) \\ &= \underbrace{-\frac{1}{2}\left(\nabla^2 \ln|\Psi(\vec{\mathbf{r}})| + (\nabla \ln|\Psi(\vec{\mathbf{r}})|)^2\right)}_{\text{kinetic energy}} + \underbrace{V(\vec{\mathbf{r}})}_{\text{potential energy}} . \end{aligned} \tag{2.18}$$

where we use the following reformulation of the kinetic energy in the last step:

$$\begin{aligned} [\nabla^2\Psi](\vec{\mathbf{r}}) &= \left[\nabla\left(\Psi\nabla \ln|\Psi|\right)\right](\vec{\mathbf{r}}) \\ &= \Psi(\vec{\mathbf{r}})\nabla^2 \ln|\Psi(\vec{\mathbf{r}})| + \Psi(\vec{\mathbf{r}})\left(\nabla \ln|\Psi(\vec{\mathbf{r}})|\right)^2 \\ &= \Psi(\vec{\mathbf{r}})\left(\nabla^2 \ln|\Psi(\vec{\mathbf{r}})| + (\nabla \ln|\Psi(\vec{\mathbf{r}})|)^2\right). \end{aligned} \tag{2.19}$$

The kinetic energy calculation dominates the cost of VMC as it requires the trace of the Hessian of the log wave function $\nabla^2 \ln|\Psi(\vec{\mathbf{r}})| = \operatorname{Tr} \mathrm{H}_{\ln|\Psi|}(\vec{\mathbf{r}})$.

To compute derivatives such as these, one can typically choose between forward and backward differentiation. While backward differentiation, frequently called backpropagation

or backprop, is common if the codomain of a function is of a lower dimension than the domain, the opposite holds for forward differentiation. Thus, to compute this second-order derivative, one may choose from one of the following combinations: backward-on-backward, forward-on-backward, backward-on-forward, or forward-on-forward. Backward-on-backward has been done in Pfau et al. (2019); Hermann et al. (2020) as backpropagation is the default in contemporary deep learning. Later, Spencer et al. (2020) found forward-on-backward improving runtime and memory cost. The standard of forward-on-forward was established by Li et al. (2024a) by propagating the Jacobian and Laplacian of the function forward, avoiding materializing the full Hessian. Specifically, one can compute the Jacobian and Laplacian of a composite function $f(\boldsymbol{x}) = (f_N \circ \ldots \circ f_0)(\boldsymbol{x}_0), f : \mathbb{R}^{D_0} \to \mathbb{R}^{D_N}$ via

$$\boldsymbol{x}_{i+1} = f_i(\boldsymbol{x}_i), \tag{2.20}$$

$$\nabla \boldsymbol{x}_{i+1} = \mathrm{J}_{f_i}(\boldsymbol{x}_i) \nabla \boldsymbol{x}_i, \tag{2.21}$$

$$\nabla^2 \boldsymbol{x}_{i+1} = \mathrm{J}_{f_i}(\boldsymbol{x}_i) \nabla^2 \boldsymbol{x}_i + \mathrm{Tr}\left(\nabla \boldsymbol{x}_i \mathrm{H}_{f_i}(\boldsymbol{x}_i) \nabla \boldsymbol{x}_i^T\right) \tag{2.22}$$

where $\boldsymbol{x}_i, \nabla^2 \boldsymbol{x}_i \in \mathbb{R}^{D_i}, \nabla \boldsymbol{x}_i \in \mathbb{R}^{D_i \times D_0}$, $\mathrm{J}_f(\boldsymbol{x}) \in \mathbb{R}^{D_{i+1} \times D_i}$ and $\mathrm{H}_f(\boldsymbol{x}) \in \mathbb{R}^{D_{i+1} \times D_i \times D_i}$ are the Jacobian and Hessian of f at $\boldsymbol{x}$, respectively, and Tr operation goes over the last two dimensions. This formulation not only eliminates the need for extensive caching (Rumelhart et al., 1986; Dao et al., 2022) but also allows for the propagation of sparse Jacobians, eliminating unnecessary computations. We compare all three methods in Figure 2.2. During the course of the PhD, the author of this thesis developed a library to compute forward Laplacians for arbitrary functions with automatic sparsity propagation in JAX (Bradbury et al., 2018), finding broad adoption (Gao et al., 2023; Pfau et al., 2024; Zhang et al., 2025; Gerard et al., 2024).

To improve numerics, local energies are typically clipped to reduce the impact of outliers (Foulkes et al., 2001). A common technique is to clip the local energies by a multiple of the mean absolute deviation of the mean (Pfau et al., 2019) or median (von Glehn et al., 2023), i.e.,

$$\hat{E_L}(\vec{\mathbf{r}}) = \begin{cases} E_L(\vec{\mathbf{r}}) & \text{if } E_L(\vec{\mathbf{r}}) \in [\mu - k\sigma, \mu + k\sigma], \\ \mu - k\sigma & \text{if } E_L(\vec{\mathbf{r}}) < \mu - k\sigma, \\ \mu + k\sigma & \text{if } E_L(\vec{\mathbf{r}}) > \mu + k\sigma, \end{cases} \tag{2.23}$$

where μ is either the mean or median of the local energies and $\sigma = \mathbb{E}_{\vec{\mathbf{r}} \sim \rho_\Psi}\left[|E_L(\vec{\mathbf{r}}) - \mu|\right]$ is the mean absolute deviation or median absolute deviation, respectively.

2.3.2 Sampling Electronic Wave Functions

To estimate the energy, the gradient, and the preconditioner samples from $\rho_\Psi(\vec{\mathbf{r}}) = \frac{\Psi(\vec{\mathbf{r}})^2}{\langle \Psi | \Psi \rangle}$ are required. As, in most parameterizations of the wave function, the normalization $\frac{1}{\langle \Psi | \Psi \rangle}$ is unknown, one typically relies on sampling methods for unnormalized distributions. In

Algorithm 1 Metropolis Hastings algorithm

Require: $\vec{\mathbf{r}} \in \mathbb{R}^{N_\mathrm{e}\times 3}, \ln|\Psi| : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}, \sigma^2_\mathrm{step} \in \mathbb{R}_+, N_\mathrm{step} \in \mathbb{N}_+$
  $\ln q = 2\ln|\Psi(\vec{\mathbf{r}})|$
  for $i = 0; i < N_\mathrm{steps}; i \leftarrow i+1$ **do**
    $\epsilon \sim \mathcal{N}(0, \sigma^2_\mathrm{step} I)$
    $\vec{\mathbf{r}}' \leftarrow \vec{\mathbf{r}} + \epsilon$
    $\ln q' \leftarrow 2\ln|\Psi(\vec{\mathbf{r}}')|$
    $\ln\alpha \leftarrow \ln q - \ln q'$
    $u \sim \mathcal{U}(0,1)$
    if $\ln u < \ln\alpha$ **then**
      $\vec{\mathbf{r}} \leftarrow \vec{\mathbf{r}}'$
      $\ln q \leftarrow \ln q'$
    end if
  end for
  return $\vec{\mathbf{r}}$

particular, the Metropolis-Hastings algorithm (Metropolis et al., 1953) is well suited for this task as it relies solely on the likelihood ratio between two samples $\frac{\rho_\Psi(x')}{\rho_\Psi(x)}$ canceling out the normalization constant. Metropolis-Hastings defines a Monte Carlo Markov Chain (MCMC) whose stationary distribution coincides with the target distribution ρ_Ψ. In each MCMC step, one proposes a new sample $\tilde{\vec{\mathbf{r}}}^t$ according to some easy-to-sample proposal distribution

$$\tilde{\vec{\mathbf{r}}}^t \sim q(\tilde{\vec{\mathbf{r}}}|\vec{\mathbf{r}}) \tag{2.24}$$

and accepts it with probability

$$\alpha = \min\left\{1, \frac{\Psi(\tilde{\vec{\mathbf{r}}}^t)^2 q(\vec{\mathbf{r}}^t|\tilde{\vec{\mathbf{r}}}^t)}{\Psi(\vec{\mathbf{r}}^t)^2 q(\tilde{\vec{\mathbf{r}}}^t|\vec{\mathbf{r}}^t)}\right\}. \tag{2.25}$$

A common choice for q is a Gaussian distribution centered at $\vec{\mathbf{r}}^t$ with variance σ^2_step, i.e., $q(\tilde{\vec{\mathbf{r}}}^t|\vec{\mathbf{r}}^t) = \mathcal{N}(\tilde{\vec{\mathbf{r}}}^t|\vec{\mathbf{r}}^t, \sigma^2_\mathrm{step} I)$. In this case, the proposal distribution is symmetric and $\frac{q(\vec{\mathbf{r}}^t|\tilde{\vec{\mathbf{r}}}^t)}{q(\tilde{\vec{\mathbf{r}}}^t|\vec{\mathbf{r}}^t)} = 1$. While in the infinite-step limit the MCMC converges to the target distribution for any $\sigma^2_\mathrm{step} > 0$, in practice, one typically optimizes σ^2_step to a target acceptance rate of 50 % to accelerate convergence and reduce autocorrelation (Schätzle et al., 2023). We summarize the Metropolis-Hastings algorithm in Algorithm 1 where we use the logarithm of the wave function $\ln|\Psi|$ to improve numerics.

One may further accelerate convergence by utilizing the gradient of the wave function to propose more efficient moves in Langevin Monte Carlo (Besag, 2004). However, the additional computational cost of computing the gradient may outweigh the benefits in neural-network VMC (Schätzle et al., 2023).

While obtaining completely independent samples is typically infeasible, one reduces the autocorrelation between samples by thinning the chain, i.e., only keeping every

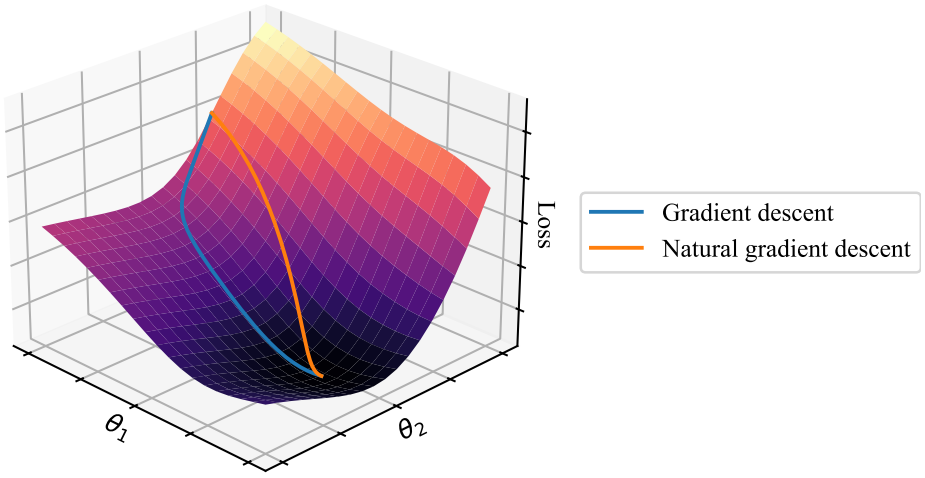


Figure 2.3: Comparison between first and second-order optimization methods. Here, we illustrate a simple quadratic loss landscape with two parameters, θ_1, θ_2. Gradient descent converges slowly along the steepest descent direction (blue), while natural gradient descent follows a straight path to the minimum (orange).

n-th sample and running multiple chains in parallel. This parallelization is particularly well-suited for the parallel architecture of modern GPUs.

2.3.3 Preconditioning

While the gradient-based optimization in VMC allows for flexibility in the design of the wave function class, gradient descent requires thousands of steps (Pfau et al., 2019) compared to the tens of steps required in Hartree-Fock theory (Pulay, 1982). To accelerate the convergence, one typically uses the quasi-Newton method Stochastic Reconfiguration (SR) (Sorella, 1998). The benefits of such a second-order method are illustrated in Figure 2.3, where the second-order method follows a direct path to the minimum compared to the first-order methods that require many less optimal steps.

In SR, the gradients of the energy $\nabla_{\theta^t} E[\Psi]$ are adjusted for the curvature of the loss landscape. Notably, SR is equivalent to natural gradient descent (Amari, 1998; Nomura et al., 2017; Pfau et al., 2019), and we use these terms interchangeably. While SR is derived as an imaginary time evolution of the Schrödinger equation, natural gradient descent stems from information geometry. To adjust the gradient to the curvature of the loss landscape, one preconditions the gradient with the inverse of the Fisher information matrix (FIM) $\boldsymbol{F} = \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\left(\nabla_\theta \ln \rho_\Psi(\vec{\mathbf{r}})\right)\left(\nabla_\theta \ln \rho_\Psi(\vec{\mathbf{r}})\right)^T\right]$:

$$\delta^t = \boldsymbol{F}^{-1} \nabla_{\theta^t} E[\Psi]. \tag{2.26}$$

In SR literature, the tensor $\mathcal{O} = \nabla_\theta \ln |\Psi(\vec{\mathbf{r}})|$ is called the quantum geometric tensor from which the FIM is computed via $\boldsymbol{F} = \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\left(\mathcal{O} - \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}[\mathcal{O}]\right)\left(\mathcal{O} - \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}[\mathcal{O}]\right)^T\right] = \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}\left[\bar{\mathcal{O}}\bar{\mathcal{O}}^T\right]$ where $\bar{\mathcal{O}} = \mathcal{O} - \mathbb{E}_{\vec{\mathbf{r}}\sim\rho_\Psi}[\mathcal{O}]$.

In practice, the expectation is computed with a finite number of samples, i.e.,

$$\boldsymbol{F} \approx \frac{1}{N_{\text{walker}}} \sum_{i=1}^{N_{\text{walker}}} \bar{\mathcal{O}}_i \bar{\mathcal{O}}_i^T = \frac{1}{N_{\text{walker}}} \bar{\boldsymbol{\mathcal{O}}} \bar{\boldsymbol{\mathcal{O}}}^T \tag{2.27}$$

If the number of parameters N_{param} is small, one may materialize $\boldsymbol{F}$ in memory and perform the inversion directly. But, for larger numbers of parameters, e.g., for neural-network wave functions, $\boldsymbol{F} \in \mathbb{R}^{N_{\text{param}} \times N_{\text{param}}}$ typically exceeds available memory. If so, one may resort to iterative matrix-free solvers such as the conjugate-gradient (CG) method (Neuscamman et al., 2012). In any case, one typically needs to add a regularization term to the FIM to ensure invertibility:

$$\tilde{\boldsymbol{F}} = \boldsymbol{F} + \lambda I. \tag{2.28}$$

By expanding $\nabla_\theta E[\Psi] = \frac{1}{N_{\text{walker}}} \bar{\boldsymbol{\mathcal{O}}} E_L[\vec{\mathbf{r}}]$ and $\boldsymbol{F} = \frac{1}{N_{\text{walker}}} \bar{\boldsymbol{\mathcal{O}}} \bar{\boldsymbol{\mathcal{O}}}^T$, one can rewrite the preconditioned gradient as

$$\delta = \left(\bar{\boldsymbol{\mathcal{O}}} \bar{\boldsymbol{\mathcal{O}}}^T + \lambda I\right)^{-1} \bar{\boldsymbol{\mathcal{O}}} E_L[\vec{\mathbf{r}}]. \tag{2.29}$$

In minSR (Chen and Heyl, 2023; Rende et al., 2024), one reformulates equation 2.29 with the Woodburry matrix identity to avoid iterative solvers:

$$\delta = \bar{\boldsymbol{\mathcal{O}}} \left(\bar{\boldsymbol{\mathcal{O}}}^T \bar{\boldsymbol{\mathcal{O}}} + \lambda I\right)^{-1} E_L[\vec{\mathbf{r}}] \tag{2.30}$$

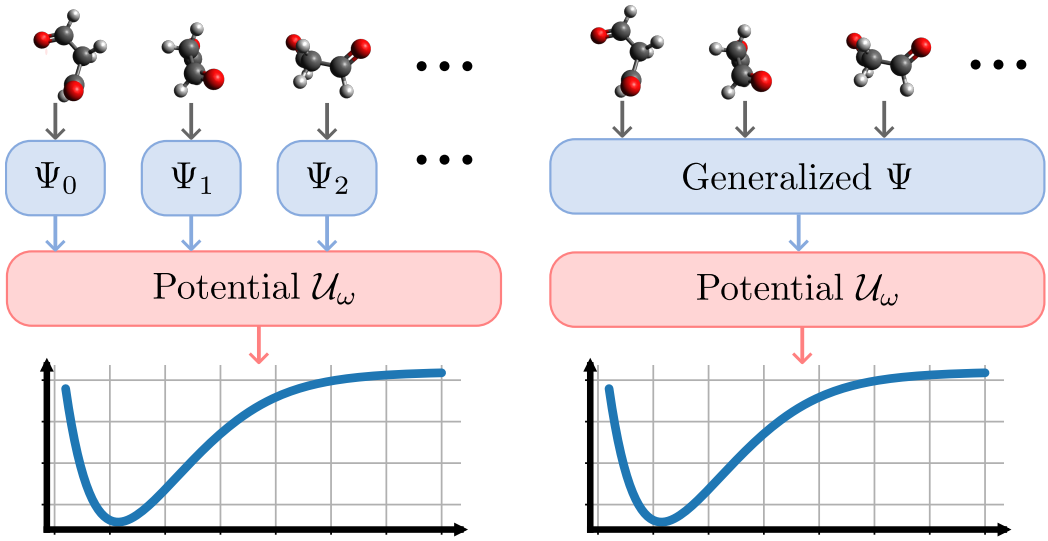
requiring only the inversion of a $N_{\text{walker}} \times N_{\text{walker}}$ matrix instead of a $N_{\text{param}} \times N_{\text{param}}$. Goldshlager et al. (2024) further improved the minSR algorithm with their Spring algorithm that introduced a Kaczmarz-inspired update rule that may also be viewed as a projected momentum update, resulting in

$$\delta^t = \bar{\boldsymbol{\mathcal{O}}} \left(\bar{\boldsymbol{\mathcal{O}}}^T \bar{\boldsymbol{\mathcal{O}}} + \lambda I\right)^{-1} \left(E_L[\vec{\mathbf{r}}] - \bar{\boldsymbol{\mathcal{O}}} \eta \delta^{t-1}\right) + \eta \delta^{t-1}. \tag{2.31}$$

In *Publications 1, 2 & 4*, we use the conjugate-gradient method to solve Equation 2.29 while in *Publication 3* we use the Spring algorithm (Equation 2.31).

2.4 Potential Energy Surface

While we so far discussed how to solve the time-independent Schrödinger equation, in practice one is typically interested in the change in energy introduced by chemical reactions or geometric formations, i.e., $\mathcal{U}(\mathbf{M}_1) - \mathcal{U}(\mathbf{M}_2)$ for molecules $\mathbf{M}_1, \mathbf{M}_2 \in \mathcal{M}$ and



(a) Traditional approach. **(b)** Generalized neural wave function.

Figure 2.4: Comparison between the traditional approach (2.4a) of obtaining the PES and our proposed generalized neural wave function approach (2.4b).

potential energy surface (PES) $\mathcal{U} : \mathcal{M} \to \mathbb{R}$. The PES maps from the set of position and charges of the nuclei $\mathcal{M} = \left\{(\vec{\mathbf{R}}, \mathbf{Z}) | N_\mathrm{n} \in \mathbb{N}_+, \vec{\mathbf{R}} \in \mathbb{R}^{N_\mathrm{n} \times 3}, \mathbf{Z} \in \mathbb{N}_+^{N_\mathrm{n}}\right\}$ to the ground-state energy $E_0 \in \mathbb{R}$ and is of central interest to compute relative energies. Unfortunately, the true PES is unavailable, and, in practice, one estimates E_0 with approximate solutions of the Schrödinger equation like the ones outlined so far.

One may accelerate the repeated solving of the Schrödinger equation by instead fitting an empirical function $\hat{\mathcal{U}}_\omega : \mathcal{M} \to \mathbb{R}$ with parameters ω to a set of reference calculations $\{\mathbf{M}_i, E_i\}_i$ (Halgren, 1996; Jiang and Guo, 2013) by minimizing some loss

$$\omega^* = \arg\min_\omega \sum_i \mathcal{L}\left(\hat{\mathcal{U}}_\omega(\mathbf{M}_i), E_i\right). \tag{2.32}$$

Such learned potentials have the advantage of being extremely fast to evaluate but are limited to the domain covered by their training set (Stocker et al., 2022) and struggle in flexibility (Pozdnyakov and Ceriotti, 2022). Traditionally, enlarging the dataset causes a linear increase in cost as each new structure needs a new computation. We visualize the traditional approach in Figure 2.4a.

In this thesis, we tackle the problem in a fundamentally novel approach to quantum chemistry, which we show in Figure 2.4b. Instead of solving the Schrödinger equation for a set of molecules $\{\mathbf{M}_i\}$ independently, we propose to learn a generalized wave function $\Psi_\theta : \mathbb{R}^{N_\mathrm{e} \times 3} \times \mathcal{M} \to \mathbb{R}$ that can solve the Schrödinger equation for all, i.e., $E[\Psi_\theta(\cdot|\mathbf{M}_i)] \approx \mathcal{U}(\mathbf{M}_i)$. By learning a joint set of parameters θ^* for all molecules instead of solving each molecule independently, we amortize the cost over all molecules, leading to sublinear scaling with the number of molecules.

3 Machine Learning in Quantum Chemistry

In recent years, machine learning has seen tremendous success in solving previously intractable problems. To name a few, it accurately simulated the folding of proteins (Abramson et al., 2024), commoditized computer vision (Redmon et al., 2016), and enabled the wide-spread use of natural language processing and generation (Brown et al., 2020). The field of quantum chemistry did not remain untouched by these developments. In various works, machine learning has been used in various ways across the quantum chemistry pipeline. Machine learning force fields offer unprecedented accuracy in molecular dynamics simulations (Batzner et al., 2022; Gasteiger et al., 2021), learned exchange-correlation functionals improve the accuracy of density functional theory (Dick and Fernandez-Serra, 2021; Nagai et al., 2022), another branch of approximate solutions to the Schrödinger equation (Kohn and Sham, 1965), and neural-network wave functions pushed the envelope of accuracy in quantum Monte Carlo simulations (Hermann et al., 2023). While each of these developments deserves its own discussion, we focus on machine-learning force fields and neural network wave functions as they are most closely related to the work presented in this thesis. However, before we explore the applications, we provide a brief overview of the underlying machine-learning concepts.

3.1 Equivariant Deep Learning

The Multilayer Perceptron (MLP) (Rosenbaltt, 1957) is the most fundamental building block of contemporary machine learning. An MLP consists of a series of feed-forward layers that sequentially transform the input $\boldsymbol{h}^{(0)} \in \mathbb{R}^{d^{(0)}}$ to the output $\boldsymbol{h}^{(L)} \in \mathbb{R}^{d^{(L)}}$:

$$\boldsymbol{h}^{(l)} = \sigma^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}\right), \tag{3.1}$$

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ and $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d^{(l)}}$ are the weights and biases of layer l, respectively, and $\sigma^{(l)}$ is an activation function. For the intermediate layers, $l < L$, the activation function is typically a non-linear function like the rectified linear unit (ReLU) (Nair and Hinton, 2010), $\sigma(x) = \max\{0, x\}$, or sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$ (Rumelhart et al., 1986). In contrast, the output activation is often a softmax or sigmoid function for classification tasks (Bridle, 1990) and the identity function for regression tasks. Because MLPs are universal function approximators (Cybenko, 1989; Hornik et al., 1989), they often serve as building blocks in specialized architectures.

Suppose the target function exhibits a well-defined structure, e.g., translational invariance in image classification or permutation invariance in graph regression. In that case, one can leverage this structure to design more data and training-efficient architectures (Batzner et al., 2022; Brehmer et al., 2024). For instance, convolutional neural networks (CNNs) (LeCun et al., 1989) implement the translational invariance of image classifiers by sharing weights across the input space, DeepSets (Zaheer et al., 2017) and Transformers (Vaswani et al., 2017) enforce permutation equivariance on sets, and graph neural networks (GNNs) (Gilmer et al., 2017) implement permutation equivariance for node-level tasks on graphs.

Permutation equivariant architectures like GNNs and Transformers are particularly interesting for this thesis as they are commonly used to model indistinguishable particles like nuclei and electrons in quantum chemistry. To be specific, a function f of indistinguishable particles frequently obeys

$$f(\tau(\boldsymbol{x})) = \operatorname{sign}(\tau)^p f(\boldsymbol{x}), \tag{3.2}$$

for a permutation $\tau \in S_N$ of N particles and a parity $p \in \{0, 1\}$. For instance, the electronic wave function Ψ is antisymmetric $p = 1$ under the exchange of electrons, i.e., $\Psi(\tau(\vec{\mathbf{r}})) = \operatorname{sign}(\tau)\Psi(\vec{\mathbf{r}})$, while the PES $\mathcal{U}$ is symmetric, i.e., invariant, $p = 0$ under the exchange of two nuclei, i.e., $\mathcal{U}(\tau(\vec{\mathbf{R}}), \tau(\mathbf{Z})) = \mathcal{U}(\vec{\mathbf{R}}, \mathbf{Z})$.

GNNs realize permutation equivariance by node-vise permutation-invariant aggregations of their neighbors' messages. In the unified framework of Gilmer et al. (2017), a single message-passing step is defined by

$$\boldsymbol{h}_i^{(t+1)} = u_t \left(\boldsymbol{h}_i^{(t)}, \bigoplus_{j \in \mathcal{N}(i)} m_t \left(\boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)}, \boldsymbol{e}_{ij}^{(t)} \right) \right), \tag{3.3}$$

$$\boldsymbol{e}_{ij}^{(t+1)} = e_t \left(\boldsymbol{e}_{ij}^{(t)} \right) \tag{3.4}$$

where $\boldsymbol{h}_i^{(t)}$ is the embedding of the ith node at step t, $\mathcal{N}(i)$ is the set of neighbors of node i, $\boldsymbol{e}_{ij}$ is the edge embedding between nodes i and j, m_t is the message function, u_t is the update function, and $\bigoplus$ is a permutation-invariant aggregation function. Frequently, one chooses the sum or mean function as aggregation and MLPs for the message m_t, update u_t, and edge e_t functions (Gilmer et al., 2017; Kipf and Welling, 2017; Thomas et al., 2018). Still, thanks to its generality, one may also view transformers as a special case of GNNs where the edge embeddings are ignored:

$$m_t(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}) = \begin{bmatrix} A(\boldsymbol{x}, \boldsymbol{y}) \\ A(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{y} \boldsymbol{W}_V \end{bmatrix}, \tag{3.5}$$

$$A(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(\frac{\boldsymbol{x} \boldsymbol{W}_Q \boldsymbol{W}_K^T \boldsymbol{y}^T}{\sqrt{d}} \right), \tag{3.6}$$

$$\bigoplus_j \boldsymbol{x}_{ij} = \frac{1}{\sum_j \boldsymbol{x}_{ij0}} \sum_j \boldsymbol{x}_{ij1}. \tag{3.7}$$

Here, $A(\boldsymbol{x}, \boldsymbol{y})$ are the unnormalized attention scores, $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d \times d}$ are learnable weights, and u_t is an MLP with a residual connection (Vaswani et al., 2017; He et al., 2015).

Within the context of quantum chemistry, the node embeddings $\boldsymbol{h}_i$ correspond to particles, e.g., electrons or nuclei, and the edge embeddings $\boldsymbol{e}_{ij}$ to interparticle distances (Schütt et al., 2018; Hermann et al., 2023). Next to the symmetric group S_N, the Euclidean group $E(3)$ is of interest in chemistry as the PES $\mathcal{U}$ is invariant under translations, rotations, and reflections of the nuclei. However, the same does not hold for the electronic wave function Ψ; thus, $E(3)$ is primarily of interest for modeling the PES and ensuring that the generalized wave function's energies are $E(3)$-invariant without enforcing $E(3)$-invariance upon the wave function itself.

3.2 Machine-Learning Force Fields

Machine-learning force fields (MLFF) approximate the PES by directly learning the mapping from molecular structures to energies, avoiding the explicit solution of the electronic Schrödinger equation. They were among the first and most impactful applications of deep learning in quantum chemistry (Zhang et al., 2023a). By enabling a feed-forward map to the energy, MLFFs are orders of magnitude faster than any quantum chemistry calculation and enable accurate molecular dynamics simulations (Schütt et al., 2018). To obtain nuclear forces $\boldsymbol{f} : \mathbb{R}^{N_\mathrm{n} \times 3} \to \mathbb{R}^{N_\mathrm{n} \times 3}$ for such simulations, one typically differentiates the energy with respect to the nuclear positions $\vec{\mathbf{R}} : \boldsymbol{f}(\vec{\mathbf{R}}) = -\nabla_{\vec{\mathbf{R}}} \hat{\mathcal{U}}_\omega(\vec{\mathbf{R}}, \mathbf{Z})$.

MLFFs are generally implemented in the GNN framework outlined in Equation 3.3 with the node embeddings $\boldsymbol{h}_i^{(t)}$ corresponding to atomic embeddings with $\boldsymbol{h}_i^{(0)}$ being an embedding of the atomic charge and the edge embeddings $\boldsymbol{e}_{ij}$ to interatomic geometric features, e.g., distances or angles. The neighbhors of each atom $\mathcal{N}(i)$ are determined by a cutoff radius $c \in \mathbb{R}_+$, i.e., $\mathcal{N}(i) = \{j \in \{1, \dots, N_\mathrm{n}\} : \|\vec{R}_i - \vec{R}_j\| < c\}$. To avoid discontinuities in the energy surface if a neighbor moves in or out of the cutoff radius, one typically scales the messages by a smooth cutoff function $f : \mathbb{R}_+ \to [0, 1]$ with $f(x) = 0$ if $\geq c$ (Schütt et al., 2018). Thus, each message passing step aggregates spatial information from the local environments of each atom by utilizing distances (Schütt et al., 2018), angles (Gasteiger et al., 2019), or higher-order interactions (Thomas et al., 2018; Gasteiger et al., 2021; Batatia et al., 2022).

For this thesis, MLFFs are of tree-fold interest:

1. To accomplish generalized neural wave functions, we have to adapt the parameters θ to the molecule at hand. We use an MLFF-like model to adapt per-nucleus parameters that change with the molecular structure in *Publication 1*. As MLFFs encode permutation and $E(3)$, the resulting neural wave function is invariant to translations, rotations, and permutations of the nuclei.

2. In *Publication 4*, we leverage the fast evaluation of MLFFs to accelerate the inference in neural-network VMC simulations by replacing the costly numerical Monte Carlo integration of the wave function with a fast on-the-fly learned MLFF.

3. Developments in MLFF have shown that neural networks can accurately approximate the PES (Batatia et al., 2022, 2024) and developed universal approximators for Euclidean point clouds (Gasteiger et al., 2021) enabling the transfer of advancements in MLFF to neural-network wave functions. This enabled us to design size-consistent neural wave functions in *Publication 2* that scale correctly in the number of particles.

3.3 Neural Wave Functions

Neural network solutions to the electronic Schrödinger equation are at the heart of this thesis. Thus, we provide an overview of existing neural wave functions in VMC and their components in the following section.

Neural networks within the VMC framework have been used to solve different problems in quantum physics (Carleo and Troyer, 2017; Luo and Clark, 2019). Here, we focus on neural wave functions that approximate the electronic wave function Ψ. These can be categorized into two approaches: first quantization and second quantization. In first quantization, one optimizes the parameters θ of a real-space antisymmetric wave function directly, i.e., $\Psi_\theta : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$. In contrast, in second quantization, one selects a finite basis set of antisymmetric functions ψ_k and optimizes the expansion coefficients α_k of the wave function, i.e., $\Psi = \sum_k \alpha_k \psi_k$ (Hermann et al., 2022). The former allows for more flexibility and empirically achieves higher accuracy but is computationally more expensive as integrals must be evaluated numerically with wave function samples that require a long MCMC (Pfau et al., 2019). Meanwhile, thanks to beneficial choices in the basis set, second quantization allows for analytical integrals, and even exact sampling of new states can frequently be possible (Choo et al., 2020; Barrett et al., 2022). Nonetheless, due to the unprecedented accuracy of neural wave functions in first quantization (Pfau et al., 2019), this thesis focuses solely on real-space wave functions and especially on their acceleration via generalization.

To enforce the fermionic antisymmetry in neural wave functions in first quantization, one typically relies on a Slater-Jastrow type wave function as in Equation 2.17. But, more generally, existing neural wave functions are well-described by

$$\Psi_\theta(\vec{\mathbf{r}}) = \mathcal{J}_\theta\left(f_\theta(\vec{\mathbf{r}}), \vec{\mathbf{r}}\right) \sum_k^{N_\mathrm{k}} \mathcal{A}_k\left(f_\theta(\vec{\mathbf{r}}), \vec{\mathbf{r}}\right) \tag{3.8}$$

where $f_\theta : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}^{N_\mathrm{e}\times N_\mathrm{f}}$ is a permutation-equivariant neural network, $\mathcal{J}_\theta : \mathbb{R}^{N_\mathrm{e}\times N_\mathrm{f}} \times \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$ is a permutation-invariant Jastrow factor, $\mathcal{A}_k : \mathbb{R}^{N_\mathrm{e}\times N_\mathrm{f}} \times \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$ are antisymmetrizers, e.g., Slater determinants (Pfau et al., 2019; Hermann et al., 2020) or

Pfaffians (Kim et al., 2023; Gao and Günnemann, 2024a). In the following, we describe each component in more detail.

While this thesis is only concerned with the ground-state wave function of molecular systems, it should be noted that VMC can be used in various settings. For instance, they have seen application in the computation excited states (Entwistle et al., 2022; Li et al., 2024b; Szabó et al., 2024; Pfau et al., 2024), as starting point for more-accurate diffusion Monte Carlo (DMC) (Wilson et al., 2021; Ren et al., 2023), for periodic systems (Wilson et al., 2022; Kim et al., 2023; Lou et al., 2023; Cassella et al., 2023b; Pescia et al., 2024; Gerard et al., 2024), or even in positron physics (Cassella et al., 2023a).

3.3.1 Embedding Neural Network

The permutation-equivariant embedding network f_θ encodes the spatial and charge information of the electrons into a fixed-size representation. Thanks to electron-electron interactions, this enables encoding electron correlation effects to capture the correlation energy in Equation 2.9. f_θ has seen various implementations, e.g., DeepSets in FermiNet (Pfau et al., 2019), GNNs in PauliNet (Hermann et al., 2020) and Moon (Gao and Günnemann, 2023a), and transformer in Psiformer (von Glehn et al., 2023). These models may be described within the GNN framework outlined in Equation 3.3. For instance, in FermiNet, the initial electron $\boldsymbol{h}_i^{(0)}$ and edge embeddings $\boldsymbol{e}_{ij}^{(0)}$ are defined based on electron-nuclei and electron-electron distances, respectively:

$$\boldsymbol{h}_i^{(0)} = \begin{bmatrix} \vec{r}_i - \vec{R}_1 \\ \|\vec{r}_i - \vec{R}_1\|_2 \\ \vdots \\ \vec{r}_i - \vec{R}_{N_\mathrm{n}} \\ \|\vec{r}_i - \vec{R}_{N_\mathrm{n}}\|_2 \end{bmatrix} \in \mathbb{R}^{4N_\mathrm{n}}, \tag{3.9}$$

$$\boldsymbol{e}_{ij}^{(0)} = \begin{bmatrix} \vec{r}_i - \vec{r}_j \\ \|\vec{r}_i - \vec{r}_j\|_2 \end{bmatrix} \in \mathbb{R}^4, \tag{3.10}$$

These embeddings are then iteratively updated by the message m_t, update u_t, and edge e_t functions

$$m_t(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{e}_{ij}) = \begin{bmatrix} \mathbb{1}(j \leq N_\uparrow)\boldsymbol{h}_j \\ \mathbb{1}(j > N_\uparrow)\boldsymbol{h}_j \\ \mathbb{1}(j \leq N_\uparrow)\boldsymbol{e}_{ij} \\ \mathbb{1}(j > N_\uparrow)\boldsymbol{e}_{ij} \end{bmatrix}, \tag{3.11}$$

$$u_t(\boldsymbol{h}_i, \boldsymbol{m}_i) = \tanh\left(\boldsymbol{h}_i \boldsymbol{W}_0^{(t)} + \boldsymbol{m}_i \boldsymbol{W}_1^{(t)} + \boldsymbol{b}_0^{(t)}\right) + \boldsymbol{h}_i^{(t)}, \tag{3.12}$$

$$e_t\left(\boldsymbol{e}_{ij}\right) = \tanh\left(\boldsymbol{e}_{ij} \boldsymbol{W}_2^{(t)} + \boldsymbol{b}_1^{(t)}\right) + \boldsymbol{e}_{ij}^{(t)}, \tag{3.13}$$

$$\bigoplus_{i \in \mathcal{N}(i)} \boldsymbol{m}_{ij} = \sum_{i=1}^{N_\mathrm{e}} \boldsymbol{m}_{ij}. \tag{3.14}$$

The message function crucially distinguishes between spin-up and spin-down electrons, as permutation invariance only holds within the same spin. The variational parameters are the weights $\boldsymbol{W}_0^{(t)}, \boldsymbol{W}_1^{(t)}, \boldsymbol{W}_2^{(t)}$ and biases $\boldsymbol{b}_0^{(t)}, \boldsymbol{b}_1^{(t)}$ of the message, update and edge functions m_t, u_t, e_t for $t \in \{1, \ldots, N_{\text{layer}}\}$. The final readout happens after N_{layer}, i.e., $f_\theta(\vec{\mathbf{r}}) = \boldsymbol{H}^{(N_{\text{layer}})}$. These iterative updates propagate the spatial arrangement of electrons and nuclei through the electron embedding, enabling the encoding of electron correlation effects. Notably, this construction of the initial electron embeddings $\boldsymbol{h}_i^{(0)}$ is not invariant with respect to nuclei permutations as the nuclear positions $\vec{\mathbf{R}}$ and charges $\mathbf{Z}$ were assumed to be fixed for a single molecule.

3.3.2 Antisymmetrizer

The antisymmetrizer $\mathcal{A}_k$ ensures that the fermionic antisymmetry holds for identical-spin electrons. To this end, the most common approach is the use of Slater determinants as in Equation 2.7 with the difference that the single-electron orbitals ϕ_i are replaced by many-electron functions $\hat{\phi}_i : \mathbb{R}^{N_\text{f}} \times \mathbb{R}^3 \to \mathbb{R}$ based on the permutation-equivariant neural network embeddings $\boldsymbol{H} = f_\theta(\vec{\mathbf{r}}) \in \mathbb{R}^{N_\text{e} \times N_\text{f}}$ and electron positions $\vec{\mathbf{r}}$ (Pfau et al., 2019; Hermann et al., 2020):

$$\begin{aligned} \mathcal{A}_k^{\text{Slater}}\left(\boldsymbol{H}, \vec{\mathbf{r}}\right) &= \det\left[\hat{\phi}_{kj}\left(\boldsymbol{h}_i, \vec{r}_i\right)\right]_{i,j=1}^{N_\uparrow} \det\left[\hat{\phi}_{kj}\left(\boldsymbol{h}_i, \vec{r}_i\right)\right]_{i,j=N_\uparrow+1}^{N_\text{e}} \\ &= \det \hat{\Phi}_k^\uparrow\left(\boldsymbol{H}^\uparrow, \vec{\mathbf{r}}^\uparrow\right) \det \hat{\Phi}_k^\downarrow\left(\boldsymbol{H}^\downarrow, \vec{\mathbf{r}}^\downarrow\right). \end{aligned} \tag{3.15}$$

This has been further refined in Spencer et al. (2020)'s implementation by observing that $\mathcal{A}_k^{\text{Slater}}$ is equivalent to the determinant of a single-block diagonal matrix:

$$\mathcal{A}_k^{\text{Slater}}\left(\boldsymbol{H}, \vec{\mathbf{r}}\right) = \det \begin{bmatrix} \hat{\Phi}_k^\uparrow\left(\boldsymbol{H}^\uparrow, \vec{\mathbf{r}}^\uparrow\right) & 0 \\ 0 & \hat{\Phi}_k^\downarrow\left(\boldsymbol{H}^\downarrow, \vec{\mathbf{r}}^\downarrow\right) \end{bmatrix}. \tag{3.16}$$

This allows one to add additional orbitals to the off-diagonal blocks without enforcing antisymmetry between non-spin-aligned electrons

$$\mathcal{A}_k^{\text{full-Slater}}\left(\boldsymbol{H}, \vec{\mathbf{r}}\right) = \det \begin{bmatrix} \hat{\phi}_{k1}\left(\boldsymbol{h}_1, \vec{r}_1\right) & \cdots & \hat{\phi}_{kN_\text{e}}\left(\boldsymbol{h}_1, \vec{r}_1\right) \\ \vdots & \ddots & \vdots \\ \hat{\phi}_{k1}\left(\boldsymbol{h}_{N_\uparrow}, \vec{r}_{N_\uparrow}\right) & \cdots & \hat{\phi}_{kN_\text{e}}\left(\boldsymbol{h}_{N_\uparrow}, \vec{r}_{N_\uparrow}\right) \\ \hat{\phi}_{kN_\text{e}+1}\left(\boldsymbol{h}_{N_\uparrow+1}, \vec{r}_{N_\uparrow+1}\right) & \cdots & \hat{\phi}_{k2N_\text{e}}\left(\boldsymbol{h}_{N_\uparrow+1}, \vec{r}_{N_\uparrow+1}\right) \\ \vdots & \ddots & \vdots \\ \hat{\phi}_{kN_\text{e}+1}\left(\boldsymbol{h}_{N_\text{e}}, \vec{r}_{N_\text{e}}\right) & \cdots & \hat{\phi}_{k2N_\text{e}}\left(\boldsymbol{h}_{N_\text{e}}, \vec{r}_{N_\text{e}}\right) \end{bmatrix}. \tag{3.17}$$

The accuracy of this approach has been confirmed by several subsequent works (Gerard et al., 2022; Lin et al., 2021; Ren et al., 2023) including our own (Gao and Günnemann, 2023b).

For specific physical systems, specialized solutions have been found to yield better performance than Slater determinants. For instance, to model superfluids, Lou et al. (2023) use AGP wave functions (Casula and Sorella, 2003), i.e.,

$$\mathcal{A}_k^{\text{AGP}}(\boldsymbol{H}, \vec{\mathbf{r}}) = \det\left(\hat{\Phi}_k^{\uparrow}\left(\boldsymbol{H}^{\uparrow}, \vec{\mathbf{r}}^{\uparrow}\right)\hat{\Phi}_k^{\downarrow}\left(\boldsymbol{H}^{\downarrow}, \vec{\mathbf{r}}^{\downarrow}\right)^T\right) \tag{3.18}$$

where $\hat{\Phi}_k^{\uparrow}\left(\boldsymbol{H}^{\uparrow}, \vec{\mathbf{r}}^{\uparrow}\right) \in \mathbb{R}^{N_{\uparrow}\times N_{\text{o}}}$, $\hat{\Phi}_k^{\downarrow}\left(\boldsymbol{H}^{\downarrow}, \vec{\mathbf{r}}^{\downarrow}\right) \in \mathbb{R}^{N_{\downarrow}\times N_{\text{o}}}$ are no longer required to be square. As a generalization of AGPs, Kim et al. (2023) propose to combine Pfaffian wave functions (Bajdich et al., 2006, 2008) with neural networks to model the ultra-cold Fermi gas. A Pfaffian can be thought of a "square root" of a determinant for skew-symmetric matrices, i.e., $\text{Pf}\,(A)^2 = \det(A)$ (Wimmer, 2012). In Kim et al. (2023), the authors parametrize the wave function's antisymmetrizer as

$$\mathcal{A}_k^{\text{Pfaffian}} = \text{Pf}\left(\left[\text{MLP}_k\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right) - \text{MLP}_k\left(\boldsymbol{h}_j, \boldsymbol{h}_i\right)\right]_{i,j=1}^{N_{\text{e}}}\right). \tag{3.19}$$

While $\mathcal{A}_k^{\text{Slater}}, \mathcal{A}_k^{\text{full-Slater}}, \mathcal{A}_k^{\text{AGP}}$ and $\mathcal{A}_k^{\text{Pfaffian}}$ dominate the computational complexity with $O({N_{\text{e}}}^3)$ in for the forward pass and $O({N_{\text{e}}}^4)$ operations for the Laplacian computation, in practice, the neural network f_θ is the computational bottleneck (Li et al., 2024a).

Still, computationally more efficient approaches have been proposed via explicitly pairwise antisymmetric functions (Han et al., 2019), Vandermonde determinants (Acevedo et al., 2020), or sorting (Richter-Powell et al., 2023). However, none of these approaches match the accuracy of Slater determinants, even on small molecules, and are thus not further considered in this thesis.

To ensure square integrability, the many-electron orbitals $\hat{\phi}_i$ are implemented as a product of a linear projection of the neural network embedding $\boldsymbol{h} \in \mathbb{R}^{N_{\text{f}}}$ and an exponentially decaying envelope function $\Lambda : \mathbb{R}^3 \to \mathbb{R}_+$ (Pfau et al., 2019; Spencer et al., 2020), i.e.,

$$\hat{\phi}_{kjk}\left(\boldsymbol{h}_i, \vec{r}_i\right) = \Lambda_{jk}(\vec{r}_i) \cdot \boldsymbol{h}_i^T \boldsymbol{w}_{jk}. \tag{3.20}$$

This functional form is reminiscent of traditional basis sets in quantum chemistry where one picks a linear combination of Gaussians as envelope functions to ensure analytical integrals (Hermann et al., 2020). However, it is known that the actual wave function has cusps as the electrons approach nuclei (Kato, 1957). Thus, a sum of exponentials yields better results (Gerard et al., 2022). A common definition stems from Spencer et al. (2020):

$$\Lambda_{jk}(\vec{r}) = \sum_{m=1}^{N_{\text{n}}} \pi_{mjk} \exp\left(-\sigma_{mjk}\|\vec{r} - \vec{R}_m\|_2\right). \tag{3.21}$$

3.3.3 Jastrow Factor

The Jastrow factor $\mathcal{J}_\theta$ has traditionally served two purposes: (1) it enforces the electron-electron and electron-nuclei cusp condition and (2) it captures the electron correlation

energy (Foulkes et al., 2001). Kato's cusp conditions state that the wave function must have cusps as electrons approach each other or a nucleus (Kato, 1957). Explicitly modeling this cusp condition within the wave function increases the accuracy and stability of VMC calculation (Foulkes et al., 2001). However, in neural-network VMC, enforcing the electron-nuclei cusps is less important as the exponential envelopes in the antisymmetrizer already provide a good approximation (Hermann et al., 2020; Gerard et al., 2022). Still, von Glehn et al. (2023) found including electron-electron cusps beneficial and proposed the following Jastrow factor:

$$\mathcal{J}_\theta(\boldsymbol{H}, \vec{\mathbf{r}}) = \exp\left(-\frac{1}{4}\sum_{i<j\leq N_\uparrow, N_\uparrow<i<j} \frac{\alpha_{\text{par}}^2}{\alpha_{\text{par}} + \|\vec{r}_i - \vec{r}_j\|} - \frac{1}{2}\sum_{i\leq N_\uparrow, j>N_\uparrow} \frac{\alpha_{\text{anti}}^2}{\alpha_{\text{anti}} + \|\vec{r}_i - \vec{r}_j\|}\right) \tag{3.22}$$

where $\alpha_{\text{par}}, \alpha_{\text{anti}} \in \mathbb{R}$ are variational parameters. To capture correlation energy, the Jastrow factor has seen less importance as neural-network embeddings in the antisymmetrizer capture most of the correlation energy (Pfau et al., 2019).

3.3.4 Pretraining Neural Wave Functions

As we eluded in Section 2.2, the mean-field approximation is a common starting point for quantum chemistry calculations as it captures most of the energy, leaving only the correlation energy to be learned. However, we ideally want to avoid coupling our neural wave function to the mean-field approximation as it introduces a bias in the wave function yielding worse energies (Gerard et al., 2022). Instead, a common strategy is to pretrain the neural wave function to replicate the mean-field solution before running the VMC optimization (Pfau et al., 2019). For Slater wave functions, the pretraining is done by minimizing the mean squared error between the neural wave function's orbitals $\hat{\phi}_i$ and the Hartree-Fock orbitals ϕ_i:

$$\theta_{\text{pretrain}} = \arg\min_\theta \mathbb{E}_{\vec{\mathbf{r}}\sim\rho}\left[\sum_{k=1}^{N_\text{k}}\sum_{i,j=1}^{N_\text{e}}\left(\hat{\phi}_{kj}\left(f_\theta(\vec{\mathbf{r}})_j, \vec{r}_j\right) - \phi_i(\vec{r}_j)\right)^2\right]. \tag{3.23}$$

As distributions for the electronic configurations ρ one either uses the Hartree-Fock distribution $\rho_{\psi_{\text{HF}}}$ (Hermann et al., 2020), the neural wave function distribution ρ_Ψ (Gerard et al., 2022), or a mixture of both (von Glehn et al., 2023). As the neural wave function changes during pretraining, its distribution ρ_Ψ also changes. To account for this, one continuously updates the samples $\vec{\mathbf{r}}$ during pretraining by running MCMC steps, as described in Section 2.3.2, between each pretraining step.

This approach has shown remarkable success as an initial guess for single-structure wave functions (Pfau et al., 2019). However, when learning generalized wave functions, the pretraining objective in Equation 3.23 may not be well-defined as the Hartree-Fock orbitals are not unique and may differ significantly between similar molecules (Szabo and Ostlund, 2012). Depending on the level of generalization, we discuss how to resolve this issue in the following section.

4 Generalized Neural Wave Functions

In this thesis, we tackle the problem of accurately evaluating the PES $\mathcal{U}$ for a set of molecules $\{\mathbf{M}_i\}$. As outlined in Section 2.4, the traditional approach is to independently repeat the calculation for each molecule or to interpolate between the results of a few selected molecules by fitting an empirical force field $\hat{\mathcal{U}}_\omega$. We approach this problem in a fundamentally novel approach for quantum chemistry, which we call *generalized neural wave functions*. Instead of solving the Schrödinger equation independently, we propose to learn a generalized wave function $\Psi_\theta : \mathbb{R}^{N_e \times 3} \times \mathcal{M} \to \mathbb{R}$ that can solve the Schrödinger equation for all, i.e., $E[\Psi_\theta(\cdot|\mathbf{M}_i)] \approx \mathcal{U}(\mathbf{M}_i)$. Notably, as the Hamiltonian's $\hat{H}$ external potential depends on the molecular structure, we define the energy for a particular molecule $\mathbf{M}$ as

$$E[\Psi_\theta(\cdot|\mathbf{M})] = \frac{\langle \Psi_\theta(\cdot|\mathbf{M})| \, \hat{H}_\mathbf{M} \, |\Psi_\theta(\cdot|\mathbf{M})\rangle}{\langle \Psi_\theta(\cdot|\mathbf{M})|\Psi_\theta(\cdot|\mathbf{M})\rangle}. \tag{4.1}$$

By learning a joint set of parameters θ^* for all molecules instead of solving for each molecule independently, we amortize the cost over all molecules, leading to sublinear scaling with the number of molecules. To find these optimal parameters, θ^*, we minimize the total energy of all molecules

$$\theta^* = \arg\min_\theta \sum_i E[\Psi_\theta(\cdot|\mathbf{M}_i)]. \tag{4.2}$$

Before we discuss the steps we took to achieve this, we first discuss desiderata for such a generalized wave function:

1. **Accuracy:** While fast and inaccurate methods like Hartree-Fock of DFT are available, neural-network VMC promises high accuracy even in so-called *strongly correlated systems*, i.e., traditionally difficult systems (Hermann et al., 2022). Sacrificing this accuracy in favor of generality would limit its scope of application. Thus, we aim to match the accuracy of such single-structure neural-network VMC calculations with our generalized wave function.

2. **Computational Cost:** The computational cost of neural-network VMC is its main drawback. Therefore, reducing this cost is the main objective of this thesis by learning a single wave function for many molecules.

3. **Symmetries:** Symmetries play a crucial role in molecular systems as the PES $\mathcal{U}$ is invariant to translations, rotations, reflections, and permutations of the nuclei. Implementing the wave function such that the resulting energies respect these

Table 4.1: Desiderata of our generalized wave functions, PESNet (Gao and Günnemann, 2022), Globe (Gao and Günnemann, 2023a), and Neural Pfaffian (Gao and Günnemann, 2024a).

Desiderata	PESNet	Globe	Neural Pfaffian
Accuracy	✓	∼	✓
Computational Cost	✓	✓	✓
Symmetries	✓	✓	✓
Size Consistency	✗	✓	✓
Level of Generalization			
Nuclear geometry $\vec{\mathbf{R}}$	✓	∼	✓
Nuclear number and charges $N_\mathrm{n}, \mathbf{Z}$	✗	∼	✓
Electronic configurations $N_\uparrow, N_\downarrow$	✗	✗	✓

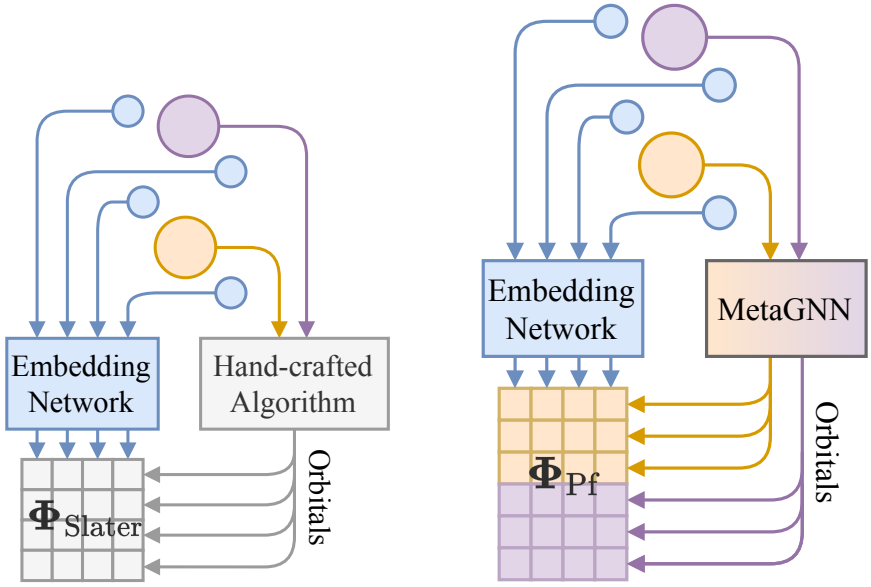
symmetries reduces training costs, results in predictable behavior, and improves generalization.

4. **Size Consistency:** Similar to symmetries, the wave function should be size-consistent, i.e., the energy should scale linearly in the number of particles. In the limit of non-interacting systems, the energy is the sum of the energies of the individual systems, i.e., $\mathcal{U}(\mathbf{M}_1 \overset{\infty}{\longleftrightarrow} \mathbf{M}_2) = \mathcal{U}(\mathbf{M}_1) + \mathcal{U}(\mathbf{M}_2)$ (Bartlett, 1981). For the wave function, this means that it should decompose into a product of wave functions for each subsystem, i.e., $\Psi(\vec{\mathbf{r}}|\mathbf{M}_1 \overset{\infty}{\longleftrightarrow} \mathbf{M}_2) = \Psi(\vec{\mathbf{r}}_{\mathbf{M}_1}|\mathbf{M}_1)\Psi(\vec{\mathbf{r}}_{\mathbf{M}_2}|\mathbf{M}_2)$. Implementing this ensures predictable scaling to larger systems.

5. **Level of Generalization:** By having a general wave function, i.e., one able to represent any molecule's ground state, we avoid specialized solutions depending on structure or molecular properties. This includes the ability to represent different molecular structures, sets of nuclei, and electronic configurations.

We approach this problem in four steps corresponding to this thesis's four "core contributions". The first three works focus on generalizing neural wave functions to molecular structures and electronic configurations. The last work focuses on the efficient evaluation of the PES without the need for Monte Carlo integration. In Table 4.1, we summarize the desiderata of our generalized wave functions and how our contributions fulfill them.

In Gao and Günnemann (2022), we introduce the *Potential Energy Surface Network (PESNet)* that learns a single wave function for a whole PES. We achieve this by introducing a meta network $\xi : \mathcal{M} \to \Theta$ that maps the molecular structure to the wave function's parameters, i.e., $\Psi_\theta(\vec{\mathbf{r}}|\mathbf{M}) := \Psi_{\xi(\mathbf{M})}(\vec{\mathbf{r}})$. Further, we introduce an equivariant canonical reference frame for each molecule and nuclei-permutation invariant initial electron embeddings to enforce the required symmetries. By constraining the generalization to different nuclear geometries, we can enforce the fermionic antisymmetry with Slater determinants as in single-structure calculations. This enables the continuous adaptation of the wave



(a) Slater orbitals. **(b)** Pfaffian orbitals.

Figure 4.1: Orbitals in generalized Slater-based wave functions (Gao and Günnemann, 2023a; Scherbela et al., 2024, 2023) in 4.1a and our Neural Pfaffians in 4.1b.

function to different molecular structures. The meta network and equivariant canonical reference frame are the foundation for the following works to perform wave function adaption. We tackle the problem of non-matching Hartree-Fock orbitals by matching the N_k different Slater determinants in the antisymmetrizer to the Hartree-Fock solutions of N_k different molecules across a PES. This work is the first to demonstrate that a single neural wave function can represent a whole PES's ground state wave function at the same accuracy and cost as a single neural-network VMC calculation.

Building on the PESNet framework, we present two core contributions in Gao and Günnemann (2023a): (1) *Graph-learned Orbital Embeddings (Globe)* to parameterize varying number of orbitals in Slater determinants and (2) *Molecular Orbital Network (Moon)* a permutation equivariant electron embedding network that is built around locality to enable size consistency. For Globe, we concentrate on stable biochemical molecules, i.e., uncharged covalent-bonded molecules. This enables us to localize orbitals as spatial objects at the nuclei and covalent bonds. The meta network adapts these orbitals to the molecular structure by using nuclei to orbital spatial message passing, enabling the generalization of different stable biochemical molecules. This way, we can represent completely different molecules' ground-state wave functions by inferring the number of orbitals from the molecular structure. Further, we demonstrate that such local orbitals paired with Moon enable size-consistent wave functions. The local nature of the Moon further avoids the dilution of electron-electron interactions in large systems, resulting in improved VMC energies in larger systems. We achieve consistent pretraining by reprojecting the Hartree-Fock orbitals to be as close as possible to the neural wave

function's orbitals without altering the resulting wave function. Compared to PESNet, Globe is limited to stable structures with clear covalent bonds and sacrifices accuracy for generalization to different molecules. But for the first time, our work demonstrates that the ground state wave function of completely different molecules can be represented as a single neural wave function.

To enable truly general wave functions, we introduce *Neural Pfaffians* in Gao and Günnemann (2024a). Here, we replace the Slater determinants with Pfaffians to enable an overparametrization, i.e., having the number of orbitals $N_\mathrm{o} \geq N_\mathrm{e}$, to enforce the antisymmetry. We visualize this in Figure 4.1. By lifting this constraint, Neural Pfaffians generalize to arbitrary molecular structures, electronic configurations, and nuclear charges. As our enlarged orbital set remains local, Neural Pfaffians remain size-consistent. To match our Pfaffian-based wave function to Hartree-Fock orbitals, we introduce an alternating two-step pretraining scheme where we on-the-fly optimize the Hartree-Fock solutions to be as close as possible to the neural wave function's orbitals without altering the resulting wave function. To accommodate the increased number of orbitals in memory, we introduce a memory-efficient envelope parametrization that also accelerates VMC convergence. Thanks to these developments, we accomplish state-of-the-art accuracy on diverse molecules, ionization potentials, and electron affinities with generalized wave functions.

Lastly, in Gao and Günnemann (2023b), we demonstrate an efficient distillation of the generalized wave function to a learned potential enabling fast and accurate predictions, which we call *Potential learning from ab-initio Networks (PlaNet)*. By utilizing the intermediate local energies of the generalized wave function, we can train an MLFF to predict the energy of a molecule at inference time without the need for Monte Carlo integration. To account for the noisy nature of the VMC energies, we perform approximate temporal averaging of the MLFF. Finally, we present several improvements to the PESNet framework to improve convergence and accuracy. We demonstrate with PlaNet that accurate and high-resolution reconstructions of multi-dimensional PESs are possible at the cost of a single neural-network VMC calculation.

Concurrently to this thesis, Scherbela et al. (2022) proposed a similar approach for amortizing the cost of neural-network VMC over multiple molecules. However, instead of learning a truly generalized wave function representing the ground state of various molecules, their works focus on partial parameter sharing and better initialization through transfer learning. In their follow-ups, Scherbela et al. (2024, 2023) adopt our generalized wave function approach from Gao and Günnemann (2022, 2023b,a) but achieve generalization by leveraging parameters from Hartree-Fock calculations. In Gerard et al. (2024), they further extend their approach to solids and demonstrate the benefits of transferable wave functions for solid systems. While overlapping, their works primarily focus on transferring solutions to unseen molecules, whereas we focus on accurate ground-state energies for a fixed set of molecules. Trained on the same dataset, our Neural Pfaffian typically yields up to an order of magnitude more accurate energies than their approach. Comparisons are provided in the respective works.

Part II: Publications

5 Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions

Authors	Nicholas Gao, Stephan Günnemann
Conference	International Conference on Learning Representations (ICLR) 2022, Selected as spotlight.
Own contribution	Conceptualization, implementation, evaluation, interpretation, and writing.
Copyright	Creative Commons attribution 4.0 International.
Source code	`https://github.com/n-gao/pesnet`
Paper	Part V: *Publication 1*

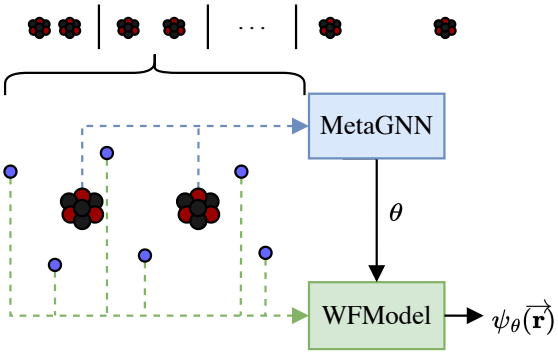


Figure 5.1: Illustration of the proposed PESNet framework.

In this work, we lay the foundation for generalized neural wave functions by introducing the first wave function, which simultaneously solves the Schrödinger equation for different structures. We accomplish this with our Potential Energy Surface Network (PESNet) within the Variational Monte Carlo (VMC) framework. PESNet accomplishes the adaption of the wave function to the molecular structure by leveraging a Graph Neural Network (GNN) that parametrizes the electronic wave function as visualized in Figure 5.1. This end-to-end differentiable adaption allows PESNet to simultaneously solve the Schrödinger equation for continuous subsets of the potential energy surface while maintaining state-of-the-art accuracy. We further enforce Euclidean invariance on the resulting potential energy surface by defining the electronic wave function within a canonical equivariant reference frame. Thanks to training only a single model, PESNet reduces the computational cost by up to 40 times compared to previous methods. This two-step architecture is the foundation for the following works in this thesis, which generalize this approach beyond different geometric configurations of the same molecule to different molecules. Further, the reparametrization approach has been adopted as the standard for generalized wave functions in related works (Scherbela et al., 2024, 2023; Gerard et al., 2024).

6 Generalizing Neural Wave Functions

Authors	Nicholas Gao, Stephan Günnemann
Conference	International Conference on Machine Learning (ICML), 2023.
Own contribution	Conceptualization, implementation, evaluation, interpretation, and writing.
Copyright	Creative Commons attribution 4.0 International.
Source code	`https://github.com/n-gao/globe`
Paper	Part V: *Publication 2*

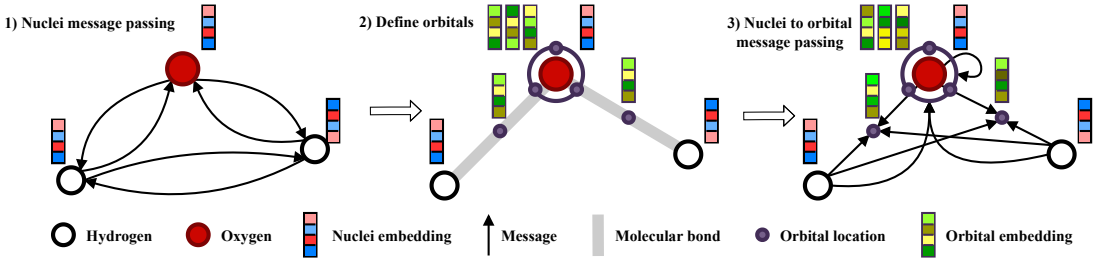


Figure 6.1: Illustration of the proposed Globe framework.

Motivated by the success of PESNet in generalizing the wave function across different structures, this work extends the concept to different molecules. The core problem when generalizing to different molecules is the different number of electrons and, thus, the different number of orbitals required to enforce the fermionic antisymmetry. We address this challenge by introducing Graph-learned Orbital Embeddings (Globe) that learns representations of local electronic structures that generalize across molecules via spatial message passing by connecting molecular orbitals to covalent bonds. We illustrate the idea in Figure 6.1, where one can see that Globe defines orbitals either as core orbital, located at the nucleus, or as valence orbital, located at the center between two nuclei. The spatial message passing enables Globe to adapt the orbitals to the chemical environment of the molecule. Additionally, we introduce a size-consistent wave function Ansatz, the Molecular Orbital Network (Moon). Our experiments demonstrate that size consistency in neural wave functions, especially in the context of generalization across molecules, accelerates convergence and improves accuracy on larger systems. Further, for the first time in quantum chemistry and machine learning, this work demonstrates that a single wave function can represent the ground state of different molecules.

7 Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations

Authors	Nicholas Gao, Stephan Günnemann
Conference	Neural Information Processing Systems (NeurIPS) 2024. Selected as oral presentation.
Own contribution	Conceptualization, implementation, evaluation, interpretation, and writing.
Copyright	Creative Commons attribution 4.0 International.
Source code	`https://github.com/n-gao/neural-pfaffian`
Paper	Part V: *Publication 3*

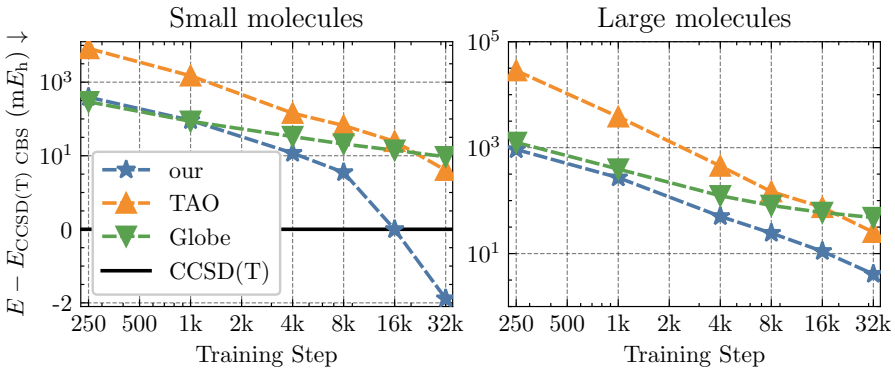


Figure 7.1: Comparison of generalized wave functions on the TinyMol dataset.

Whereas Globe demonstrated the possibility of generalizing neural wave functions across different molecules, this work improves the concept by extending its arbitrary structures and improving accuracy. We tackle the problem of enforcing the fermionic antisymmetry of electrons for arbitrary structures by defining an overparametrized neural wave function based on Pfaffians. We introduce a new class of Pfaffian-based wave functions, a strict super class of existing Slater-based wave functions. These Neural Pfaffians relax the constraint of Slater determinants needing exactly $N_\text{o} = N_\text{e}$ orbitals to $N_\text{o} \geq N_\text{e}$ orbitals, which we enforce by parametrizing a fixed number of orbitals per nucleus $N_\text{orb/nuc}$ such that $N_\text{n} N_\text{orb/nuc} = N_\text{o} \geq N_\text{e}$. This yields a fully differentiable wave function that can solve the Schrödinger equation for arbitrary structures and electron configurations. Additionally, we present practical techniques to reduce the memory footprint of the

exponential envelopes due to the additional orbitals and a pretraining scheme to initialize Neural Pfaffians based on Hartree-Fock calculations. Our empirical evaluation shows that a single Neural Pfaffians succeeds in the ground state and ionization energies with chemical accuracy across second-row elements. Further, on the diverse TinyMol dataset, Neural Pfaffians surpass 'gold-standard' CCSD(T) CBS energies and reduce errors compared to previous generalized wave functions by up to an order of magnitude within the same number of training steps. We show this core result in Figure 7.1. In summary, Neural Pfaffians are more accurate, simpler to implement, fully learnable, and flexible than previous and concurrent generalized wave functions.

8 Sampling-free Inference for Ab-Initio Potential Energy Surface Networks

Authors	Nicholas Gao, Stephan Günnemann
Conference	International Conference on Learning Representations (ICLR), 2023.
Own contribution	Conceptualization, implementation, evaluation, interpretation, and writing.
Copyright	Creative Commons attribution 4.0 International.
Source code	`https://github.com/n-gao/pesnet`
Paper	Part V: *Publication 4*

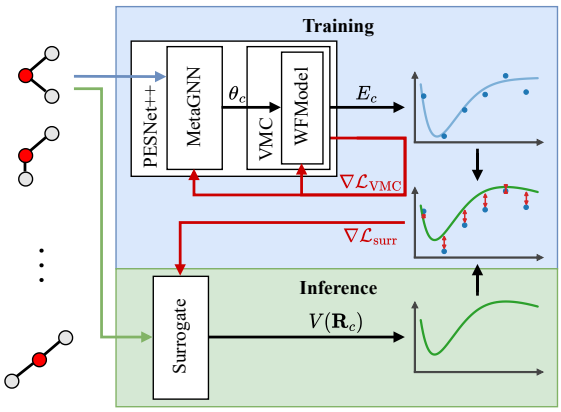


Figure 8.1: Illustration of the proposed PlaNet framework.

Generalized neural wave functions have been shown to accurately model the ground state wave function of various molecules sampled from a potential energy surface at unprecedented accuracy. However, inferring the energy per system from the wave function still requires expensive numerical Monte Carlo integration, limiting the number of geometries that can be evaluated. This work addresses this shortcoming by introducing the Potential learning from ab-initio Networks (PlaNet) framework that simultaneously trains a surrogate model and the neural wave function. At inference time, the surrogate model estimates the energy directly, avoiding the numerical integration and accelerating the process from hours to milliseconds. In this way, we can accurately model high-resolution multi-dimensional energy surfaces for larger systems that were previously unobtainable via neural wave functions. Further, we introduce a new physically-principled restricted neural wave function concept and implement it in the new PESNet++ model. In restricted neural wave functions, we enforce symmetry between swapping all spin-up and spin-down electrons. In our experiments, we find PESNet++ reduces errors by up to 74% compared to PESNet. Further, the surrogate-inferred energy surfaces accurately reflect the true energy surfaces and yield equilibrium structures in agreement with experimental results.

Part III: Concluding Remarks

9 Conclusion & Retrospective

In this thesis, we looked at the problem of obtaining highly accurate *ab-initio* solutions to the electronic Schrödinger equation for a wide range of molecules. Whereas traditional quantum chemistry tackles this problem by repeatedly solving the electronic Schrödinger equation for each molecule, we propose a fundamentally novel approach through *generalized neural wave functions*. In this approach, we use the flexibility offered by the variational Monte Carlo framework and the universal function approximation of deep learning to learn a single wave function representing the ground state of the electronic Schrödinger equation for a wide range of molecules. We accomplished this in four steps corresponding to the four publications of this thesis. In the following, we briefly summarize each and offer a retrospective view of the works.

In our first work, we concentrate on the simplified problem of obtaining a potential energy surface for a fixed set of nuclei. This constraint problem enabled us to rely on previous works to enforce the fermionic antisymmetry. We have shown that by differentiably reparametrizing the electronic wave function depending on the molecular structure, we can learn a single neural network across the whole PES, achieving the same accuracy as single structure calculations at the cost of a single calculation. In retrospect, this approach proved promising in compute time and accuracy compared to weight sharing and fine-tuning as done by Scherbela et al. (2022). Thus, their subsequent works (Scherbela et al., 2023, 2024; Gerard et al., 2024) and our follow-ups followed the same principle of reparametrizing a singular wave function across different structures.

Second, we tackled the problem of generalizing the wave function across different biochemical compounds. Here, we enforced the fermionic antisymmetry by relying on a deterministic algorithm that locates orbitals, i.e., functions describing the electron density, as points within the Euclidean space, enabling their parametrization by a neural network. For the time ever, our experiments confirm that the same neural network can be used to accurately predict the ground state of different molecules. Looking back, the strong bias toward biochemical compounds limits the generalization to arbitrary systems and accuracy and increases the overall complexity. For instance, the Hartree-Fock features relying on a solution to enforce the antisymmetry constraint in Scherbela et al. (2024) offered broader applicability and higher accuracy. Further, the pretraining algorithm had to be hand-crafted to fit the hand-crafted nature of our orbital construction. These constraints impede the generality, ease of implementation, and future work with the given architecture. Nonetheless, this work functions as the first step towards more general wave functions, and the proposed size-consistent wave function model found application in Gerard et al. (2024).

Third, we generalized the previous approach by avoiding deterministic algorithms and relying on a fully differentiable and learnable overparametrization. We accomplish this by replacing the traditional Slater determinant with a Pfaffian that relaxes the antisymmetry constraint from finding exactly N_e to $N_\mathrm{o} \geq N_\mathrm{e}$ orbitals. Thanks to this overparametrization, our Neural Pfaffian generalizes well to ionization potentials, electron affinities, molecular compounds, and other arbitrary chemical systems. Our experimental analysis confirms these foundations and shows that a single Neural Pfaffian can accurately describe the ground state of a wide range of molecules at 'gold-standard' accuracy. At the time of writing, this work remains the most general and accurate approach to the generalized wave function problem.

Lastly, while the preceding three works focused on representing the electronic wave function for as many molecules as possible, obtaining energy at inference time for any structure remains costly as it requires numerical integration over the wave function. We address this issue by presenting an online distillation method that enables inference with a single forward pass with the same accuracy as numerical integration. By correlating adjacent structures, we can distill the generalized wave function's energy surface into a force field potential solely based on intermediate results at a fraction of the numerical integration cost. In retrospect, this work focuses on practitioner's needs and offers a solution to the inference problem. However, from a methodological perspective, the distillation method is limited to the training domain of the wave function. The curse of dimensionality further limits the method's applicability to low-dimensional energy surfaces (Bellman, 1957). Still, access to such a fast and accurate force field could enable free energy calculations, molecular dynamics solutions, or geometry optimizations at high accuracy (Tuckerman, 2023).

Despite the significant progress, neural wave functions still face challenges before becoming a standard tool in quantum chemistry. Some of these are common problems in quantum chemistry as a whole, such as converging Hartree Fock for large systems without expert-driven initialization (Zhai et al., 2023; Shee et al., 2019). Others are specific to generalized neural wave functions, e.g., the drop in accuracy when scaling to diverse molecules, the required canonicalization of mean-field targets, or the discontinuities in the PES when low-lying excited states swap with the ground state. Like all accurate quantum chemistry methods, scaling to large structures at an affordable computational cost poses the biggest challenge. Lastly, as the required number of parameters and optimization steps increase with system and training set size, it remains an open question whether a foundation model wave function for the whole chemical space can be achieved efficiently.

Nevertheless, even without a foundation model, generalized neural wave functions offer a valuable framework for solving the Schrödinger equation at scale, e.g., in reaction paths or potential energy surface scans. Thus, we believe future research in this area is crucial and outline key directions in the following chapter.

10 Future Work

While this work has made significant progress in neural network-based quantum chemistry and in commoditizing highly accurate neural wave functions, many challenges remain. To inspire future researchers, we outline a non-exhaustive list of potential research directions in the following and discuss their implications.

10.1 Generative Wave Functions

The standard procedure to obtain samples from the neural wave function is to simulate a Markov chain for many steps. In between optimization steps, we perform a low number of update steps to align our samples with the updated wave function and reduce temporal correlation. In generalized wave functions, obtaining samples from the wave function may be computationally demanding if the dataset is large, as one has to repeat the Markov chain for each molecule. The issue worsens with system size increases as sampling quality worsens, and further tricks like block updates have to be deployed von Glehn et al. (2023). A neural wave function that permits exact sampling like recent generative models, e.g., diffusion models (Sohl-Dickstein et al., 2015), could eliminate this time costly step while reducing biases in the sampling process at the same time. This would enable both increases in system sizes, the dataset size, and optimization convergence. The challenge here lies in ensuring that the modeled probability distribution belongs to a valid wave function and vice versa while being fast to sample Zhang et al. (2023b). Initial works investigated this issue but had to constrain the problem to 1D to enable the use of normalizing flows (Thiede et al., 2024; Papamakarios et al., 2019).

10.2 Generalization Across Excited States

This work solely focused on the ground state of the electronic Schrödinger equation. However, many quantum chemical applications require the computation of excited states, e.g., for photochemistry, spectroscopy, or excited-state dynamics. While recent works have shown that neural networks can be used to model excited states, compute scales linearly (Szabó et al., 2024) to cubically with the number of states (Pfau et al., 2024). Achieving sublinear scaling for excited states similar to the ground state in potential energy surfaces would reduce compute times. This could be achieved in various ways, such as generalizing neural excited states, i.e., learning excited state wave functions that generalize across molecules, or by leveraging the mechanics developed in this work to

parametrize excited states through parameter sharing. In the latter case, one would use the fact that the ground state and excited states share many features and could thus be modeled with a shared neural network (Szabo and Ostlund, 2012). Naturally, both proposals could be combined to further reduce computing costs.

10.3 Scaling to Large Systems

While compute costs have been significantly reduced in this work by amortizing the cost of training over multiple molecules, the computational scaling in the number of electrons remains difficult. For instance, in von Glehn et al. (2023)'s work, computing a single energy for a benzene dimer with 84 electrons took >10 000 h on an Nvidia A100 to converge. Such high costs are out of reach for most researchers and currently limit the applicability of neural wave functions to small systems. While Li et al. (2024a) improved the compute cost of the Laplacian operator, the scaling remains quartic. Traditional VMC achieves efficient scaling by heavily relying on single-electron orbitals, enabling low-rank updates of the wave function and efficient Laplacian computations. A potential avenue for scaling lies in sparsifying the neural network, i.e., only considering a subset of the electrons for each electron's embedding. This would enable low-rank updates for neural-network VMC and reduce the overall compute cost. A disadvantage of this approach could be the reduced expressiveness of the wave function, which may limit accuracy. Though, if the accuracy of relative energies can be maintained, scaling neural wave functions to large systems may disrupt high-accuracy quantum chemistry where specialized knowledge and treatment warrants full publications per molecule (Shee et al., 2019; Tzeli et al., 2024; Zhai et al., 2023).

10.4 Generalization in Periodic Systems

This work primarily focuses on biochemistry and molecules, though other chemical applications, such as solid-state physics, require the modeling of periodic systems. Neural wave functions have shown success on small periodic systems (Li et al., 2022b) large systems remain computationally infeasible. Concurrently to this thesis, Gerard et al. (2024) proposed a transferable neural wave function approach that promises faster convergence and the efficient calculation of twist averaging (Lin et al., 2001). However, compared to the work in this thesis, their approach still requires fine-tuning the wave function for each system. Periodic systems face additional challenges as long-range interactions are significantly more pronounced and orbitals are not localized, violating some of the assumptions made in this work (Bloch, 1929). While locality allows for a straightforward generalization to larger systems by decomposing them into smaller subsystems, this does not necessarily hold for periodic systems due to the long-range interactions. Lastly, solids typically include heavy atoms and the modeling of supercells, further increasing the number of electrons and the computational cost, though pseudopotentials may reduce the severity of this (Wang et al., 2019; Li et al., 2022a).

10.5 Distillation to Faster Methods

In Gao and Günnemann (2023b), we proposed an on-the-fly distillation method to speed up the inference of neural wave functions. While working well within the training domain of the wave function, force fields are known to be unreliable even for small perturbations (Stocker et al., 2022). Instead of force fields, one may consider distilling neural wave functions into faster quantum mechanical methods such as Kohn-Sham density functional theory (KS-DFT) (Kohn and Sham, 1965). While theoretically complete, KS-DFT includes the in-practice empirically-fitted exchange-correlation functional. The fitting of such a functional typically involves highly accurate reference electron densities and energies. Neural wave functions could be the perfect candidate for generating data for fitting such a functional as it provides access to accurate energies and electron densities (Cheng et al., 2024). Combined with the recent distilling methods, such a functional could enable accurate extrapolation (Dick and Fernandez-Serra, 2021; Nagai et al., 2020; Gao et al., 2025; Kanungo et al., 2019, 2023).

Bibliography

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. ISSN 1476-4687.

A. Acevedo, M. Curry, S. H. Joshi, B. Leroux, and N. Malaya. Vandermonde Wave Function Ansatz for Improved Variational Monte Carlo. In *2020 IEEE/ACM Fourth Workshop on Deep Learning on Supercomputers (DLS)*, pages 40–47, 2020.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10 (2):251–276, 1998.

M. Bajdich, L. Mitas, G. Drobný, L. K. Wagner, and K. E. Schmidt. Pfaffian Pairing Wave Functions in Electronic-Structure Quantum Monte Carlo Simulations. *Physical Review Letters*, 96(13):130201, 2006. ISSN 0031-9007, 1079-7114.

M. Bajdich, L. Mitas, L. K. Wagner, and K. E. Schmidt. Pfaffian pairing and backflow wavefunctions for electronic structure quantum Monte Carlo methods. *Physical Review B*, 77(11):115112, 2008.

Thomas D. Barrett, Aleksei Malyshev, and A. I. Lvovsky. Autoregressive neural-network wavefunctions for ab initio quantum chemistry. *Nature Machine Intelligence*, 4(4): 351–358, 2022. ISSN 2522-5839.

Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M. Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C. Lawrence Zitnick, and Zachary W. Ulissi. Open Materials 2024 (OMat24) Inorganic Materials Dataset and Models, 2024.

R. J. Bartlett. Many-Body Perturbation Theory and Coupled Cluster Theory for Electron Correlation in Molecules. *Annual Review of Physical Chemistry*, 32(Volume 32,):359–401, 1981. ISSN 0066-426X, 1545-1593.

Ilyes Batatia, David P. Kovacs, Gregor Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Fabio Falcioni, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry, 2024.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, 2022. ISSN 2041-1723.

Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957. ISBN 978-0-691-07951-6.

Julian Besag. Markov chain Monte Carlo methods for statistical inference. *Department of Statistics University of Washington, Seattle, USA*, 2004.

Felix Bloch. Über die Quantenmechanik der Elektronen in Kristallgittern. *Zeitschrift für Physik*, 52(7):555–600, 1929. ISSN 0044-3328.

N. Bohr. Atomic Structure. *Nature*, 107(2682):104–107, 1921. ISSN 1476-4687.

M. Born and W. Heisenberg. Zur Quantentheorie der Molekeln. In Walter Blum, Helmut Rechenberg, and Hans-Peter Dürr, editors, *Original Scientific Papers Wissenschaftliche Originalarbeiten*, pages 216–246. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985. ISBN 978-3-642-64900-4 978-3-642-61659-4.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018.

Johann Brehmer, Sönke Behrends, Pim de Haan, and Taco Cohen. Does equivariance matter at scale?, 2024.

John S. Bridle. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In Françoise Fogelman Soulié and Jeanny Hérault, editors, *Neurocomputing*, pages 227–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990. ISBN 978-3-642-76155-3 978-3-642-76153-9.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, 2020.

Hartmut Bueckert, Stuart M. Rothstein, and Jan Vrbik. Optimization of quantum Monte Carlo wavefunctions using analytical derivatives. *Canadian Journal of Chemistry*, 70 (2):366–371, 1992. ISSN 0008-4042, 1480-3291.

Giuseppe Carleo and Matthias Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, 2017. ISSN 0036-8075, 1095-9203.

G. Cassella, W. M. C. Foulkes, D. Pfau, and J. S. Spencer. Neural network variational Monte Carlo for positronic chemistry, 2023a.

Gino Cassella, Halvard Sutterud, Sam Azadi, N. D. Drummond, David Pfau, James S. Spencer, and W. M. C. Foulkes. Discovering Quantum Phase Transitions with Fermionic Neural Networks. *Physical Review Letters*, 130(3):036401, 2023b.

Michele Casula and Sandro Sorella. Geminal wavefunctions with Jastrow correlation: A first application to atoms. *The Journal of Chemical Physics*, 119(13):6500–6511, 2003. ISSN 0021-9606, 1089-7690.

Michele Casula, Claudio Attaccalite, and Sandro Sorella. Correlated geminal wave function for molecules: An efficient resonating valence bond approach. *The Journal of Chemical Physics*, 121(15):7110–7126, 2004. ISSN 0021-9606.

D. Ceperley, G. V. Chester, and M. H. Kalos. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, 1977.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. The Open Catalyst 2020 (OC20) Dataset and Community Challenges. *arXiv:2010.09990 [cond-mat]*, 2021.

Ao Chen and Markus Heyl. Efficient optimization of deep neural quantum states toward machine precision, 2023.

Lixue Cheng, P. Bernát Szabó, Zeno Schätzle, Derk Kooi, Jonas Köhler, Klaas J. H. Giesbertz, Frank Noé, Jan Hermann, Paola Gori-Giorgi, and Adam Foster. Highly Accurate Real-space Electron Densities with Neural Networks, 2024.

Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature Communications*, 11(1):2368, 2020. ISSN 2041-1723.

Jiří Čížek. On the Correlation Problem in Atomic and Molecular Systems. Calculation of Wavefunction Components in Ursell-Type Expansion Using Quantum-Field Theoretical Methods. *The Journal of Chemical Physics*, 45(11):4256–4266, 1966. ISSN 0021-9606.

Aron J. Cohen, Paula Mori-Sánchez, and Weitao Yang. Insights into Current Limitations of Density Functional Theory. *Science*, 321(5890):792–794, 2008.

Alice Cuzzocrea, Anthony Scemama, Wim J. Briels, Saverio Moroni, and Claudia Filippi. Variational Principles in Quantum Monte Carlo: The Troubled Story of Variance Minimization. *Journal of Chemical Theory and Computation*, 16(7):4203–4212, 2020. ISSN 1549-9618.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989. ISSN 0932-4194, 1435-568X.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Sebastian Dick and Marivi Fernandez-Serra. Highly accurate and constrained density functional obtained with differentiable programming. *Physical Review B*, 104(16): L161109, 2021.

P. a. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(3):416–418, 1939. ISSN 1469-8064, 0305-0041.

Paul Adrien Maurice Dirac and Ralph Howard Fowler. On the theory of quantum mechanics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 112(762):661–677, 1997.

Thom H. Dunning, Jr. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *The Journal of Chemical Physics*, 90 (2):1007–1023, 1989. ISSN 0021-9606.

Mike Entwistle, Zeno Schätzle, Paolo A. Erdman, Jan Hermann, and Frank Noé. Electronic excited states in deep variational Monte Carlo. *arXiv:2203.09472 [physics, stat]*, 2022.

R. P. Feynman and Michael Cohen. Energy Spectrum of the Excitations in Liquid Helium. *Physical Review*, 102(5):1189–1204, 1956. ISSN 0031-899X.

V. Fock. Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems. *Zeitschrift für Physik*, 61(1):126–148, 1930. ISSN 0044-3328.

W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33–83, 2001.

Nicola Franco, Tom Wollschläger, Nicholas Gao, Jeanette Miriam Lorenz, and Stephan Günnemann. Quantum Robustness Verification: A Hybrid Quantum-Classical Neural Network Certification Algorithm. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 142–153, 2022.

Nicholas Gao and Stephan Günnemann. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, 2022.

Nicholas Gao and Stephan Günnemann. Generalizing Neural Wave Functions. In *International Conference on Machine Learning*, 2023a.

Nicholas Gao and Stephan Günnemann. Sampling-free Inference for Ab-Initio Potential Energy Surface Networks. In *The Eleventh International Conference on Learning Representations*, 2023b.

Nicholas Gao and Stephan Günnemann. Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Nicholas Gao and Stephan Günnemann. On Representing Electronic Wave Functions with Sign Equivariant Neural Networks. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024b.

Nicholas Gao, Jonas Köhler, and Adam Foster. Folx - Forward Laplacian for JAX, 2023.

Nicholas Gao, Eike Eberhard, and Stephan Günnemann. Learning Equivariant Non-Local Electron Density Functionals. In *The Thirteenth International Conference on Learning Representations*, 2025.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, 2019.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. GemNet: Universal Directional Graph Neural Networks for Molecules. In *Advances in Neural Information Processing Systems*, 2021.

Leon Gerard, Michael Scherbela, Philipp Marquetand, and Philipp Grohs. Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need? *Advances in Neural Information Processing Systems*, 2022.

Leon Gerard, Michael Scherbela, Halvard Sutterud, Matthew Foulkes, and Philipp Grohs. Transferable Neural Wavefunctions for Solids. 2024.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

Gil Goldshlager, Nilin Abrahamsen, and Lin Lin. A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions, 2024.

Thomas A. Halgren. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of Computational Chemistry*, 17(5-6):490–519, 1996. ISSN 1096-987X.

G. G. Hall. The Molecular Orbital Theory of Chemical Valency. VIII. A Method of Calculating Ionization Potentials. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 541–552, 1951.

Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, 2019. ISSN 0021-9991.

D. R. Hartree. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, 1928. ISSN 1469-8064, 0305-0041.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, 2015.

W. Heitler and F. London. Wechselwirkung neutraler Atome und homöopolare Bindung nach der Quantenmechanik. *Zeitschrift für Physik*, 44(6):455–472, 1927. ISSN 0044-3328.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020. ISSN 1755-4330, 1755-4349.

Jan Hermann, James Spencer, Kenny Choo, Antonio Mezzacapo, W. M. C. Foulkes, David Pfau, Giuseppe Carleo, and Frank Noé. Ab-initio quantum chemistry with neural-network wavefunctions, 2022.

Jan Hermann, James Spencer, Kenny Choo, Antonio Mezzacapo, W. M. C. Foulkes, David Pfau, Giuseppe Carleo, and Frank Noé. Ab initio quantum chemistry with neural-network wavefunctions. *Nature Reviews Chemistry*, 7(10):692–709, 2023. ISSN 2397-3358.

P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Physical Review*, 136(3B): B864–B871, 1964. ISSN 0031-899X.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Robert Jastrow. Many-body problem with strong forces. *Physical Review*, 98(5):1479, 1955.

Bin Jiang and Hua Guo. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. *The Journal of Chemical Physics*, 139(5):054112, 2013. ISSN 0021-9606.

Bikash Kanungo, Paul M. Zimmerman, and Vikram Gavini. Exact exchange-correlation potentials from ground-state electron densities. *Nature Communications*, 10(1):4497, 2019. ISSN 2041-1723.

Bikash Kanungo, Jeffrey Hatch, Paul M. Zimmerman, and Vikram Gavini. Exact and model exchange-correlation potentials for open-shell systems. *The Journal of Physical Chemistry Letters*, 14(44):10039–10045, 2023. ISSN 1948-7185, 1948-7185.

Tosio Kato. On the eigenfunctions of many-particle systems in quantum mechanics. *Communications on Pure and Applied Mathematics*, 10(2):151–177, 1957. ISSN 1097-0312.

Mohamed Amine Ketata, Nicholas Gao, Johanna Sommer, Tom Wollschläger, and Stephan Günnemann. Lift Your Molecules: Molecular Graph Generation in Latent Euclidean Space. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024.

Mohamed Amine Ketata, Nicholas Gao, Johanna Sommer, Tom Wollschläger, and Stephan Günnemann. Lift Your Molecules: Molecular Graph Generation in Latent Euclidean Space. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jane Kim, Gabriel Pescia, Bryce Fore, Jannes Nys, Giuseppe Carleo, Stefano Gandolfi, Morten Hjorth-Jensen, and Alessandro Lovato. Neural-network quantum states for ultra-cold Fermi gases, 2023.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2017.

W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140(4A):A1133–A1138, 1965. ISSN 0031-899X.

Arthur Kosmala, Johannes Gasteiger, Nicholas Gao, and Stephan Günnemann. Ewald-based Long-Range Message Passing for Molecular Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17544–17563. PMLR, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Ruichen Li, Haotian Ye, Du Jiang, Xuelan Wen, Chuwei Wang, Zhe Li, Xiang Li, Di He, Ji Chen, Weiluo Ren, and Liwei Wang. A computational framework for neural network-based variational Monte Carlo with Forward Laplacian. *Nature Machine Intelligence*, 6(2):209–219, 2024a. ISSN 2522-5839.

Xiang Li, Cunwei Fan, Weiluo Ren, and Ji Chen. Fermionic neural network with effective core potential. *Physical Review Research*, 4(1):013021, 2022a. ISSN 2643-1564.

Xiang Li, Zhe Li, and Ji Chen. Ab initio calculation of real solids via neural network ansatz. *Nature Communications*, 13(1):7895, 2022b. ISSN 2041-1723.

Zhe Li, Zixiang Lu, Ruichen Li, Xuelan Wen, Xiang Li, Liwei Wang, Ji Chen, and Weiluo Ren. Spin-symmetry-enforced solution of the many-body Schrödinger equation with a deep neural network. *Nature Computational Science*, 4(12):910–919, 2024b. ISSN 2662-8457.

C. Lin, F. H. Zong, and D. M. Ceperley. Twist-averaged boundary conditions in continuum quantum Monte Carlo algorithms. *Physical Review E*, 64(1):016702, 2001.

Jeffmin Lin, Gil Goldshlager, and Lin Lin. Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation. *arXiv:2112.03491 [physics]*, 2021.

Li-Chiang Lin, Adam H. Berger, Richard L. Martin, Jihan Kim, Joseph A. Swisher, Kuldeep Jariwala, Chris H. Rycroft, Abhoyjit S. Bhown, Michael W. Deem, and Maciej Haranczyk. In silico screening of carbon-capture materials. *Nature materials*, 11(7): 633–641, 2012.

Wan Tong Lou, Halvard Sutterud, Gino Cassella, W. M. C. Foulkes, Johannes Knolle, David Pfau, and James S. Spencer. Neural Wave Functions for Superfluids, 2023.

Di Luo and Bryan K. Clark. Backflow Transformations via Neural Networks for Quantum Many-Body Wave-Functions. *Physical Review Letters*, 122(22):226401, 2019. ISSN 0031-9007, 1079-7114.

Dmitry I. Lyakh, Monika Musiał, Victor F. Lotrich, and Rodney J. Bartlett. Multireference Nature of Chemistry: The Coupled-Cluster View. *Chemical Reviews*, 112(1): 182–243, 2012. ISSN 0009-2665, 1520-6890.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 2408–2417, 2015.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Chr. Moller and M. S. Plesset. Note on an Approximation Treatment for Many-Electron Systems. *Physical Review*, 46(7):618–622, 1934. ISSN 0031-899X.

Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. Completing density functional theory by machine learning hidden messages from molecules. *npj Computational Materials*, 6 (1):1–8, 2020. ISSN 2057-3960.

Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. Machine-learning-based exchange correlation functional with physical asymptotic constraints. *Physical Review Research*, 4(1):013106, 2022.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

Kirill Neklyudov, Jannes Nys, Luca Thiede, Juan Carrasquilla, Qiang Liu, Max Welling, and Alireza Makhzani. Wasserstein Quantum Monte Carlo: A Novel Approach for Solving the Quantum Many-Body Schr\"odinger Equation, 2023.

Eric Neuscamman, C. J. Umrigar, and Garnet Kin-Lic Chan. Optimizing large parameter sets in variational quantum Monte Carlo. *Physical Review B*, 85(4):045103, 2012. ISSN 1098-0121, 1550-235X.

Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted Boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152, 2017. ISSN 2469-9950, 2469-9969.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, 2019.

Gabriel Pescia, Jannes Nys, Jane Kim, Alessandro Lovato, and Giuseppe Carleo. Message-passing neural quantum states for the homogeneous electron gas. *Physical Review B*, 110(3):035108, 2024.

David Pfau, James S. Spencer, Alexander G. de G. Matthews, and W. M. C. Foulkes. Ab-Initio Solution of the Many-Electron Schr\"odinger Equation with Deep Neural Networks. *arXiv:1909.02487 [physics]*, 2019.

David Pfau, Simon Axelrod, Halvard Sutterud, Ingrid von Glehn, and James S. Spencer. Accurate computation of quantum excited states with neural networks. *Science*, 385 (6711):eadn0137, 2024.

Sergey N. Pozdnyakov and Michele Ceriotti. Incompleteness of graph neural networks for points clouds in three dimensions. *Machine Learning: Science and Technology*, 3 (4):045020, 2022. ISSN 2632-2153.

P. Pulay. Improved SCF convergence acceleration. *Journal of Computational Chemistry*, 3(4):556–560, 1982. ISSN 1096-987X.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection, 2016.

Weiluo Ren, Weizhong Fu, Xiaojie Wu, and Ji Chen. Towards the ground state of molecules via diffusion Monte Carlo on neural networks. *Nature Communications*, 14 (1):1860, 2023. ISSN 2041-1723.

Riccardo Rende, Sebastian Goldt, Federico Becca, and Luciano Loris Viteritti. Fine-tuning Neural Network Quantum States, 2024.

Jack Richter-Powell, Luca Thiede, Alán Asparu-Guzik, and David Duvenaud. Sorting Out Quantum Monte Carlo, 2023.

C. C. J. Roothaan. New Developments in Molecular Orbital Theory. *Reviews of Modern Physics*, 23(2):69–89, 1951.

Frank Rosenbaltt. The perceptron–a perciving and recognizing automation. *Cornell Aeronautical Laboratory*, 1957.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Z. Schätzle, P. B. Szabó, M. Mezera, J. Hermann, and F. Noé. DeepQMC: An open-source software suite for variational optimization of deep-learning molecular wave functions. *The Journal of Chemical Physics*, 159(9):094108, 2023. ISSN 0021-9606, 1089-7690.

Michael Scherbela, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nature Computational Science*, 2(5):331–341, 2022. ISSN 2662-8457.

Michael Scherbela, Leon Gerard, and Philipp Grohs. Variational Monte Carlo on a Budget — Fine-tuning pre-trained Neural Wavefunctions. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

Michael Scherbela, Leon Gerard, and Philipp Grohs. Towards a transferable fermionic neural wavefunction for molecules. *Nature Communications*, 15(1):120, 2024. ISSN 2041-1723.

Erwin Schrödinger. Quantisierung als eigenwertproblem. *Annalen der physik*, 385(13): 437–490, 1926.

K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018. ISSN 0021-9606, 1089-7690.

Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv:1706.08566 [physics, stat]*, 2017.

James Shee, Evan J. Arthur, Shiwei Zhang, David R. Reichman, and Richard A. Friesner. Singlet–Triplet Energy Gaps of Organic Biradicals and Polyacenes with Auxiliary-Field Quantum Monte Carlo. *Journal of Chemical Theory and Computation*, 15(9): 4924–4932, 2019. ISSN 1549-9618.

J. C. Slater. The Theory of Complex Spectra. *Physical Review*, 34(10):1293–1322, 1929.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Sandro Sorella. Green Function Monte Carlo with Stochastic Reconfiguration. *Physical Review Letters*, 80(20):4558–4561, 1998. ISSN 0031-9007, 1079-7114.

James S. Spencer, David Pfau, Aleksandar Botev, and W. M. C. Foulkes. Better, Faster Fermionic Neural Networks. *3rd NeurIPS Workshop on Machine Learning and Physical Science*, 2020.

Sina Stocker, Johannes Gasteiger, Florian Becker, Stephan Günnemann, and Johannes T. Margraf. How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations? Preprint, Chemistry, 2022.

Baiyu Su and Qiang Liu. Quadratic Quantum Variational Monte Carlo. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, 2012.

P. Bernát Szabó, Zeno Schätzle, Mike T. Entwistle, and Frank Noé. An improved penalty-based excited-state variational Monte Carlo approach with deep-learning ansatzes, 2024.

Georg C. Terstappen and Angelo Reggiani. In silico research in drug discovery. *Trends in Pharmacological Sciences*, 22(1):23–26, 2001. ISSN 01656147.

Luca Thiede, Chong Sun, and Alán Aspuru-Guzik. Waveflow: Boundary-conditioned normalizing flows applied to fermionic wave functions. *APL Machine Learning*, 2(4): 046106, 2024. ISSN 2770-9019.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv:1802.08219 [cs]*, 2018.

Artur P. Toshev, Teodor Kalinov, Nicholas Gao, Stephan Günnemann, and Nikolaus A. Adams. On Learning Quasi-Lagrangian Turbulence. In *ICLR 2025 Workshop on Machine Learning Multiscale Processes*, 2025.

Matthias Troyer and Uwe-Jens Wiese. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Physical Review Letters*, 94(17):170201, 2005.

Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford university press, 2023.

Demeter Tzeli, Pavlo Golub, Jiri Brabec, Mikuláš Matoušek, Katarzyna Pernal, Libor Veis, Simone Raugei, and Sotiris S. Xantheas. Importance of Electron Correlation on the Geometry and Electronic Structure of [2Fe–2S] Systems: A Benchmark Study of the [Fe2S2(SCH3)4]2–,3–,4–, [Fe2S2(SCys)4]2–, [Fe2S2(S-p-tol)4]2–, and [Fe2S2(S-o-xyl)4]2– Complexes. *Journal of Chemical Theory and Computation*, 20(23):10406–10423, 2024. ISSN 1549-9618.

C. J. Umrigar, K. G. Wilson, and J. W. Wilkins. Optimized trial wave functions for quantum Monte Carlo calculations. *Physical Review Letters*, 60(17):1719–1722, 1988. ISSN 0031-9007.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

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.

Guangming Wang, Abdulgani Annaberdiyev, Cody A. Melton, M. Chandler Bennett, Luke Shulenburger, and Lubos Mitas. A new generation of effective core potentials from correlated calculations: 4s and 4p main group elements and first row additions. *The Journal of Chemical Physics*, 151(14):144110, 2019. ISSN 0021-9606, 1089-7690.

Max Wilson, Nicholas Gao, Filip Wudarski, Eleanor Rieffel, and Norm M. Tubman. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo, 2021.

Max Wilson, Saverio Moroni, Markus Holzmann, Nicholas Gao, Filip Wudarski, Tejs Vegge, and Arghya Bhowmik. Wave function Ansatz (but Periodic) Networks and the Homogeneous Electron Gas, 2022.

Max Wilson, Saverio Moroni, Markus Holzmann, Nicholas Gao, Filip Wudarski, Tejs Vegge, and Arghya Bhowmik. Neural network ansatz for periodic wave functions and the homogeneous electron gas. *Physical Review B*, 107(23):235139, 2023.

M. Wimmer. Algorithm 923: Efficient Numerical Computation of the Pfaffian for Dense and Banded Skew-Symmetric Matrices. *ACM Transactions on Mathematical Software*, 38(4):30:1–30:17, 2012. ISSN 0098-3500.

Tom Wollschläger, Nicholas Gao, Bertrand Charpentier, Mohamed Amine Ketata, and Stephan Günnemann. Uncertainty Estimation for Molecules: Desiderata and Methods. In *Proceedings of the 40th International Conference on Machine Learning*, pages 37133–37156. PMLR, 2023.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander J. Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Huanchen Zhai, Seunghoon Lee, Zhi-Hao Cui, Lili Cao, Ulf Ryde, and Garnet Kin-Lic Chan. Multireference Protonation Energetics of a Dimeric Model of Nitrogenase Iron–Sulfur Clusters. *The Journal of Physical Chemistry A*, 127(47):9974–9984, 2023. ISSN 1089-5639.

Xuan Zhang, 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. Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems, 2023a.

Xuan Zhang, Shenglong Xu, and Shuiwang Ji. A Score-Based Model for Learning Neural Wavefunctions. 2023b.

Yaolong Zhang, Bin Jiang, and Hua Guo. SchrödingerNet: A Universal Neural Network Solver for the Schrödinger Equation. *Journal of Chemical Theory and Computation*, 21(2):670–677, 2025. ISSN 1549-9618.

Part IV: Appendix

A *Publication 1*: Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions

Publication: Nicholas Gao and Stephan Günnemann. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, 2022

Location: Virtual

License: The rights of the publication are held by the authors; from the respective web page `https://iclr.cc/FAQ/Copyright`, June 9th 2025:

> Authors do not transfer the copyright of their paper to ICLR, instead they grant ICLR a non-exclusive, perpetual, royalty-free, fully-paid, fully-assignable license to copy, distribute and publicly display all or part of the paper.

Therefore, no explicit permission is required to include the publications in this thesis.

AB-INITIO POTENTIAL ENERGY SURFACES BY PAIRING GNNS WITH NEURAL WAVE FUNCTIONS

Nicholas Gao & Stephan Günnemann
Department of Informatics & Munich Data Science Institute
Technical University of Munich, Germany
{gaoni,guennemann}@in.tum.de

ABSTRACT

Solving the Schrödinger equation is key to many quantum mechanical properties. However, an analytical solution is only tractable for single-electron systems. Recently, neural networks succeeded at modeling wave functions of many-electron systems. Together with the variational Monte-Carlo (VMC) framework, this led to solutions on par with the best known classical methods. Still, these neural methods require tremendous amounts of computational resources as one has to train a separate model for each molecular geometry. In this work, we combine a Graph Neural Network (GNN) with a neural wave function to simultaneously solve the Schrödinger equation for multiple geometries via VMC. This enables us to model continuous subsets of the potential energy surface with a single training pass. Compared to existing state-of-the-art networks, our Potential Energy Surface Network (PESNet) speeds up training for multiple geometries by up to 40 times while matching or surpassing their accuracy. This may open the path to accurate and orders of magnitude cheaper quantum mechanical calculations.

1 INTRODUCTION

In recent years, machine learning gained importance in computational quantum physics and chemistry to accelerate material discovery by approximating quantum mechanical (QM) calculations (Huang & von Lilienfeld, 2021). In particular, a lot of work has gone into building surrogate models to reproduce QM properties, e.g., energies. These models learn from datasets created using classical techniques such as density functional theory (DFT) (Ramakrishnan et al., 2014; Klicpera et al., 2019) or coupled clusters (CCSD) (Chmiela et al., 2018). While this approach has shown great success in recovering the baseline calculations, it suffers from several disadvantages. Firstly, due to the tremendous success of graph neural networks (GNNs) in this area, the regression target quality became the limiting factor for accuracy (Klicpera et al., 2019; Qiao et al., 2021; Batzner et al., 2021), i.e., the network's prediction is closer to the data label than the data label is to the actual QM property. Secondly, these surrogate models are subject to the usual difficulties of neural networks such as overconfidence outside the training domain (Pappu & Paige, 2020; Guo et al., 2017).

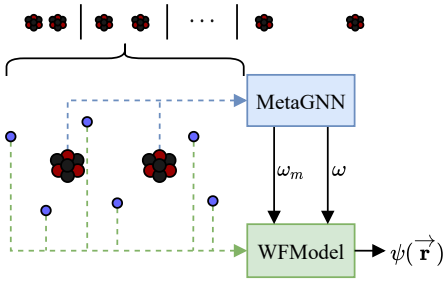


Figure 1: Schematic of PESNet. For each molecular structure (top row), the MetaGNN takes the nuclei graph and parametrizes the WFModel via $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_m$. Given these, the WFModel evaluates the electronic wave function $\psi(\vec{\boldsymbol{r}})$.

In orthogonal research, neural networks have been used as wave function Ansätze to solve the stationary Schrödinger equation (Kessler et al., 2021; Han et al., 2019). These methods use the variational Monte Carlo (VMC) (McMillan, 1965) framework to iteratively optimize a neural wave function to obtain the ground-state electronic wave function of a given system. Chemists refer to such methods as *ab-initio*, whereas the machine learning community may refer to this as a form of self-generative learning as no dataset is required. The data (electron positions) are sampled from the

wave function itself, and the loss is derived from the Schrödinger equation (Ceperley et al., 1977). This approach has shown great success as multiple authors report results outperforming the traditional 'gold-standard' CCSD on various systems (Pfau et al., 2020; Hermann et al., 2020). However, these techniques require expensive training for each geometry, resulting in high computational requirements and, thus, limiting their application to small sets of configurations.

In this work, we accelerate VMC with neural wave functions by proposing an architecture that solves the Schrödinger equation for multiple systems simultaneously. The core idea is to predict a set of parameters such that a given wave function, e.g., FermiNet (Pfau et al., 2020), solves the Schrödinger equation for a specific geometry. Previously, these parameters were obtained by optimizing a separate wave function for each geometry. We improve this procedure by generating the parameters with a GNN, as illustrated in Figure 1. This enables us to capture continuous subsets of the potential energy surface in one training pass, removing the need for costly retraining. Additionally, we take inspiration from supervised surrogate networks and enforce the invariances of the energy to physical symmetries such as translation, rotation, and reflection (Schütt et al., 2018). While these symmetries hold for observable metrics such as energies, the wave function itself may not have these symmetries. We solve this issue by defining a coordinate system that is equivariant to the symmetries of the energy. In our experiments, our Potential Energy Surface Network (PESNet) consistently matches or surpasses the results of the previous best neural wave functions while training less than $\frac{1}{40}$ of the time for high-resolution potential energy surface scans.

2 Related Work

Molecular property prediction has seen a surge in publications in recent years with the goal of predicting QM properties such as the energy of a system. Classically, features were constructed by hand and fed into a machine learning model to predict target properties (Christensen et al., 2020; Behler, 2011; Bartók et al., 2013). Lately, GNNs have proven to be more accurate and took over the field (Yang et al., 2019; Klicpera et al., 2019; Schütt et al., 2018). As GNNs approach the accuracy limit, recent work focuses on improving generalization by integrating calculations from computational chemistry. For instance, QDF (Tsubaki & Mizoguchi, 2020) and EANN (Zhang et al., 2019) approximate the electron density while OrbNet (Qiao et al., 2020) and UNiTE (Qiao et al., 2021) include features taken from QM calculations. Another promising direction is Δ-ML models, which only predict the delta between a high-accuracy QM calculation and a faster low-accuracy one (Wengert et al., 2021). Despite their success, surrogate models lack reliability. Even if uncertainty estimates are available (Lamb & Paige, 2020; Hirschfeld et al., 2020), generalization outside of the training regime is unpredictable (Guo et al., 2017).

While such supervised models are architecturally related, they pursue a fundamentally different objective than PESNet. Where surrogate models approximate QM calculations from data, this work focuses on performing the exact QM calculations from first principles.

Neural wave function Ansätze in combination with the VMC framework have recently been proposed as an alternative (Carleo & Troyer, 2017) to classical self-consistent field (SCF) methods such as Hartree-Fock, DFT, or CCSD to solve the Schrödinger equation (Szabo & Ostlund, 2012). However, early works were limited to small systems and low accuracy (Kessler et al., 2021; Han et al., 2019; Choo et al., 2020). Recently, FermiNet (Pfau et al., 2020) and PauliNet(Hermann et al., 2020) presented more scalable approaches and accuracy on par with the best traditional QM computations. To further improve accuracy, Wilson et al. (2021) coupled FermiNet with diffusion Monte-Carlo (DMC). But, all these methods need to be trained for each configuration individually. To address this issue, weight-sharing has been proposed to reduce the time per training, but this was initially limited to non-fermionic systems (Yang et al., 2020). In a concurrent work, Scherbela et al. (2021) extend this idea to electronic wave functions. However, their DeepErwin model still requires separate models for each geometry, does not account for symmetries and achieves lower accuracy, as we show in Section 4.

3 Method

To build a model that solves the Schrödinger equation for many geometries simultaneously and accounts for the symmetries of the energy, we use three key ingredients.

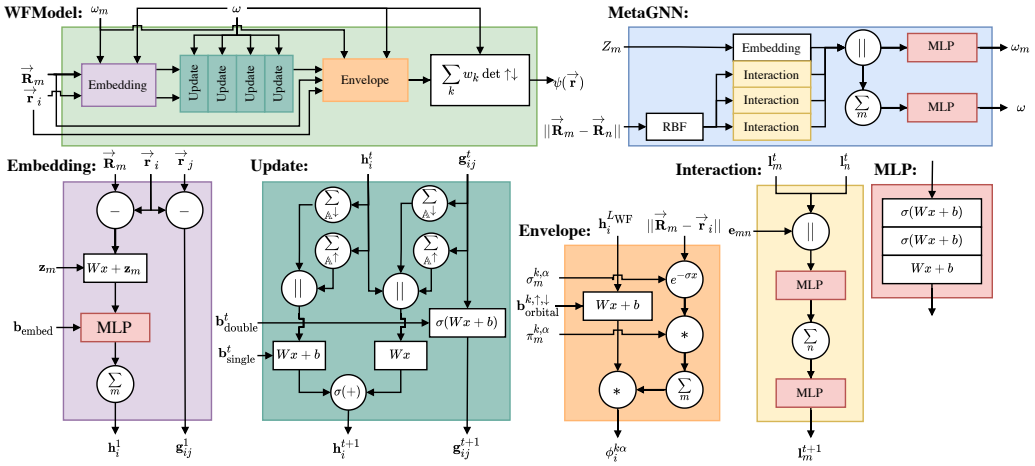


Figure 2: PESNet's architecture is split into two main components, the MetaGNN and the WFModel. Circles indicate parameter-free and rectangles parametrized functions, $\circ\|\circ$ denotes the vector concatenation, $\mathbb{A}^\uparrow$ and $\mathbb{A}^\downarrow$ denote the index sets of the spin-up and spin-down electrons, respectively. To avoid clutter, we left out residual connections.

Firstly, to solve the Schrödinger equation, we leverage the VMC framework, i.e., we iteratively update our wave function model (WFModel) until it converges to the ground-state electronic wave function. The WFModel $\psi_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}}) : \mathbb{R}^{N\times 3} \mapsto \mathbb{R}$ is a function parametrized by $\boldsymbol{\theta}$ that maps electron configurations to amplitudes. It must obey the Fermi-Dirac statistics, i.e., the sign of the output must flip under the exchange of two electrons of the same spin. As we cover in Section 3.4, the WFModel is essential for sampling electron configurations and computing energies.

Secondly, we extend this to multiple geometries by introducing a GNN that reparametrizes the WFModel. In reference to meta-learning, we call this the MetaGNN. It takes the nuclei coordinates $\overrightarrow{\boldsymbol{R}}_m$ and charges Z_m and outputs subsets $\boldsymbol{\omega}, \boldsymbol{\omega}_m \subset \boldsymbol{\theta}, m \in \{1, \dots, M\}$ of WFModel's parameters. Thanks to message passing, the MetaGNN can capture the full 3D geometry of the nuclei graph.

Lastly, as we prove in Appendix A, to predict energies invariant to rotations and reflections the wave function needs to be equivariant. We accomplish this by constructing an equivariant coordinate system $\boldsymbol{E} = [\overrightarrow{\boldsymbol{e}}_1, \overrightarrow{\boldsymbol{e}}_2, \overrightarrow{\boldsymbol{e}}_3]$ based on the principle component analysis (PCA).

Together, these components form PESNet, whose architecture is shown in Figure 2. Since sampling and energy computations only need the WFModel, a single forward pass of the MetaGNN is sufficient for each geometry during evaluation. Furthermore, its end-to-end differentiability facilitates optimization, see Section 3.4, and we may benefit from better generalization thanks to our equivariant wave function (Elesedy & Zaidi, 2021; Kondor & Trivedi, 2018).

Notation. We use bold lower-case letters $\boldsymbol{h}$ for vectors, bold upper-case $\boldsymbol{W}$ letters for matrices, $\overrightarrow{\text{arrows}}$ to indicate vectors in 3D, $\overrightarrow{\boldsymbol{r}}_i$ to denote electron coordinates, $\overrightarrow{\boldsymbol{R}}_m, Z_m$ for nuclei coordinates and charge, respectively. $[\circ, \circ]$ and $[\circ]_{i=1}^N$ denote vector concatenations.

3.1 Wave Function Model

We use the FermiNet (Pfau et al., 2020) architecture and augment it with a new feature construction that is invariant to reindexing nuclei. In the original FermiNet, the inputs to the first layer are simply concatenations of the electron-nuclei distances. This causes the features to permute if nuclei indexing changes. To circumvent this issue, we propose a new feature construction as follows:

$$\boldsymbol{h}_i^1 = \sum_{m=1}^{M} \text{MLP}\left(\boldsymbol{W}\left[(\overrightarrow{\boldsymbol{r}}_i - \overrightarrow{\boldsymbol{R}}_m)\boldsymbol{E}, \|\overrightarrow{\boldsymbol{r}}_i - \overrightarrow{\boldsymbol{R}}_m\|\right] + \boldsymbol{z}_m\right), \tag{1}$$

$$\boldsymbol{g}_{ij}^1 = \left((\overrightarrow{\boldsymbol{r}}_i - \overrightarrow{\boldsymbol{r}}_j)\boldsymbol{E}, \|\overrightarrow{\boldsymbol{r}}_i - \overrightarrow{\boldsymbol{r}}_j\|\right) \tag{2}$$

where $\boldsymbol{z}_m$ is an embedding of the m-th nuclei and $\boldsymbol{E} \in \mathbb{R}^{3\times 3}$ is our equivariant coordinate system, see Section 3.3. By summing over all nuclei instead of concatenating we obtain the desired invariance. The features are then iteratively updated using the update rule from Wilson et al. (2021)

$$\boldsymbol{h}_i^{t+1} = \sigma\left(\boldsymbol{W}_{\text{single}}^t \left[\boldsymbol{h}_i^t, \sum_{j\in\mathbb{A}^\uparrow} \boldsymbol{g}_{ij}^t, \sum_{j\in\mathbb{A}^\downarrow} \boldsymbol{g}_{ij}^t\right] + \boldsymbol{b}_{\text{single}}^t + \boldsymbol{W}_{\text{global}}^t \left[\sum_{j\in\mathbb{A}^\uparrow} \boldsymbol{h}_j^t, \sum_{j\in\mathbb{A}^\downarrow} \boldsymbol{h}_j^t\right]\right), \tag{3}$$

$$\boldsymbol{g}_{ij}^{t+1} = \sigma\left(\boldsymbol{W}_{\text{double}}^t \boldsymbol{g}_{ij}^t + \boldsymbol{b}_{\text{double}}^t\right) \tag{4}$$

where σ is an activation function, $\mathbb{A}^\uparrow$ and $\mathbb{A}^\downarrow$ are the index sets of the spin-up and spin-down electrons, respectively. We also add skip connections where possible. We chose $\sigma \coloneqq \tanh$ since it must be at least twice differentiable to compute the energy, see Section 3.4. After L_{WF} many updates, we take the electron embeddings $\boldsymbol{h}_i^{L_{\text{WF}}}$ and construct K orbitals:

$$\begin{aligned} \phi_{ij}^{k\alpha} &= (\boldsymbol{w}_i^{k\alpha} \boldsymbol{h}_j^{L_{\text{WF}}} + b_{\text{orbital},i}^{k\alpha}) \sum_m^M \pi_{im}^{k\alpha} \exp(-\sigma_{im}^{k\alpha} \| \overrightarrow{\boldsymbol{r}}_j - \overrightarrow{\boldsymbol{R}}_m \|), \\ \pi_{im}^{k\alpha} &= \text{Sigmoid}(p_{im}^{k\alpha}), \qquad \sigma_{im}^{k\alpha} = \text{Softplus}(s_{im}^{k\alpha}) \end{aligned} \tag{5}$$

where $k \in \{1, \cdots, K\}$, $\alpha \in \{\uparrow, \downarrow\}$, $i, j \in \mathbb{A}^\alpha$, and $\boldsymbol{p}_i, \boldsymbol{s}_i$ are free parameters. Here, we use the sigmoid and softplus functions to ensure that the wave function decays to 0 infinitely far away from any nuclei. To satisfy the antisymmetry to the exchange of same-spin electrons, the output is a weighted sum of determinants (Hutter, 2020)

$$\psi(\overrightarrow{\boldsymbol{r}}) = \sum_{k=1}^K w_k \det \phi^{k\uparrow} \det \phi^{k\downarrow}. \tag{6}$$

3.2 METAGNN

The MetaGNN's task is to adapt the WFModel to the geometry at hand. It does so by substituting subsets, $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_m$, of WFModel's parameters. While $\boldsymbol{\omega}_m$ contains parameters specific to nuclei m, $\boldsymbol{\omega}$ is a set of nuclei-independent parameters such as biases. To capture the geometry of the nuclei, the GNN embeds the nuclei in a vector space and updates the embeddings via learning message passing. Contrary to surrogate GNNs, we also account for the position in our equivariant coordinate system when initializing the node embeddings to avoid identical embeddings in symmetric structures. Hence, our node embeddings are initialized by

$$\boldsymbol{l}_m^1 = \left[\boldsymbol{G}_{Z_m}, f_{\text{pos}}(\overrightarrow{\boldsymbol{R}}_m' \boldsymbol{E})\right] \tag{7}$$

where $\boldsymbol{G}$ is a matrix of charge embeddings, $Z_m \in \mathbb{N}_+$ is the charge of nucleus m, $f_{\text{pos}} : \mathbb{R}^3 \mapsto \mathbb{R}^{N_{\text{SBF}} \cdot N_{\text{RBF}}}$ is our positional encoding function, and $\overrightarrow{\boldsymbol{R}}_m' \boldsymbol{E}$ is the relative position of the mth nucleus in our equivariant coordinate system $\boldsymbol{E}$ (see Section 3.3). As positional encoding function, we use the spherical Fourier-Bessel basis functions $\tilde{a}_{\text{SBF},ln}$ from Klicpera et al. (2019)

$$f_{\text{pos}}(\overrightarrow{\boldsymbol{x}}) = \sum_{i=1}^3 \left[\tilde{a}_{\text{SBF},ln}(\|\overrightarrow{\boldsymbol{x}}\|, \angle(\overrightarrow{\boldsymbol{x}}, \overrightarrow{\boldsymbol{e}}_i))\right]_{l\in\{0,\ldots,N_{\text{SBF}}-1\}, n\in\{1,\ldots,N_{\text{RBF}}\}} \tag{8}$$

with $\overrightarrow{\boldsymbol{e}}_i$ being the ith axis of our equivariant coordinate system $\boldsymbol{E}$. Unlike Klicpera et al. (2019), we are working on the fully connected graph and, thus, neither include a cutoff nor the envelope function that decays to 0 at the cutoff.

A message passing layer consists of a message function f_{msg} and an update function f_{update}. Together, one can compute an update to the embeddings as

$$\boldsymbol{l}_m^{t+1} = f_{\text{update}}^l \left(\boldsymbol{l}_m^t, \sum_n f_{\text{msg}}^t(\boldsymbol{l}_m^t, \boldsymbol{l}_n^t, \boldsymbol{e}_{mn})\right) \tag{9}$$

where $\boldsymbol{e}_{mn}$ is an embedding of the edge between nucleus m and nucleus n. We use Bessel radial basis functions to encode the distances between nuclei (Klicpera et al., 2019). Both f_{msg} and f_{update} are realized by simple feed-forward neural networks with residual connections.

After L_{GNN} many message passing steps, we compute WFModel's parameters on two levels. On the global level, $f^{\text{out}}_{\text{global}}$ outputs the biases of the network and, on the node level, $f^{\text{out}}_{\text{node}}$ outputs nuclei specific parameters:

$$\begin{aligned}\boldsymbol{\omega} &= \left[\boldsymbol{b}^1_{\text{single/double}}, \ldots, \boldsymbol{b}^{\uparrow/\downarrow}_1, \ldots, \boldsymbol{w}\right] \coloneqq f^{\text{out}}_{\text{global}}\left(\left[\sum_m \boldsymbol{l}^t_m\right]^{L_{\text{GNN}}}_{t=1}\right), \\ \boldsymbol{\omega}_m &= \left[\boldsymbol{z}_m, \boldsymbol{s}^{1,\uparrow/\downarrow}_m, \ldots, \boldsymbol{p}^{1,\uparrow/\downarrow}_m, \ldots\right] \coloneqq f^{\text{out}}_{\text{node}}\left(\left[\boldsymbol{l}^t_m\right]^{L_{\text{GNN}}}_{t=1}\right).\end{aligned} \tag{10}$$

We use distinct feed-forward neural networks with multiple heads for the specific types of parameters estimated to implement $f^{\text{out}}_{\text{node}}$ and $f^{\text{out}}_{\text{global}}$.

3.3 Equivariant Coordinate Systems

Incorporating symmetries helps to reduce the training space significantly. In GNNs this is done by only operating on inter-nuclei distances without a clear directionality in space, i.e., without x, y, z coordinates. While this works for predicting observable metrics such as energies, it does not work for wave functions. For instance, any such GNN could only describe spherically symmetric wave functions for the hydrogen atom despite all excited states (the real spherical harmonics) not having such symmetries. Unfortunately, as we show in Appendix B, recently proposed equivariant GNNs (Thomas et al., 2018; Batzner et al., 2021) also suffer from the same limitation.

To solve this issue, we introduce directionality in the form of a coordinate system that is equivariant to rotations and reflections. The axes of our coordinate system $\boldsymbol{E} = [\vec{\boldsymbol{e}}_1, \vec{\boldsymbol{e}}_2, \vec{\boldsymbol{e}}_3]$ are defined by the principal components of the nuclei coordinates, $\vec{\boldsymbol{e}}^{\text{PCA}}_1$, $\vec{\boldsymbol{e}}^{\text{PCA}}_2$, $\vec{\boldsymbol{e}}^{\text{PCA}}_3$. Using PCA is robust to reindexing nuclei and ensures that the axes rotate with the system and form an orthonormal basis. However, as PCA only returns directions up to a sign, we have to resolve the sign ambiguity. We do this by computing an equivariant vector $\vec{\boldsymbol{v}}$, i.e, a vector that rotates and reflects with the system, and defining the direction of the axes as

$$\vec{\boldsymbol{e}}_i = \begin{cases} \vec{\boldsymbol{e}}^{\text{PCA}}_i & \text{, if } \vec{\boldsymbol{v}}^T \vec{\boldsymbol{e}}^{\text{PCA}}_i \geq 0, \\ -\vec{\boldsymbol{e}}^{\text{PCA}}_i & \text{, else.} \end{cases} \tag{11}$$

As equivariant vector we use the difference between a weighted and the regular center of mass

$$\vec{\boldsymbol{v}} \coloneqq \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{n=1}^{M}\|\vec{\boldsymbol{R}}_m - \vec{\boldsymbol{R}}_n\|^2\right) Z_m \vec{\boldsymbol{R}}'_m, \tag{12}$$

$$\vec{\boldsymbol{R}}'_m = \vec{\boldsymbol{R}}_m - \frac{1}{M}\sum_{m=1}^{M}\vec{\boldsymbol{R}}_m. \tag{13}$$

With this construction, we obtain an equivariant coordinate system that defines directionality in space. However, we are aware that PCA may not be robust, e.g., if eigenvalues of the covariance matrix are identical. A detailed discussion on such edge cases can be found in Appendix C.

3.4 Optimization

We use the standard VMC optimization procedure (Ceperley et al., 1977) where we seek to minimize the expected energy of a wave function $\psi_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$:

$$\mathcal{L} = \frac{\langle\psi_{\boldsymbol{\theta}}|\boldsymbol{H}|\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}|\psi_{\boldsymbol{\theta}}\rangle} \tag{14}$$

where $\boldsymbol{H}$ is the Hamiltonian of the Schrödinger equation

$$\boldsymbol{H} = -\frac{1}{2}\sum_{i=1}^{N}\nabla^2_i + \underbrace{\sum_{i=1}^{N}\sum_{j=i}^{N}\frac{1}{\|\vec{\boldsymbol{r}}_i - \vec{\boldsymbol{r}}_j\|} - \sum_{i=1}^{N}\sum_{m=1}^{M}\frac{1}{\|\vec{\boldsymbol{r}}_i - \vec{\boldsymbol{R}}_m\|} + \sum_{m=1}^{M}\sum_{n=m}^{M}\frac{Z_m Z_n}{\|\vec{\boldsymbol{R}}_m - \vec{\boldsymbol{R}}_n\|}}_{V(\vec{\boldsymbol{r}})} \tag{15}$$

with ∇^2 being the Laplacian operator and $V(\overrightarrow{\boldsymbol{r}})$ describing the potential energy. Given samples from the probability distribution $\sim \psi^2_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})$, one can obtain an unbiased estimate of the gradient

$$\begin{aligned} E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}}) &= \psi^{-1}_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})\boldsymbol{H}\psi_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}}) \\ &= -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{3}\left[\frac{\partial^2 \log|\psi_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})|}{\partial \overrightarrow{\boldsymbol{r}}^2_{ik}} + \frac{\partial \log|\psi_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})|}{\partial \overrightarrow{\boldsymbol{r}}_{ik}}^2\right] + V(\overrightarrow{\boldsymbol{r}}), \end{aligned} \tag{16}$$

$$\nabla_{\boldsymbol{\theta}}\mathcal{L} = \mathbb{E}_{\overrightarrow{\boldsymbol{r}}\sim\psi^2_{\boldsymbol{\theta}}}\left[\left(E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}}) - \mathbb{E}_{\overrightarrow{\boldsymbol{r}}\sim\psi^2_{\boldsymbol{\theta}}}[E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})]\right)\nabla_{\boldsymbol{\theta}}\log|\psi_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})|\right] \tag{17}$$

where $E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})$ denotes the local energy of the WFModel with parameters $\boldsymbol{\theta}$ for the electron configuration $\overrightarrow{\boldsymbol{r}}$. One can see that for the energy computation, we only need the derivative of the wave function w.r.t. the electron coordinates. As these are no inputs to the MetaGNN, we do not have to differentiate through the MetaGNN to obtain the local energies. We clip the local energy as in Pfau et al. (2020) and obtain samples from $\sim \psi^2_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})$ via Metropolis-Hastings. The gradients for the MetaGNN are computed jointly with those of the WFModel by altering Equation 17:

$$\nabla_{\Theta}\mathcal{L} = \mathbb{E}_{\overrightarrow{\boldsymbol{r}}\sim\psi^2_{\boldsymbol{\theta}}}\left[\left(E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}}) - \mathbb{E}_{\overrightarrow{\boldsymbol{r}}\sim\psi^2_{\boldsymbol{\theta}}}[E_{\boldsymbol{\theta}}(\overrightarrow{\boldsymbol{r}})]\right)\nabla_{\Theta}\log|\psi_{\Theta}(\overrightarrow{\boldsymbol{r}})|\right] \tag{18}$$

where Θ is the joint set of WFModel and MetaGNN parameters. To obtain the gradient for multiple geometries, we compute the gradient as in Equation 18 multiple times and average. This joint gradient of the WFModel and the MetaGNN enables us to use a single training pass to simultaneously solve multiple Schrödinger equations.

While Equation 18 provides us with a raw estimate of the gradient, different techniques have been used to construct proper updates to the parameters (Hermann et al., 2020; Pfau et al., 2020). Here, we use natural gradient descent to enable the use of larger learning rates. So, instead of doing a regular gradient descent step in the form of $\Theta^{t+1} = \Theta^t - \eta\nabla_{\Theta}\mathcal{L}$, where η is the learning rate, we add the inverse of the Fisher information matrix as a preconditioner

$$\Theta^{t+1} = \Theta^t - \eta\boldsymbol{F}^{-1}\nabla_{\Theta}\mathcal{L}, \tag{19}$$

$$\boldsymbol{F} = \mathbb{E}_{\overrightarrow{\boldsymbol{r}}\sim\psi^2_{\boldsymbol{\theta}}}\left[\nabla_{\Theta}\log|\psi_{\Theta}(\overrightarrow{\boldsymbol{r}})|\nabla_{\Theta}\log|\psi_{\Theta}(\overrightarrow{\boldsymbol{r}})|^T\right]. \tag{20}$$

Since the Fisher $\boldsymbol{F}$ scales quadratically with the numbers of parameters, we approximate $\boldsymbol{F}^{-1}\nabla_{\Theta}\mathcal{L}$ via the conjugate gradient (CG) method (Neuscamman et al., 2012). To determine the convergence of the CG method, we follow Martens (2010) and stop based on the quadratic error. To avoid tuning the learning rate η, we clip the norm of the preconditioned gradient $\boldsymbol{F}^{-1}\nabla_{\Theta}\mathcal{L}$ (Pascanu et al., 2013) and use a fixed learning rate for all systems.

We pretrain all networks with the Lamb optimizer (You et al., 2020) on Hartree-Fock orbitals, i.e., we match each of the K orbitals to a Hartree-Fock orbital of a different configuration. During pretraining, only the WFModel and the final biases of the MetaGNN are optimized.

3.5 Limitations

While PESNet is capable of accurately modeling complex potential energy surfaces, we have not focused on architecture search yet. Furthermore, as we discuss in Section 4, PauliNet (Hermann et al., 2020) still offers a better initialization and converges in fewer iterations than our network. Lastly, PESNet is limited to geometries of the same set of nuclei with identical electron spin configurations, i.e., to access properties like the electron affinity one still needs to train two models.

4 Experiments

To investigate PESNet's accuracy and training time benefit, we compare it to FermiNet (Pfau et al., 2020; Spencer et al., 2020), PauliNet (Hermann et al., 2020), and DeepErwin (Scherbela et al., 2021) on diverse systems ranging from 3 to 28 electrons. Note, the concurrently developed DeepErwin was only recently released as a pre-print and still requires separate models and training for each configuration. While viewing the results on energies one should be aware that, except for PESNet, all methods must be trained separately for each configuration resulting in significantly higher training times, as discussed in Section 4.1.

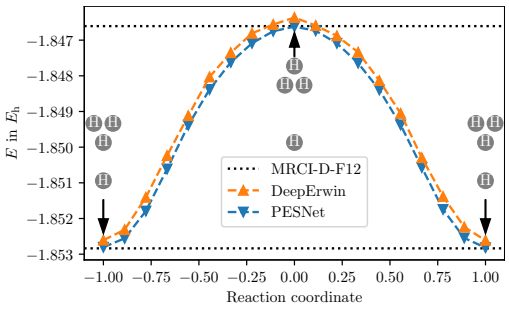


Figure 3: The energy of H_4^+ along the first reaction path (Alijah & Varandas, 2008). While PESNet and DeepErwin match the barrier height estimate of the MRCI-D-F12 calculation, PESNet estimates $\approx 0.27\,\mathrm{m}E_\mathrm{h}$ lower energies. Reference data is taken from Scherbela et al. (2021).

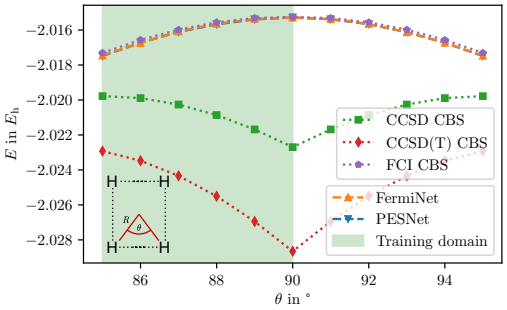


Figure 4: Potential energy surface scan of the hydrogen rectangle. Similar to FermiNet, PESNet does not produce the fake minimum at 90°. Since PESNet respects the symmetries of the energy, we only trained on half of the config space. Reference data is taken from Pfau et al. (2020).

Evaluation of *ab-initio* methods is still a challenge as true energies are rarely known, and experimental data are subject to uncertainties. In addition, many energy differences may seem small due to their large absolute values, but chemists set the threshold for chemical accuracy to $1\,\mathrm{kcal\,mol}^{-1} \approx 1.6\,\mathrm{m}E_\mathrm{h}$. Thus, seemingly small differences in energy are significant. Therefore, to put all results into perspective, we always include highly accurate classical reference calculations. When comparing VMC methods such as PESNet, FermiNet, PauliNet, and DeepErwin, interpretation is simpler: lower is always better as VMC energies are upper bounds (Szabo & Ostlund, 2012).

To analyze PESNet's ability to capture continuous subsets of the potential energy surface, we train on the continuous energy surface rather than on a discrete set of configurations for potential energy surface scans. The exact procedure and the general experimental setup are described in Appendix D. Additional ablation studies are available in Appendix E.

Transition path of H_4^+ and weight sharing. Scherbela et al. (2021) use the first transition path of H_4^+ (Alijah & Varandas, 2008) to demonstrate the acceleration gained by weight-sharing. But, they found their weight-sharing scheme to be too restrictive and additionally optimized each wave function separately. Unlike DeepErwin, our novel PESNet is flexible enough such that we do not need any extra optimization. In Figure 3, we see the DeepErwin results after their multi-step optimization and the energies of a single PESNet. We notice that while both methods estimate similar transition barriers PESNet results in $\approx 0.27\,\mathrm{m}E_\mathrm{h}$ smaller energies which match the very accurate MRCI-D-F12 results ($\approx 0.015\,\mathrm{m}E_\mathrm{h}$).

Hydrogen rectangle and symmetries. The Hydrogen rectangle is a known failure case for CCSD and CCSD(T). While the exact solution, FCI, indicates a local maximum at $\theta = 90°$, both, CCSD and CCSD(T), predict local minima. Figure 4 shows that VMC methods such as FermiNet and our PESNet do not suffer from the same issue. PESNet's energies are identical to FermiNet's ($\approx 0.014\,\mathrm{m}E_\mathrm{h}$) despite training only a single network on half of the configuration space thanks to our equivariant coordinate system.

The hydrogen chain is a very common benchmark geometry that allows us to compare our method to a range of classical methods (Motta et al., 2017) as well as to FermiNet, PauliNet, and DeepErwin. Figure 5 shows the potential energy surface of the hydrogen chain computed by a range of methods. While our PESNet generally performs identical to FermiNet, we predict on average $0.31\,\mathrm{m}E_\mathrm{h}$ lower energies. Further, we notice that our results are consistently better than PauliNet and DeepErwin despite only training a single model.

The nitrogen molecule poses a challenge as classical methods such as CCSD or CCSD(T) fail to reproduce the experimental results (Lyakh et al., 2012; Le Roy et al., 2006). While the accurate r12-MR-ACPF method more closely matches the experimental results, it scales factorially (Gdanitz, 1998). After Pfau et al. (2020) have shown that FermiNet is capable of modeling such complex triple bonds, we are interested in PESNet's performance. To better represent both methods, we

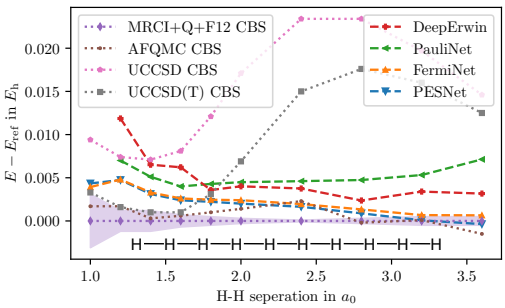


Figure 5: Potential energy surface scan of the hydrogen chain with 10 atoms. We find our PESNet to outperform PauliNet and DeepErwin strictly while matching the results of FermiNet across all configurations. Reference data is taken from Hermann et al. (2020); Pfau et al. (2020); Scherbela et al. (2021); Motta et al. (2017).

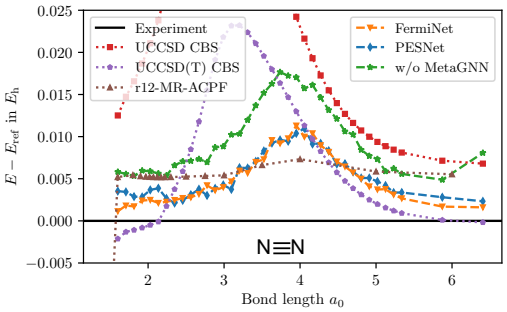


Figure 6: Potential energy surface scan of the nitrogen molecule. PESNet yields very similar but slightly higher ($\approx 0.37\,\mathrm{m}E_\mathrm{h}$) energies than FermiNet. Without the MetaGNN the accuracy drops significantly by $\approx 4.3\,\mathrm{m}E_\mathrm{h}$ on average. Reference data is taken from Le Roy et al. (2006); Gdanitz (1998); Pfau et al. (2020).

decided to compare both FermiNet as well as PESNet with 32 determinants due to a substantial performance gain for both methods. The results in Figure 6 show that PESNet agrees very well with FermiNet and is on average just $0.37\,\mathrm{m}E_\mathrm{h}$ higher, despite training only a single model for less than $\frac{1}{47}$ of FermiNet's training time, see Section 4.1. In addition, the ablation of PESNet without the MetaGNN shows a significant loss of accuracy of $4.3\,\mathrm{m}E_\mathrm{h}$ on average.

Cyclobutadiene and the MetaGNN. The automerization of cyclobutadiene is challenging due to its multi-reference nature, i.e., single reference methods such as CCSD(T) overestimate the transition barrier (Lyakh et al., 2012). In contrast, PauliNet and FermiNet had success at modelling this challenging system. Naturally, we are interested in how well PESNet can estimate the transition barrier. To be comparable to Spencer et al. (2020), we increased the number of determinants to 32 and the single-stream size to 512. Similar to PauliNet Hermann et al. (2020), we found PESNet to occasionally converge to a higher energy for the transition state depending on the initialization and pretraining. To avoid this, we pick a well-initialized model by training 5 models for 1000 iterations and then continue the rest of the optimization with the model yielding the lowest energy.

As shown in Figure 7, all neural methods converge to the same transition barrier which aligns with the highest MR-CC results at the upper end of the experimental range. But, they require different numbers of training steps and result in different total energies. PauliNet generally converges fastest, but results in the highest energies, whereas FermiNet's transition barrier converges slower but its energies are $70\,\mathrm{m}E_\mathrm{h}$ smaller. Lastly, PESNet's transition barrier converges similar PauliNet's, but its energies are $54\,\mathrm{m}E_\mathrm{h}$ lower than PauliNet's, placing it closer to FermiNet than PauliNet in terms of accuracy. Considering that PESNet has only been trained for $\approx \frac{1}{6}$ of FermiNet's time (see Section 4.1), we are confident that additional optimization would further narrow the gap to FermiNet.

In an additional ablation study, we compare to PESNet without the MetaGNN, i.e, we still train a single model for both states of cyclobutadiene but without weight adaptation. While the results in Figure 7 show that the truncated network's energies continuously decrease, it fails to reproduce the same transition barrier and its energies are $18\,\mathrm{m}E_\mathrm{h}$ worse compared to the full PESNet.

4.1 Training Time

While the previous experiments have shown that our model's accuracy is on par with FermiNet, PESNet's main appeal is its capability to fit multiple geometries simultaneously. Here, we study the training times for all systems from the previous section. We compare the official JAX (Bradbury et al., 2018) implementation of FermiNet (Spencer et al., 2020), the official PyTorch implementation of PauliNet (Hermann et al., 2020), the official TensorFlow implementation of DeepErwin (Scherbela et al., 2021), and our JAX implementation of PESNet. We use the same hyperparameters as in the experiments or the defaults from the respective works. All measurements have been conducted on a machine with 16 AMD EPYC 7543 cores and a single Nvidia A100 GPU. Here, we only mea-

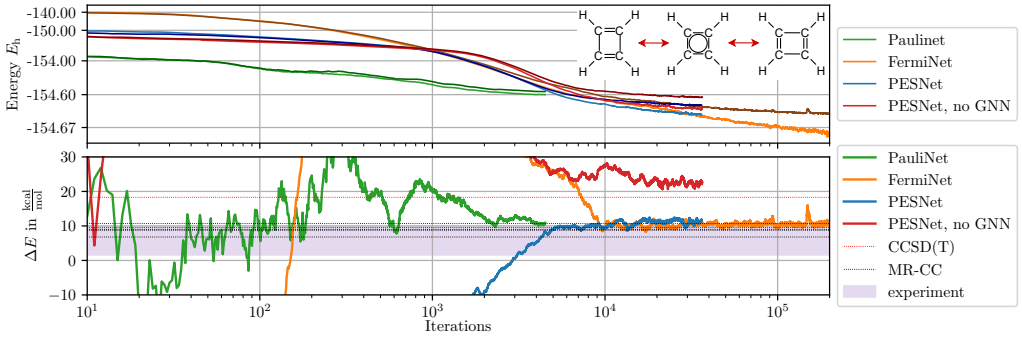


Figure 7: Comparison between the ground and transition states of cyclobutadiene. The top figure shows the total energy plotted in log scale zeroed at $-154.68\,E_\text{h}$ with light colors for the ground state and darker colors for the transition state. The bottom figure shows the estimate of the transition barrier. Both figures use a logarithmic x-axis. All neural methods estimate the same transition barriers in line with the highest MR-CC results at the upper end of the experimental data. Reference energies are taken from Hermann et al. (2020); Spencer et al. (2020); Shen & Piecuch (2012).

	H_4^+ (Fig. 3)	H4 (Fig. 4)	H10 (Fig. 5)	N2 (Fig. 6)	Cyclobutadiene (Fig. 7)
PauliNet	43h*	34h*	153h	854h*	437h
DeepErwin	34h	27h*	111h	—	—
FermiNet	127h*	118h	594h	4196h	2309h
PESNet	20h	24h	65h	89h	381h

Table 1: Total GPU (A100) hours to train all models of the respective figures. *Experiments are not included in the original works and timings are measured with the default parameters for the respective models. — Larger molecules did not work with DeepErwin.

sure the VMC training time and explicitly exclude the time to perform the SCFs calculations or any pretraining as these take up less than 1% of the total training time for all methods. Furthermore, note that the timings refer to training all models of the respective experiments.

Table 1 shows the GPU hours to train the models of the last section. It is apparent that PESNet used the fewest GPU hours across all systems. Compared to the similar accurate FermiNet, PESNet is up to 47 times faster to train. This speedup is especially noticeable if many configurations shall be evaluated, e.g., 39 nitrogen geometries. Compared to the less accurate PauliNet and DeepErwin, PESNet's speed gain shrinks, but our training times are still consistently lower while achieving significantly better results. Additionally, for H4, H10, and N2, PESNet is not fitted to the plotted discrete set of configurations but instead on a continuous subset of the energy surface. Thus, if one is interested in additional configurations, PESNet's speedup grows linearly in the number of configurations. Still, the numbers in this section do not tell the whole story, thus, we would like to refer the reader to Appendix F and G for additional discussion on training and convergence.

5 Discussion

We presented a novel architecture that can simultaneously solve the Schrödinger equation for multiple geometries. Compared to the existing state-of-the-art networks, our PESNet accelerates the training for many configurations by up to 40 times while often achieving better accuracy. The integration of physical symmetries enables us to reduce our training space. Finally, our results show that a single model can capture a continuous subset of the potential energy surface. This acceleration of neural wave functions opens access to accurate quantum mechanical calculations to a broader audience. For instance, it may enable significantly higher-resolution analyses of complex potential energy surfaces with foreseeable applications in generating new datasets with unprecedented accuracy as well as possible integration into molecular dynamics simulations.

Ethics and reproducibility. Advanced computational chemistry tools may have a positive impact in chemistry research, for instance in material discovery. However, they also pose the risk of misuse, e.g., for the development of chemical weapons. To the best of our knowledge, our work does not promote misuse any more than general computational chemistry research. To reduce the likelihood of such misuse, we publish our source code under the Hippocratic license (Ehmke, 2019)[1]. To facilitate reproducibility, the source code includes simple scripts to reproduce all experiments from Section 4. Furthermore, we provide a detailed schematic of the computational graph in Figure 2 and additional details on the experimental setup including all hyperparameters in Appendix D.

Acknowledgements. We thank David Pfau, James Spencer, Jan Hermann, and Rafael Reisenhofer for providing their results and data, Johannes Margraf, Max Wilson and Christoph Scheurer for helpful discussions, and Johannes Klicpera and Leon Hetzel for their valuable feedback.

Funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

REFERENCES

Alexander Alijah and António J. C. Varandas. H4+: What do we know about it? *The Journal of Chemical Physics*, 129(3):034303, July 2008. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2953571.

Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, May 2013. ISSN 1098-0121, 1550-235X. doi: 10.1103/PhysRevB.87.184115.

Simon Batzner, Tess E. Smidt, Lixin Sun, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, and Boris Kozinsky. SE(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. *arXiv:2101.03164 [cond-mat, physics:physics]*, January 2021.

Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, February 2011. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.3553717.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018.

Giuseppe Carleo and Matthias Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, February 2017. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aag2302.

D. Ceperley, G. V. Chester, and M. H. Kalos. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, October 1977. doi: 10.1103/PhysRevB.16.3081.

Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, 9(1):3887, September 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06169-2.

Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature Communications*, 11(1):2368, December 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15724-9.

Anders S. Christensen, Lars A. Bratholm, Felix A. Faber, and O. Anatole von Lilienfeld. FCHL revisited: Faster and more accurate quantum machine learning. *The Journal of chemical physics*, 152(4):044107, 2020.

Coraline Ada Ehmke. The Hippocratic License 2.1: An Ethical License for Open Source. https://firstdonoharm.dev, 2019.

[1] https://www.daml.in.tum.de/pesnet

Bryn Elesedy and Sheheryar Zaidi. Provably Strict Generalisation Benefit for Equivariant Models. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2959–2969. PMLR, July 2021.

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. *Chemical Physics Letters*, 283(5):253–261, February 1998. ISSN 0009-2614. doi: 10.1016/S0009-2614(97)01392-4.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330. PMLR, July 2017.

Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, December 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108929.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y.

Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W. Coley. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *Journal of Chemical Information and Modeling*, 60(8):3770–3780, August 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00502.

Bing Huang and O. Anatole von Lilienfeld. Ab Initio Machine Learning in Chemical Compound Space. *Chemical Reviews*, 121(16):10001–10036, August 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.0c01303.

Marcus Hutter. On Representing (Anti)Symmetric Functions. *arXiv:2007.15298 [quant-ph]*, July 2020.

Jan Kessler, Francesco Calcavecchia, and Thomas D. Kühne. Artificial Neural Networks as Trial Wave Functions for Quantum Monte Carlo. *Advanced Theory and Simulations*, 4(4):2000269, 2021. ISSN 2513-0390. doi: 10.1002/adts.202000269.

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. *The Journal of Physical Chemistry A*, 111(4):734–742, February 2007. ISSN 1089-5639, 1520-5215. doi: 10.1021/jp065721k.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*, December 2014.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, September 2019.

Risi Kondor and Shubhendu Trivedi. On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups. In *International Conference on Machine Learning*, pp. 2747–2755. PMLR, July 2018.

George Lamb and Brooks Paige. Bayesian Graph Neural Networks for Molecular Property Prediction. *arXiv:2012.02089 [cs, q-bio]*, November 2020.

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. *The Journal of Chemical Physics*, 125(16):164310, October 2006. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2354502.

Dmitry I. Lyakh, Monika Musiał, Victor F. Lotrich, and Rodney J. Bartlett. Multireference Nature of Chemistry: The Coupled-Cluster View. *Chemical Reviews*, 112(1):182–243, January 2012. ISSN 0009-2665, 1520-6890. doi: 10.1021/cr2001417.

James Martens. Deep learning via Hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 735–742, Madison, WI, USA, June 2010. Omnipress. ISBN 978-1-60558-907-7.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pp. 2408–2417, 2015.

W. L. McMillan. Ground State of Liquid He4. *Physical Review*, 138(2A):A442–A451, April 1965. doi: 10.1103/PhysRev.138.A442.

Mario Motta, David M. Ceperley, Garnet Kin-Lic Chan, John A. Gomez, Emanuel Gull, Sheng Guo, Carlos A. Jiménez-Hoyos, Tran Nguyen Lan, Jia Li, Fengjie Ma, Andrew J. Millis, Nikolay V. Prokof'ev, Ushnish Ray, Gustavo E. Scuseria, Sandro Sorella, Edwin M. Stoudenmire, Qiming Sun, Igor S. Tupitsyn, Steven R. White, Dominika Zgid, Shiwei Zhang, and Simons Collaboration on the Many-Electron Problem. 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. *Physical Review X*, 7(3):031059, September 2017. ISSN 2160-3308. doi: 10.1103/PhysRevX.7.031059.

Eric Neuscamman, C. J. Umrigar, and Garnet Kin-Lic Chan. Optimizing large parameter sets in variational quantum Monte Carlo. *Physical Review B*, 85(4):045103, January 2012. ISSN 1098-0121, 1550-235X. doi: 10.1103/PhysRevB.85.045103.

Aneesh Pappu and Brooks Paige. Making Graph Neural Networks Worth It for Low-Data Molecular Machine Learning. *arXiv:2011.12203 [cs]*, November 2020.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pp. 1310–1318. PMLR, 2013.

David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, September 2020. doi: 10.1103/PhysRevResearch.2.033429.

Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller III. OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital Features. *The Journal of Chemical Physics*, 153(12):124111, September 2020. ISSN 0021-9606, 1089-7690. doi: 10.1063/5.0021955.

Zhuoran Qiao, Anders S. Christensen, Frederick R. Manby, Matthew Welborn, Anima Anandkumar, and Thomas F. Miller III. UNiTE: Unitary N-body Tensor Equivariant Network with Applications to Quantum Chemistry. *arXiv:2105.14655 [physics]*, May 2021.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, December 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22.

Michael Scherbela, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *arXiv:2105.08351 [physics]*, May 2021.

K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24): 241722, June 2018. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.5019779.

Jun Shen and Piotr Piecuch. Combining active-space coupled-cluster methods with moment energy corrections via the CC(P ; Q) methodology, with benchmark calculations for biradical transition states. *The Journal of Chemical Physics*, 136(14):144104, April 2012. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.3700802.

James S. Spencer, David Pfau, Aleksandar Botev, and W. M. C. Foulkes. Better, Faster Fermionic Neural Networks. *3rd NeurIPS Workshop on Machine Learning and Physical Science*, November 2020.

Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, 2012.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv:1802.08219 [cs]*, May 2018.

Masashi Tsubaki and Teruyasu Mizoguchi. Quantum Deep Field: Data-Driven Wave Function, Electron Density Generation, and Atomization Energy Prediction and Extrapolation with Machine Learning. *Physical Review Letters*, pp. 6, 2020.

Simon Wengert, Gábor Csányi, Karsten Reuter, and Johannes T. Margraf. Data-efficient machine learning for molecular crystal structure prediction. *Chemical Science*, pp. 10.1039.D0SC05765G, 2021. ISSN 2041-6520, 2041-6539. doi: 10.1039/D0SC05765G.

Max Wilson, Nicholas Gao, Filip Wudarski, Eleanor Rieffel, and Norm M. Tubman. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo. *arXiv:2103.12570 [physics, physics:quant-ph]*, March 2021.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, August 2019. ISSN 1549-9596, 1549-960X. doi: 10.1021/acs.jcim.9b00237.

Li Yang, Wenjun Hu, and Li Li. Scalable variational Monte Carlo with graph neural ansatz. In *NeurIPS Workshop on Machine Learning and the Physical Sciences*, November 2020.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *Eighth International Conference on Learning Representations*, April 2020.

Yaolong Zhang, Ce Hu, and Bin Jiang. Embedded Atom Neural Network Potentials: Efficient and Accurate Machine Learning with a Physically Inspired Representation. *The Journal of Physical Chemistry Letters*, 10(17):4962–4967, September 2019. ISSN 1948-7185, 1948-7185. doi: 10.1021/acs.jpclett.9b02037.

A Invariant Energies Require Equivariant Wave Functions

Here, we prove that a wave function needs to be equivariant with respect to rotations and reflections if the energy is invariant. Recall, our goal is to solve the stationary Schrödinger equation

$$\boldsymbol{H}\psi = E\psi \tag{21}$$

where $\psi : \mathbb{R}^{3N} \mapsto \mathbb{R}$ is the electronic wave function. Since the Hamiltonian $\boldsymbol{H}$ encodes the molecular structure via its potential energy term, see Equation 15, a rotation or reflection $\boldsymbol{U} \in O(3)$ of the system results is a unitary transformation applied to the Hamiltonian $\boldsymbol{H} \to \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^\dagger$. Since the energy E is invariant to rotation and reflection, we obtain the transformed equation

$$\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^\dagger\psi' = E\psi'. \tag{22}$$

One can see that if ψ is a solution to Equation 21, the equivariantly transformed $\psi \to \boldsymbol{U}\psi$ solves Equation 22

$$\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^\dagger\boldsymbol{U}\psi = E\boldsymbol{U}\psi, \tag{23}$$

$$\boldsymbol{U}\boldsymbol{H}\psi = E\boldsymbol{U}\psi, \tag{24}$$

$$\boldsymbol{H}\psi = E\psi \tag{25}$$

with the same energy. □

B EQUIVARIANT NEURAL NETWORKS AS WAVE FUNCTIONS

Here, we want to briefly discuss why equivariant neural networks as proposed by Thomas et al. (2018) or Batzner et al. (2021) are no alternative to our equivariant coordinate system. The issue is the same as for regular GNNs (Klicpera et al., 2019), namely that such networks can only represent spherically symmetric functions for atomic systems which, as discussed in Section 3.3, is insufficient for wave functions. While this is obvious for regular GNNs, as they operate only on inter-particle distances rather than vectors, equivariant neural networks take advantage of higher SO(3) representations. However, if one would construct the orbitals $\phi(\overrightarrow{\boldsymbol{r}}) = [\phi_1(\overrightarrow{\boldsymbol{r}}), \dots, \phi_N(\overrightarrow{\boldsymbol{r}})]$ by concatenating E equivariant SO(3) representations $\phi(\overrightarrow{\boldsymbol{r}}) = [\phi_1(\overrightarrow{\boldsymbol{r}}), \dots, \phi_E(\overrightarrow{\boldsymbol{r}})]$ with $\sum_{e=1}^{E} \dim(\phi_e(\overrightarrow{\boldsymbol{r}}_i)) = N$, any resulting real-valued wave function $\psi(\overrightarrow{\boldsymbol{r}}) = \det \phi(\overrightarrow{\boldsymbol{r}})$ would be spherically symmetric, i.e., $\psi(\overrightarrow{\boldsymbol{r}} R) = \psi(\overrightarrow{\boldsymbol{r}}), \forall R \in SO(3)$.

The proof is as follows: If one rotates the electrons $\overrightarrow{\boldsymbol{r}} \in \mathbb{R}^{N\times 3}$ by any rotation matrix $R \in SO(3)$, the orbital matrix changes as

$$\phi(\overrightarrow{\boldsymbol{r}} R) = \phi(\overrightarrow{\boldsymbol{r}})\boldsymbol{D}^R, \tag{26}$$

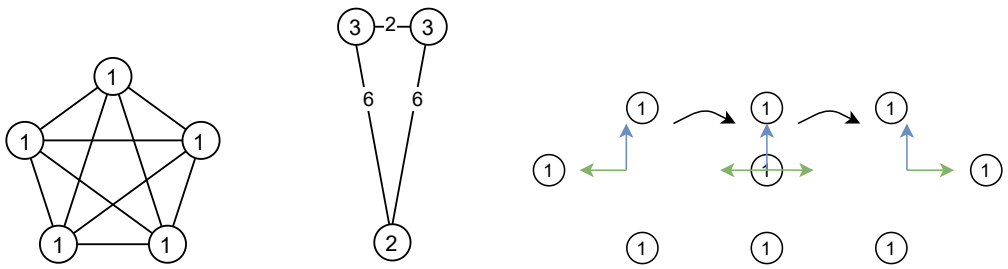
$$\boldsymbol{D}^R = \operatorname{diag}(\boldsymbol{D}_1^R, \dots, \boldsymbol{D}_E^R) \tag{27}$$

where $\boldsymbol{D}^R \in \mathbb{R}^{N\times N}$ is a block-diagonal matrix and $\boldsymbol{D}_e^R$ is the Wigner-D matrix induced by rotation R corresponding to the e-th SO(3) representation. Since Wigner-D matrices are unitary and we restrict our wave function to real-valued

$$\begin{aligned}
\psi(\overrightarrow{\boldsymbol{r}} R) &= \det \phi(\overrightarrow{\boldsymbol{r}} R) \\
&= \det(\phi(\overrightarrow{\boldsymbol{r}})\boldsymbol{D}^R) \\
&= \det \phi(\overrightarrow{\boldsymbol{r}}) \det \boldsymbol{D}^R \\
&= \det \phi(\overrightarrow{\boldsymbol{r}}) \prod_{e=1}^{E} \det \boldsymbol{D}_e^R \\
&= \det \phi(\overrightarrow{\boldsymbol{r}}) \\
&= \psi(\overrightarrow{\boldsymbol{r}}).
\end{aligned} \tag{28}$$

□

C EDGE CASES OF THE EQUIVARIANT COORDINATE SYSTEM



(a) Example of a regular polygon. For any regular polygon on a plane, two eigenvalues of the covariance matrix are going to be identical.

(b) Example of $\overrightarrow{\boldsymbol{v}} = \mathbf{0}$.

(c) Example why one can not construct a unique equivariant coordinate system that changes smoothly with changes in the geometry.

Figure 8: Edge cases in the construction of our equivariant coordinate system. Circles indicate nuclei and the numbers their charges.

While the definition of the coordinate system in Section 3.3 works in most cases. There still exist edge cases where the coordinate system may not be unique. To maximize transparency, we discuss some of these cases, when they occur, how we handle them, and what their implications are.

For certain geometries, two eigenvalues of the nuclei covariance matrix might be identical. If that is the case the PCA axes are not unique. This occurs for any geometry where the nuclei coordinates are distributed regularly on a sphere around the center of mass. Examples of such geometries are regular polygons such as the pentagram depicted in Figure 8a. In such cases, we compute the PCA on pseudo coordinates which we obtain by stretching the graph in the direction of the largest Coulomb potential $\frac{Z_m Z_n}{\|\vec{\boldsymbol{R}}_m - \vec{\boldsymbol{R}}_n\|_2}$. In the example from Figure 8a, this is equivalent to stretching the graph along one of the outer edges as they are all of the same lengths. The actual direction does not matter, as it is simply a rotation or reflection of the whole system. While regular spherical patterns are not the only case where this issue arises they are the most common cause.

Another potential issue arises if $\overrightarrow{v} = \mathbf{0}$, this occurs for any geometry which is point symmetric in the center of mass such as the pentagram in Figure 8a. In such cases, the signs of the axes do not matter as reflections result in identical geometries. However, there also exist other geometries for which Equation 12 is $\mathbf{0}$. An example is depicted in Figure 8b. But these cases are rare and can be resolved by applying a nonlinear function on the distances in Equation 12.

The occurrence of these edge cases leads us to question: 'Why can we not design a unique coordinate system for each geometry?'. While we ideally would want an equivariant coordinate system that changes smoothly with changes in the geometry, it is impossible. We show a counter example of this in Figure 8c where we see a system of 3 nuclei. In their starting configuration, one can uniquely define two axes as indicated by the colored arrows. But, when moving the leftmost nuclei such that it is in line with the other two nuclei one is left with only one uniquely defined direction as there is no way to differentiate between any orthogonal vector and the blue one. By moving the center nuclei to the right, we again can define two axes for this system, though, one axis is flipped compared to the initial state. So, we neither can define a smoothly changing coordinate system nor a unique one for every system. However, in practice, we do not need a smoothly changing one but only a unique one. While this is already unattainable as shown by the central figure, we also want to stress that any arbitrary orthogonal vector is equivalent due to the symmetries of the system. Considering these aspects, we believe that our coordinate system definition is sufficient in most scenarios.

D Experimental Setup

Hyperparameters. If not otherwise specified we used the hyperparameters from Table 2. These result in a similarly sized WFModel to FermiNet from Pfau et al. (2020). For cyclobutadiene, we did not train for the full 60000 iterations but stopped the training after 2 weeks.

Numerical Stability To stabilize the optimization, we initialize the last layers of the MetaGNN $f^{\text{out}}_{\text{node}}$ and $f^{\text{out}}_{\text{global}}$ such that the biases play the dominant role. Furthermore, we compute the final output of the WFModel in log-domain and use the log-sum-exp trick.

Learning continuous subsets. To demonstrate PESNet's ability to capture continuous subsets of the potential energy surface, we train PESNet on a dynamic set of configurations along the energy surface. Specifically, we subdivide the potential energy surface into even-sized bins and place a random walker within each bin. These walkers slightly alter the molecular structure after each step by moving within their bin. This procedure ensures that our model is not restricted to a discrete set of configurations. Therefore, after training, our model can be evaluated at any arbitrary configuration within the training domain without retraining. But, we only evaluate PESNet on configurations where reference calculations are available. This procedure is done for the hydrogen rectangle, the hydrogen chain, and the nitrogen molecule. For H_4^+ and cyclobutadiene, we train on discrete sets of geometries from the literature (Scherbela et al., 2021; Kinal & Piecuch, 2007).

Convergence is easy to detect in cases where we optimize for a fixed set of configurations as the energy will slowly converge to an optimal value, specifically, the average of the optimal energies. However, in cases where we optimize for a continuous set of configurations, the optimal energy value depends on the current batch of geometries. To still access convergence, we use the fact that the local energy E_L of any eigenfunction (including the ground-state) has 0 variance. So, our convergence criteria is the expected variance of the local energy $\mathbb{E}_{\vec{\boldsymbol{R}} \sim \text{PES}}\left[\mathbb{E}_{\vec{\boldsymbol{r}} \sim \psi^2_{\vec{\boldsymbol{R}}}}\left[\left(E_L - \mathbb{E}_{\vec{\boldsymbol{r}} \sim \psi^2_{\vec{\boldsymbol{R}}}}[E_L]\right)^2\right]\right]$ where the optimal value is 0.

	Parameter	Value
Optimization	Local energy clipping	5.0
Optimization	Batch size	4096
Optimization	Iterations	60000
Optimization	#Geometry random walker	16
Optimization	Learning rate η	$\frac{0.1}{(1+t/1000)}$
Natural Gradient	Damping	$10^{-4} \cdot \text{Std}[E_L]$
Natural Gradient	CG max steps	100
WFModel	Nuclei embedding dim	64
WFModel	Single-stream width	256
WFModel	Double-stream width	32
WFModel	#Update layers	4
WFModel	#Determinants	16
MetaGNN	#Message passings	2
MetaGNN	Embedding dim	64
MetaGNN	Message dim	32
MetaGNN	N_{SBF}	7
MetaGNN	N_{RBF}	6
MetaGNN	MLP depth	2
MCMC	Proposal step size	0.02
MCMC	Steps between updates	40
Pretraining	Iterations	2000
Pretraining	Learning rate	0.003
Pretraining	Method	UHF
Pretraining	Basis set	STO-6G
Evaluation	#Samples	10^6
Evaluation	MCMC Steps	200

Table 2: Default hyperparameters.

	H_4^+	H4	H10	N2	Cyclobutadiene
no MetaGNN	-1.849286(9)	-2.016199(5)	-5.328944(14)	-109.28322(9)	-154.64419(31)
MetaGNN	-1.849363(6)	-2.016208(5)	-5.328916(15)	-109.28570(7)	-154.65469(27)
ΔE	0.000077(11)	0.000009(7)	-0.000028(21)	0.00248(11)	0.0105(4)

Table 3: Energies in E_h averaged over the PES for PESNets with and without the MetaGNN. Numbers in brackets indicate the standard error at the last digit(s). In cases without the MetaGNN, we still train a single model for all configurations of a system.

E Ablation Studies

As hyperparameters often play a significant in the machine learning community, we present some ablation studies in this appendix. All the following experiments only alter one variable at a time, while the rest is fixed as in Table 2. The results in the tables are averaged over the same configurations as in the main body.

Table 3 presents results with and without the MetaGNN. It is noticeable that the gain of the MetaGNN is little to none for small molecules consisting of simple hydrogen atoms. But, for more

#Dets	H_4^+	H4	H10	N2	Cyclobutadiene
16	-1.849363(6)	-2.016208(5)	-5.328916(15)	-109.28570(7)	-154.65469(27)
32	-1.849342(4)	-2.016188(6)	-5.328999(13)	-109.28706(7)	-154.65322(27)
ΔE	-0.000021(7)	-0.000019(8)	0.000083(20)	0.00136(10)	-0.0015(4)

Table 4: Energies in E_h averaged over the PES for different number of determinants in our PESNet model. Numbers in brackets indicate the standard error at the last digit(s).

$\dim(\boldsymbol{h}_i^t)$	H_4^+	H4	H10	N2	Cyclobutadiene
256	-1.849363(6)	-2.016208(5)	-5.328916(15)	-109.28570(7)	-154.65469(27)
512	-1.8493543(28)	-2.016190(7)	-5.328794(17)	-109.28662(6)	-154.65042(28)
ΔE	-0.000009(7)	-0.000017(8)	-0.000122(23)	0.00092(9)	-0.0043(4)

Table 5: Energies in E_h averaged over the PES for different single-stream sizes in our PESNet model. Numbers in brackets indicate the standard error at the last digit(s).

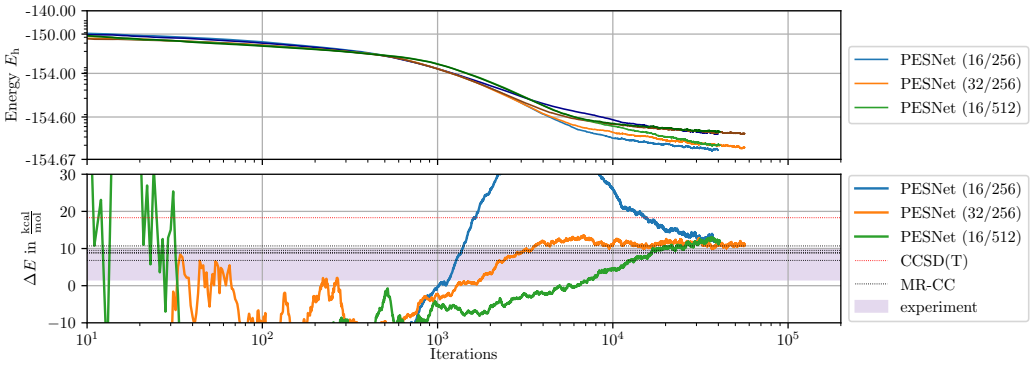


Figure 9: Comparison of different PESNet configurations on cyclobutadiene. The configurations are named (#determinant/single-stream width) with light colors for the ground state and darker colors for the transition state.

complex molecules such as nitrogen and cyclobutadiene, we notice significant improvements of $2.5\,\mathrm{m}E_\mathrm{h}$ and $10.5\,\mathrm{m}E_\mathrm{h}$, respectively. Moreover, the MetaGNN enables us to account for symmetries of the energy while the WFModel itself is only invariant w.r.t. to translation but not to rotation, reflection, and reindexing of nuclei.

Table 4 shows the impact of the number of determinants on the average energy for the systems from Section 4. For small hydrogen-based systems, the number of determinants is mostly irrelevant, while larger numbers of determinants improve performance for nitrogen. But, this does not seem to carry over to cyclobutadiene. While the total estimated energy is higher for the larger model, we noticed that it is significantly faster at converging the transition barrier. Figure 9 illustrates this by comparing the convergence of the 16 and 32 determinant models.

Increasing the single-stream size does not seem to result in any benefit for most models as Table 5 shows. Again, the hydrogen systems are mostly unaffected by this hyperparameter while nitrogen benefits from larger hidden dimensions but cyclobutadiene converges worse. We suspect that this is due to the optimization problem becoming significantly harder. Firstly, increasing the WFModel increases the number of parameters the MetaGNN has to predict. Secondly, we estimate the inverse of the Fisher with a finite fixed-sized batch but the Fisher grows quadratically with the number of parameters which also grow quadratically with the single size stream.

F Time per Iteration

While the main document already covers the time it took to reproduce the results from the figures, we want to use this appendix to provide more details on this. Table 6 lists the time per iteration for a single model instead of the whole training time to reproduce the potential energy surfaces as in Table 1. While we find these numbers to be misleading we still want to disclose them to support open research. The main issue with these numbers is that they do not take the quality of the update into account, e.g., PauliNet and DeepErwin are trained with Adam (Kingma & Ba, 2014), FermiNet with K-FAC (Martens & Grosse, 2015), and PESNet with CG-computed natural gradient (Neuscamman et al., 2012). This has implications on the number of iterations one has to train. For Table 1 in the main body, we assumed that PauliNet is trained for 10000 iterations (Hermann et al., 2020), DeepErwin for 7000 iterations (Scherbela et al., 2021), FermiNet for 200000 iterations (Pfau et al., 2020),

	H_4^+	H4	H10	N2	Cyclobutadiene
PauliNet	0.83s	1.13s	5.51s	8.09s	175s
DeepErwin	0.92s	1.28s	4.88s	—	—
FermiNet	0.12s	0.19s	1.07s	1.99s	20.8s
PESNet	1.19s	1.42s	3.91s	5.32s	33.2s

Table 6: Time per training step.

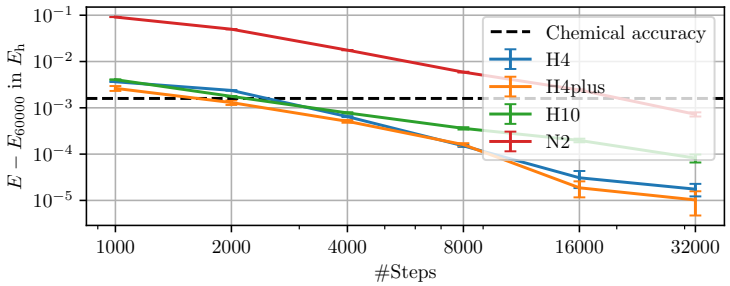


Figure 10: Convergence behavior of PESNet. Error bars indicate the standard error of the mean.

and PESNet for 60000 iterations. Though, one can assume that neither PauliNet nor DeepErwin are going to produce similar results to FermiNet and PESNet on the more complex nitrogen molecule or cyclobutadiene in 10000 or 7000 iterations. This is further discussed in Appendix G. When viewing these results, one also has to keep in mind the different quality of the results, e.g., FermiNet and PESNet strictly outperform PauliNet and DeepErwin on all systems. Another potential issue arises due to the choice of deep learning framework the models are implemented in. It has been shown that JAX works very well for computing the kinetic energy (Spencer et al., 2020) which usually is the largest workload when computing an update.

G CONVERGENCE

When choosing a method to solve the Schrödinger equation one has to pick between accuracy and speed. For classical methods, this might be the difference between choosing DFT, CCSD(T), or FCI. In the neural VMC setting, one way to reflect this is by choosing the number of training steps. To better investigate this, we present convergence graphs for H4, H_4^+, H10, and N2 in Figure 10. For cyclobutadiene, we can see the convergence of different configurations of PESNet in Figure 9. One can see that our network converges quite quickly on hydrogen-based systems such as the hydrogen rectangle, H_4^+, or the hydrogen chain. For these small systems, PESNet surpasses PauliNet's and DeepErwin's accuracy in less than 8000 training steps, reducing the training times to 2.9h, 3.2h, and 8.7h, respectively. In less than 16000 steps, about 17.4h of training, PESNet surpasses FermiNet on the hydrogen chain. For more complicated molecules such as N2 and cyclobutadiene, the training requires more iterations to converge. So, we may also expect the extrapolated numbers from Table 1 for PauliNet to be an optimistic estimate.

B *Publication 2*: Generalizing Neural Wave Functions

Publication: Nicholas Gao and Stephan Günnemann. Generalizing Neural Wave Functions. In *International Conference on Machine Learning*, 2023a

Location: Honolulu, USA

License: The rights of the publication are held by the authors; from the respective web page `https://icml.cc/FAQ/Copyright`, June 9th 2025:

> Authors do not tranfer the copyright of their paper to ICML, instead they grant ICML a non-exclusive, perpetual, royalty-free, fully-paid, fully-assignable license to copy, distribute and publicly display all or part of the paper.

Therefore, no explicit permission is required to include the publications in this thesis.

Generalizing Neural Wave Functions

Nicholas Gao [1] **Stephan Günnemann** [1]

Abstract

Recent neural network-based wave functions have achieved state-of-the-art accuracies in modeling *ab-initio* ground-state potential energy surface. However, these networks can only solve different spatial arrangements of the same set of atoms. To overcome this limitation, we present Graph-learned Orbital Embeddings (Globe), a neural network-based reparametrization method that can adapt neural wave functions to different molecules. Globe learns representations of local electronic structures that generalize across molecules via spatial message passing by connecting molecular orbitals to covalent bonds. Further, we propose a size-consistent wave function Ansatz, the Molecular Orbital Network (Moon), tailored to jointly solve Schrödinger equations of different molecules. In our experiments, we find Moon converging in 4.5 times fewer steps to similar accuracy as previous methods or to lower energies given the same time. Further, our analysis shows that Moon's energy estimate scales additively with increased system sizes, unlike previous work where we observe divergence. In both computational chemistry and machine learning, we are the first to demonstrate that a single wave function can solve the Schrödinger equation of molecules with different atoms jointly.

1. Introduction

In silico design of molecules requires accessing their quantum mechanical properties. This requires solving the associated Schrödinger equation. However, exact solutions are often intractable and approximations can become computationally expensive for larger and more complex systems. In recent years, neural network-based wave functions have emerged as a promising alternative, providing accurate and well-scaling approximation solutions in $O(N^4)$ with the number of electrons N (Hermann et al., 2022). Despite their theoretical scaling, this time complexity comes with a large prefactor leading to exploding computational requirements if one screens many different molecules. To address this limitation, Gao & Günnemann (2022) proposed the Potential Energy Surface Network (PESNet), a neural network wave function that generalizes across different structures. While reducing computational costs, PESNet is limited to different spatial arrangements of the same set of atoms.

A key challenge in the generalization to arbitrary molecules is the variable number of molecular orbitals. To resolve this, we introduce Graph-learned Orbital Embeddings (Globe), a generalization of PESNet that can solve arbitrary Schrödinger equations jointly. Like previous work, Globe uses a two-level approach where one network represents the electronic wave function and the other reparametrizes the wave function depending on the molecule. We resolve the issue of the dynamical numbers of molecular orbitals by embedding orbitals in 3D space. This enables us to learn local electronic structures via spatial message passing in graph neural networks (GNNs). For the wave function, we present the Molecular Orbital Network (Moon), the first size-consistent neural wave function. We accomplish size consistency in two key steps, firstly, by using spatial message passing Moon focuses on local interactions, and, secondly, by using the nuclei as anchor points for message passing. While the first step is strictly required for size consistency, the latter enables efficient reparametrization via Globe.

In our experiments, we find Moon accelerating convergence in joint training by up to 4.5 times and performing similarly to the attention-based PsiFormer on larger systems (von Glehn et al., 2023). Further, we observe that transfers of neural wave functions to larger structures do not require additional self-consistent field (SCF) calculations. In summary, our main contributions are:[1]

- **Globe**, a reparametrization method for adapting neural wave functions to arbitrary molecules based on localized molecular orbital embeddings.
- **Moon**, a size-consistent neural wave function enabling generalization to larger structures, faster convergence, and accurate energies.

[1]Department of Computer Science & Munich Data Science Institute, Technical University of Munich, Germany. Correspondence to: Nicholas gao <n.gao@tum.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]Source code: https://www.cs.cit.tum.de/daml/globe/

2. Background

Notation. We use the term molecule for a point cloud in $\mathbb{R}^3$ with charges assigned to each node. The term 'geometry' refers to different spatial arrangements associated with the same set of charges. We use N to denote the number of electrons and M for the number of nuclei. $\mathbf{r} \in \mathbb{R}^{N\times 3}$ denotes a complete electron configuration whereas $\boldsymbol{r} \in \mathbb{R}^3$ denotes a single electron's position. For nuclei, we use $\mathbf{R} \in \mathbb{R}^{M\times 3}$ and $\boldsymbol{R} \in \mathbb{R}^3$, respectively. Z_m denotes the charge of the mth nucleus. We use $[\ ]$ for the concatenation of vectors, $\circ$ for the Hadamard product, $\|x\|$ for the L_2-norm, bold capital letters $\boldsymbol{A}$ for matrices, bold lower case letters $\boldsymbol{a}$ for vectors, and normal face letters a for scalars. Bracketed superscripts$^{(l)}$ index sequences, e.g., layers in a neural network.

2.1. Quantum chemistry

At the heart of quantum chemistry is the Schrödinger equation. Its time-independent form is

$$\boldsymbol{H}\psi = E\psi \tag{1}$$

where $\psi : \mathbb{R}^{N\times 3} \to \mathbb{R}$ is the electronic wave function, E the energy and the Hamiltonian operator is

$$\boldsymbol{H} = -\frac{1}{2}\sum_{n=1}^{3N}\nabla_n^2 + V(\mathbf{r}), \tag{2}$$

$$V(\mathbf{r}) = \sum_{n>m=1}^{N}\frac{1}{\|r_n - r_m\|} - \sum_{n=1}^{N}\sum_{m=1}^{M}\frac{Z_m}{\|r_n - R_m\|} + \sum_{m>n=1}^{M}\frac{Z_m Z_n}{\|R_m - R_n\|}, \tag{3}$$

within the Born-Oppenheimer approximation, i.e., we approximate nuclei as particles with fixed positions. In linear algebra, Equation (1) is an eigenvalue problem where one wants to find the eigenfunction ψ_0 associated with the lowest eigenvalue E_0. These are commonly called the ground-state wave function and energy, respectively.

The electronic wave function ψ describes the behavior of electrons. Note that electrons are not only specified by their spatial location $\boldsymbol{r} \in \mathbb{R}^3$ but also by their spin $\alpha \in \{\uparrow, \downarrow\}$. Though, since the spins do not occur in the Hamiltonian, they can be fixed a priori (Foulkes et al., 2001). For a function to be a valid wave function, it must satisfy two criteria. First, ψ must obey the Fermi-Dirac statistics, i.e., it must be antisymmetric $\psi(\mathbf{r}) = \mathrm{sign}(\pi)\psi(\pi(\mathbf{r}))$ w.r.t. permutations of same-spin electrons π. Second, the integral of its square must be one $\int \psi(\mathbf{r})^2 d\mathbf{r} = 1$.

The challenge in computational chemistry is accurately approximating the ground-state energy. For instance, the total energy of a system can be decomposed into a mean-field energy and correlation energy, where the mean-field energy accounts for $\approx 99.5\%$ of the total energy. To reach chemical accuracy (typically defined as $1\,\mathrm{kcal\,mol^{-1}}$), one has to accurately estimate $> 99\%$ of the correlation energy, $> 99.999\%$ of the total energy.

Most commonly, the wave function is represented by a determinant of molecular orbital functions (Slater, 1929):

$$\psi(\mathbf{r}) = \det\Phi, \qquad \Phi_{ij} = \phi_j(r_i) \tag{4}$$

where the determinant ensures the antisymmetry w.r.t. permutations. The Hartree-Fock (HF) method provides a simple mean-field approximate solution to the Schrödinger equation where the molecular orbital functions $\phi_i^{\mathrm{HF}} : \mathbb{R}^3 \to \mathbb{R}$ are constructed with Linear Combinations of Atomic Orbitals (LCAO) $\varphi_j : \mathbb{R}^3 \to \mathbb{R}$ (Lennard-Jones, 1929), $\phi_i^{\mathrm{HF}}(x) = \sum_{m=1}^{M}\sum_{n=1}^{O_m}\omega_{i,m,n}\varphi_{m,n}(x)$ with O_m being the number of atomic orbitals and $\varphi_{m,n}$ being the nth atomic orbital function of the mth atom, respectively. In matrix notation, Equation (4) can be written as

$$\psi(\mathbf{r}) = \det\Phi = \det(\Phi\Omega^T) \qquad \Phi, \Omega \in \mathbb{R}^{N\times\eta} \tag{5}$$

with $\eta = \sum_{m=1}^{M} O_m$, Φ being a matrix of all atomic orbital functions evaluated at every electron position, and Ω being an optimized weight matrix.

2.2. Variational Monte Carlo

In Variational Monte Carlo (VMC), one approximates a solution to Equation (1) by picking a trial wave function ψ_θ parametrized by θ and iteratively minimizing the energy via gradient descent on θ. Since the eigenfunctions of $\boldsymbol{H}$ are a complete basis, this variational optimization is an upper bound to the true ground-state energy. By reformulating Equation (1), one gets

$$E = \frac{\int \psi_\theta(\mathbf{r})\boldsymbol{H}\psi_\theta(\mathbf{r})d\mathbf{r}}{\int \psi_\theta^2(\mathbf{r})d\mathbf{r}} \tag{6}$$

$$= \mathbb{E}_{\mathbf{r}\sim\psi_\theta^2}\left[\psi_\theta(\mathbf{r})^{-1}\boldsymbol{H}\psi_\theta(\mathbf{r})\right] = \mathbb{E}_{\mathbf{r}\sim\psi_\theta^2}\left[E_\theta(\mathbf{r})\right]. \tag{7}$$

In contrast to Equation (1), we here assumed an unnormalized wave function ψ_θ, thus, the normalization factor. Further, we reformulate the integral in the second line using importance sampling. E_θ is the so-called local energy:

$$E_\theta(\mathbf{r}) = \psi_\theta(\mathbf{r})^{-1}\boldsymbol{H}\psi_\theta(\mathbf{r}) \tag{8}$$

$$= -\frac{1}{2}\sum_{i=1}^{3N}\left[\frac{\partial^2\log|\psi_\theta(\mathbf{r})|}{\partial\mathbf{r}_i^2} + \frac{\partial\log|\psi_\theta(\mathbf{r})|}{\partial\mathbf{r}_i}^2\right] + V(\mathbf{r}). \tag{9}$$

Finally, one optimize ψ_θ via gradient descent with

$$\nabla_\theta E = \mathbb{E}_{\mathbf{r}\sim\psi_\theta^2}\left[\left[E_\theta(\mathbf{r}) - \mathbb{E}_{\mathbf{r}\sim\psi_\theta^2}\left[E_\theta(\mathbf{r})\right]\right]\nabla_\theta\log\psi_\theta(\mathbf{r})\right] \tag{10}$$

where we estimate all expectations with Monte Carlo estimates using Metropolis-Hastings (Ceperley et al., 1977).

Due to the few constraints imposed on wave functions, recent works used neural networks to model them (Pfau et al., 2020; Hermann et al., 2020). In the neural network setting, learnable many-electron orbital functions $\phi_i : \mathbb{R}^3 \times \mathbb{R}^{N\times 3} \to \mathbb{R}$ implemented by permutation equivariant neural networks replace the single-electron molecular orbital functions ϕ_i^{HF} in Equation (4).

3. Related Work

Traditional methods for modeling electronic wave functions have relied on Linear Combinations of Atomic Orbitals (LCAO) (Lennard-Jones, 1929) arranged in a Slater determinant (Slater, 1929). However, they cannot capture electron-electron interactions beyond a mean-field approximation. To address this issue, backflow transformations (Feynman & Cohen, 1956) and Jastrow factors (Jastrow, 1955) have been introduced. Later, Carleo & Troyer (2017) were the first to demonstrate the use of neural networks to model to quantum systems, though only for discrete spin systems. This approach has since been improved upon by using deep neural networks for real-space electronic systems (Han et al., 2019; Pfau et al., 2020; Hermann et al., 2020). In subsequent works, such neural networks have further refined (Gerard et al., 2022; von Glehn et al., 2023) and adopted to different settings like pseudopotentials (Li et al., 2022a), periodic systems (Wilson et al., 2022; Li et al., 2022b; Cassella et al., 2023) or diffusion Monte Carlo (DMC) (Wilson et al., 2021; Ren et al., 2022).

Despite their high accuracy, neural network-based wave function models are still inherently expensive for multiple systems. Two recent concurrent approaches addressed this challenge: DeepErwin, a weight-sharing method across geometries (Scherbela et al., 2022), and PESNet (Gao & Günnemann, 2022; 2023), a two-network approach that allows for joint training of several geometries, eliminating the need for retraining. But, while the former needs retraining for each structure, the latter is limited to different spatial arrangements of the same set of atoms.

4. Generalizing Neural Wave Functions

Compared to different geometries, generalization across different molecules comes with additional difficulties as the number of atoms, electrons, and orbitals change. To address these challenges, we identify two key desiderata that such a system should fulfill:

1. *Invariance:* A molecule's energy is invariant to Euclidean transformations and permutations. Thus, a generalizing wave function should result in invariant energy estimates. To achieve this, the wave function must be equivariant to Euclidean transformations and nuclei permutation (Gao & Günnemann, 2022).
2. *Size consistency:* As most quantum mechanical interactions happen within a short distance, a molecule's energy is an extensive quantity and scales additively with its size. For wave functions, this implies that the wave function decomposes into a product of the individual wave functions for distant molecules.

To incorporate Euclidean symmetries, we follow Gao & Günnemann (2022) by defining a PCA-based equivariant coordinate frame. Thus, in the following all references to the electron $\mathbf{r}$ and nuclei $\mathbf{R}$ positions are measured in the equivariant frame. To accomplish size consistency, the wave function must decompose into a product of two wave functions if two systems are sufficiently separated. In Appendix A, we show that decaying the value of molecular orbital functions ϕ_i to 0 far from the involved atoms is sufficient to implement this. We achieve this by relying on local interactions between pairs of particles (electrons/atoms) that exponentially decay with distance.

Like previous work, we adopt a two-network approach. While Moon represents the electronic wave function ψ_θ, Globe acts solely on the nuclei and adapts the former's parameters to the molecule.

4.1. Graph-learned Orbital Embeddings (Globe)

Globe's task is to reparametrize the wave function depending on the molecular structure, i.e., it only acts on the atoms and does not consider electrons. To perform such a reparametrization, it must extract local electronic structure information from the atomic point cloud. Further, we must parametrize N molecular orbital functions ϕ_i, see Equation (4), which poses a challenge as their number depends on the number of electrons rather than atoms. As illustrated in Figure 1, we achieve both in a three-step procedure. First, we learn about atomic neighborhoods via message passing. Next, we localize orbitals and, finally, learn orbital embeddings via unidirectional message passing.

Message-passing network. Our message-passing network relies on the use of continuous filter convolutions (Schütt et al., 2018). We initialize the node embeddings by a charge embedding $\boldsymbol{h}_i^{\text{atom}(0)} = \boldsymbol{F}_{Z_i}^{\text{atom}}$ and iteratively update them through message-passing as

$$\boldsymbol{h}_i^{(l+1)} = f^{(l)}(\boldsymbol{h}_i^{(l)}, \boldsymbol{m}_i^{(l)}), \tag{11}$$

$$\boldsymbol{m}_i^{(l)} = \frac{1}{\nu_{\boldsymbol{R}_i}^{\mathbf{R}}} \sum_{j=1}^{M} g^{(l)}(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}) \circ \Gamma^{(l)}(\boldsymbol{R}_i - \boldsymbol{R}_j), \tag{12}$$

$$\nu_x^{\mathcal{N}} = 1 + \sum_{y\in\mathcal{N}} \exp\left(-\frac{\|x-y\|^2}{\sigma_{\text{norm}}^2}\right) \tag{13}$$

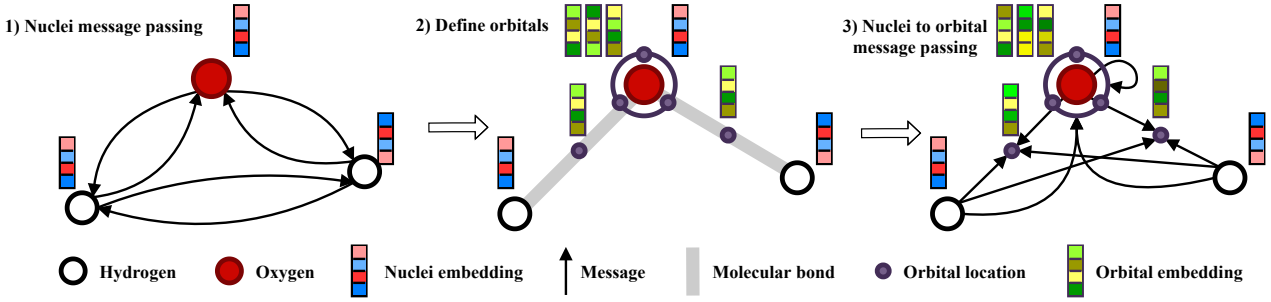


Figure 1. Globe steps. 1) Atom embeddings are obtained by message passing between atoms. 2) Orbital locations and embeddings are determined by core orbitals and molecular bonds. 3) Orbital embeddings are updated via a unidirectional message passing. For clarity, we omitted messages from each atom to each of the three core orbitals.

where $f^{(l)}$ and $g^{(l)}$ are implemented by MLPs, Γ are spatial filters and ν is a spatial normalization with σ_{norm} being a learnable parameter. By multiplying elementwise with spatial filters rather than concatenating with them, as done in Gao & Günnemann (2022), we decay long-range interactions between atoms and strengthen local interactions. Further, instead of averaging over all atoms, we normalize the message by a learnable normalization factor to account for the size of its neighborhood.

Spatial filters. To model arbitrary wave functions, we must break euclidean symmetries in our reparametrization (Gao & Günnemann, 2022). Previous work used positional encodings relative to the center of mass to achieve this. But, as the center of mass is an inherently global property, we instead break the symmetries in our spatial filters enforcing locality. Instead of being radial, our filters operate on the full three-dimensional space by constructing them as a Hadamard product of a Gaussian envelope and an MLP on the three-dimensional input:

$$\Gamma^{(l)}(\boldsymbol{x}) = \boldsymbol{W}^{(l)}\beta(\boldsymbol{x}), \tag{14}$$

$$\beta(\boldsymbol{x}) = \boldsymbol{W}^{\text{env}}\left[\exp\left(-\left(\frac{\|\boldsymbol{x}\|}{\varsigma_i}\right)^2\right)\right]_{i=1}^{D} \circ \left(\sigma\left(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}\right) \tag{15}$$

with D being the number of envelope ranges ς_i, and σ being an activation function. While a combination of spherical harmonics and radial basis functions achieves similar symmetry breaking (Gasteiger et al., 2021; Zitnick et al., 2022), we found such freely learnable filters to perform better.

Orbital localization. A key challenge in adapting a wave function to arbitrary molecules is the molecular orbital functions ϕ_i as their number is not fully specified by the number of atoms but by the number of electrons. While generating one molecular orbital function per electron seems like an intuitive solution, this would cause two rows *and* columns in Equation (4) to permute if two electrons permute, resulting in a permutation symmetric rather than an antisymmetric function. Thus, the orbital functions must be independent of the actual electrons. One could generate the orbitals by a global graph embedding, e.g., via an RNN, but such a construction does not preserve locality and behaves unpredictably to changes in the nuclei.

We avoid such global constructions, by assigning each molecular orbital a location $\boldsymbol{L}_i \in \mathbb{R}^3$ and learning the parameters of the associated orbital function ϕ_i via message passing. To localize the orbitals, we distinguish between core and valence orbitals as core orbitals tend to interact little with other atoms (Foulkes et al., 2001). For each atom type, we define its valency by the number of bonds it can form, e.g., for hydrogen one, for carbon four, for oxygen two, etc. The number of core orbitals for the ith atom is then $\frac{Z_i - V_i}{2}$ with V_i being the valency of the ith atom. These core orbitals are located at the same location as the nuclei. To determine the valence orbitals, we identify covalent bonds and locate the orbital in the center of that bond. We do this by iteratively picking the pairs of atoms closest to each other where each atom has at least one unpaired electron left. An example of our localized orbitals is depicted in Figure 1. In Appendix B, we provide a full definition of the algorithm.

While obtaining the orbital locations $\boldsymbol{L}_i$, we also define their types T_i. Where the order and the charge of the nucleus define the core orbital types, the bond's cardinality defines the valence orbital types. This categorial distinction avoids identical embeddings for two orbitals at the same location.

Orbital embedding. We initialize the orbital embeddings via their types $\boldsymbol{h}_i^{\text{o}(0)} = \boldsymbol{F}^{\text{o}}_{T_i}$ and iteratively update them with an architecturally identical GNN as used for the atoms but with unidirectional message passing from the atoms to the orbitals. Here, we avoid bidirectional message passing as the orbital structure is wholly inferred from the nuclei and, thus, carries no additional geometric information.

Parameter estimation. Changes in a molecule's structure manifest in wave function parameters that depend either on atoms, orbitals, or a combination of both. Thus, Globe updates these parameters via their respective embeddings $(\boldsymbol{h}^{\text{a}}, \boldsymbol{h}^{\text{o}}, \boldsymbol{h}^{\text{a-o}})$. Before generating parameters via individ-

ual MLPs, we pass them through shared MLPs and LayerNorms (Ba et al., 2016). To ensure that the wave function converges to a product for distant systems (Desiderata 2), we define atom-orbital embeddings as

$$\boldsymbol{h}_{i,m}^{\text{a-o}} = \left[\boldsymbol{h}_m^{\text{a}(L)}, \boldsymbol{h}_i^{\text{o}(L)}\right] \boldsymbol{W} \circ \Gamma^{\text{a-o}}(\boldsymbol{R}_m - \boldsymbol{L}_i). \tag{16}$$

where the spatial filters vanish the contribution of the mth atom to the ith orbital with increasing distances.

4.2. Molecular Orbital Network (Moon)

Moon represents the electronic wave function ψ_θ, but, unlike previous work, Moon encourages local interactions via spatial message passing and avoids strong global interactions (Pfau et al., 2020; von Glehn et al., 2023). While PauliNet already represents a GNN-based wave function, it relies heavily on HF calculations and does not reach similar accuracy (Hermann et al., 2020; Gerard et al., 2022).

To avoid many expensive message-passing steps between electrons and nuclei, we simplify the message-passing structure. Figure 2 provides a conceptual overview of Moon. First, we encode the local electronic neighborhood for each electron. Next, nuclei aggregate electronic structure information. The nuclei embeddings are then iteratively updated and, lastly, diffused to the electrons. The basic functional form of Moon follows a Slater-Jastrow wave function

$$\psi_\theta(\mathbf{r}) = \exp(J(\mathbf{r})) \sum_{k=1}^{K} w_k \det \Phi^k, \tag{17}$$

i.e., a product of a permutation invariant Jastrow factor and a weighted sum of Slater determinants. As Jastrow factor, we additively combine the Jastrow factors from Gao & Günnemann (2023) and von Glehn et al. (2023). In the following, we use bars $\bar{a}$ for atom parameters, tildes $\tilde{a}$ for orbital parameters, and both $\tilde{\bar{a}}$ for atom-orbital interaction parameters. These are the parameters that are updated through Globe. For clarity, we omit the Jastrow factor, residual connections, and normalization coefficient definitions here and refer the reader to Appendix C for detailed descriptions.

Embedding. As initial features, we use the pairwise distances between electrons and nuclei $\boldsymbol{g}_{ij}^{\text{e-n}} = [\boldsymbol{r}_i - \boldsymbol{R}_j, \|\boldsymbol{r}_i - \boldsymbol{R}_j\|]$, and electron and electrons $\boldsymbol{g}_{ij}^{\text{e-e}} = [\boldsymbol{r}_i - \boldsymbol{r}_j, \|\boldsymbol{r}_i - \boldsymbol{R}_j\|]$. As illustrated in Figure 2, we initialize the electron embeddings by a single electron-electron message-passing step

$$\boldsymbol{h}_i^{\text{e}(0)} = \sum_{j=1}^{N} \sigma\left(\boldsymbol{g}_{ij}^{\text{e-e}} \boldsymbol{W}^{\delta_{\alpha_i}^{\alpha_j}}\right) \circ \Gamma^{\delta_{\alpha_i}^{\alpha_j}}(\|\boldsymbol{r}_i - \boldsymbol{r}_j) \| \boldsymbol{W} \tag{18}$$

where functions and matrices superscripted by the Kronecker delta $\delta_{\alpha_i}^{\alpha_j}$ indicate different weights. Here, we again use the spatial filters to decay interactions from far-apart particles. Like Gao & Günnemann (2022), we construct electron-nuclei interaction embeddings by combining electron embeddings $\boldsymbol{h}_i^{\text{e}(0)}$, nuclei embeddings $\bar{\boldsymbol{z}}_m$, and their distance $\boldsymbol{g}_{im}^{\text{e-n}}$ via

$$\boldsymbol{h}_{im}^{\text{e-n}(0)} = \sigma\left(\boldsymbol{h}_i^{\text{e}(0)} + \bar{\boldsymbol{z}}_m + \boldsymbol{g}_{im}^{\text{e-n}} \bar{\boldsymbol{W}}_m\right). \tag{19}$$

As the second step in Figure 2, these embeddings are then aggregated towards electrons and nuclei via spatial message passing while keeping separate embeddings for each spin state $\alpha \in \{\uparrow, \downarrow\}$ per nuclei:

$$\boldsymbol{h}_m^{\text{n}\alpha(1)} = \sum_{i \in \mathbb{A}^\alpha} \boldsymbol{h}_{i,m}^{\text{e-n}} \circ \bar{\Gamma}_m^{\text{n}}(\boldsymbol{r}_i - \boldsymbol{R}_m), \tag{20}$$

$$\boldsymbol{h}_i^{\text{e}(1)} = \sum_{m=1}^{M} \boldsymbol{h}_{i,m}^{\text{e-n}} \circ \bar{\Gamma}_m^{\text{e}}(\boldsymbol{r}_i - \boldsymbol{R}_m) \tag{21}$$

where $\mathbb{A}^\alpha$ is the index set of electrons with spin α and $\bar{\Gamma}$ being the spatial filters from Equation (14) with atom-parameters, see Appendix C. By using message passing instead of concatenation as commonly done in single-molecule works (von Glehn et al., 2023), we achieve invariance to nuclei permutations, see Desiderata 1 in Section 4.

Update. We iteratively update the nuclei embeddings

$$\boldsymbol{h}_m^{\text{n}\alpha(l+1)} = \boldsymbol{h}_m^{\text{n}\alpha(l)} + \sigma([\boldsymbol{h}_m^{\text{n}\alpha(l)}, \boldsymbol{h}_m^{\text{n}\hat{\alpha}(l)}]\boldsymbol{W}^{(l)} + \boldsymbol{b}^{(l)}) \tag{22}$$

where $\hat{\alpha}$ denotes the opposing spin of α. For efficiency reasons, we do not perform message passing between nuclei here as we found it to have no significant impact.

Diffusion. After L-many update steps, a single message-passing step diffuses the nuclei embeddings to the electrons

$$\boldsymbol{h}_i^{\text{e}(L)} = \sigma(\boldsymbol{h}_i^{\text{e}(0)} \boldsymbol{W} + \boldsymbol{m}_i), \tag{23}$$

$$\begin{aligned} \boldsymbol{m}_i = \sum_{m=1}^{M} &\left(\left[\boldsymbol{h}_m^{\text{n}\alpha_i(L)}, \boldsymbol{h}_m^{\text{n}\hat{\alpha}_i(L)}\right] \boldsymbol{W} + \boldsymbol{b}\right) \\ &\circ \bar{\Gamma}_m^{\text{diff}}(\boldsymbol{r}_i - \boldsymbol{R}_m) \end{aligned} \tag{24}$$

with α_i denoting the spin of the ith electron. The spatial filters in this step enable the network to learn different directional messages which are important in modeling directional wave functions, e.g., the excited states of the hydrogen atom.

Orbital construction. After diffusion, we construct restricted orbitals like Gao & Günnemann (2023) with adaptive orbital and envelope parameters:

$$\begin{aligned} \phi_i^k(\boldsymbol{r}_j) = &\left((\tilde{w}_i^{k\delta_{\alpha_i}^{\alpha_j}})^T \boldsymbol{h}_j^{\text{e}(L)} + \tilde{b}_j^{k\delta_{\alpha_i}^{\alpha_j}}\right) \\ &\sum_{m=1}^{M} \tilde{\bar{\pi}}_{im}^{k\delta_{\alpha_i}^{\alpha_j}} \exp(-\tilde{\bar{\sigma}}_{im}^{k\delta_{\alpha_i}^{\alpha_j}} \|\boldsymbol{r}_j - \boldsymbol{R}_m\|). \end{aligned} \tag{25}$$

Here, the exponential envelope from Spencer et al. (2020) guarantees that our wave function will have a finite integral. In Appendix C, we describe how we restrict the envelope parameters σ such that we fulfill size consistency desiderata.

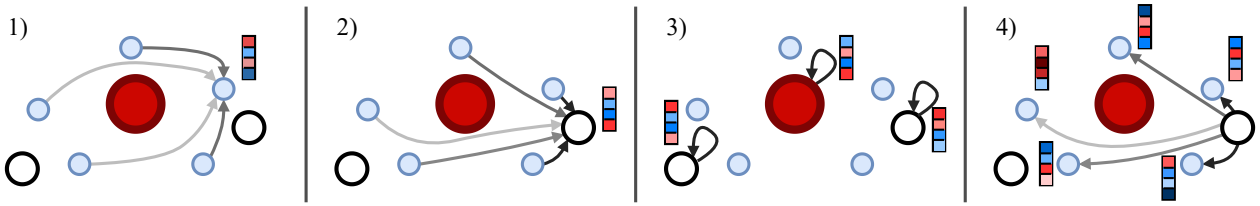


Figure 2. Illustration of Moon. 1) We initialize electron embeddings by aggregating their local neighborhood of electrons. 2) Nuclei aggregate electron embeddings via message passing. 3) Nuclei embeddings are iteratively updated. 4) Nuclei embeddings are structurally diffused towards the electrons via message passing. For clarity, we omitted most messages in 1), 2), and 4).

4.3. Optimization

We train the whole network end-to-end in a two-step procedure. We first pretrain the orbitals ϕ_i on HF solutions and, next, perform variational optimization (Pfau et al., 2020).

Pretraining. Pretraining is important to ensure a stable variational optimization (Pfau et al., 2020; von Glehn et al., 2023). Traditionally, one would match the neural network's orbitals ϕ_i with those of an HF solution ϕ_i^{HF}. But, with our localized orbitals this may cause a mismatch between nuclei involved in the ith neural orbital function and HF orbital function as the HF solution is typically sorted by energy state rather than locality. We resolve this issue by noticing that the HF wave function does not change if one multiplies the orbital matrix $\Phi = \Phi\Omega^T$ by a matrix $\boldsymbol{A}$ with unit determinant, i.e., $\psi(\mathbf{r}) = \det(\Phi\Omega^T) = \det(\Phi\Omega^T\boldsymbol{A})$. With this in mind, we can find a matrix $\boldsymbol{A}$ such that the new coefficient matrix $\hat{\Omega}^T = \Omega^T\boldsymbol{A}$ enforces locality. We describe this optimization procedure in Appendix D. After finding $\hat{\Omega}^T$, we perform traditional pretraining by matching the neural network orbitals to the localized HF orbitals. To avoid overfitting, we add a regularization loss on the outputs of the reparametrization network detailed in Appendix E.

Variational optimization. Like Gao & Günnemann (2022), we train both the wave function and the reparametrization network end-to-end and precondition the VMC gradients with natural gradient descent. But, since we deal with molecules of varying sizes unlike previous work, the gradients obtained from the different molecules may vary by orders of magnitude (like their energy). To avoid larger molecules from dominating the gradients, we rescale the gradients based on the standard deviation of the local energies associated with the molecule. We discuss this rescaling in Appendix F and the full VMC optimization in Appendix G.

4.4. Limitations

While Globe can learn a generalized wave function across different geometries of molecules, there are limitations. Firstly, despite having a rotation equivariant wave function, Globe is not smooth under arbitrary geometric perturbations. For instance, changes that cause the equivariant frame from Gao & Günnemann (2022) to flip result in discrete changes in the wave function, similar discontinuities may happen to our orbital localization as discussed in Appendix B. The coordinate frame also breaks the size-consistency of Moon as the frames of the two molecules do not necessarily align anymore. These are general issues introduced by natural symmetries that one must break to model arbitrary wave functions. Secondly, while we found our gradient rescaling based on the energy's standard deviation easing optimization, we found it to be insufficient if the discrepancy between molecules is large. For instance, if one trains a hydrogen-based system jointly with heavier atoms like nitrogen we found the optimization resulting in worse results than one obtains from training solely on the hydrogen-based system. Lastly, in its current state, the number of electrons is equal to the sum of atomic charges, and the number of spin-up and down electrons may differ by at most 1, prohibiting modeling ionic systems.

5. Experiments

Here, we analyze Globe and Moon across a variety of different experimental settings. Firstly, we investigate the behavior of Globe when training on similar geometries where one would expect significant information overlap between molecular orbitals. Next, we take a look at its extrapolation behavior on such similar structures. Thirdly, we train Globe on molecules that share no common structure. Fourthly, we investigate the transferability of a trained Globe to similar and larger molecules. Lastly, we compare Moon with recent neural wave functions on the larger benzene molecule.

As the true ground-state energies for any molecular system are rarely known, we either compare them to highly accurate reference calculations or report variational energies or their standard deviation. As discussed in Section 2, VMC energies are upper bounds to the true energy and, thus, lower is better. Further, as the wave function approaches the ground state, the standard deviation of the local energy approaches zero providing a proxy for the convergence to the ground state. Appendix H details the setup and Appendix I lists the geometries we used in the following. Timings and model sizes can be found in Appendix J and Appendix K. Appendix L provides a model size ablation.

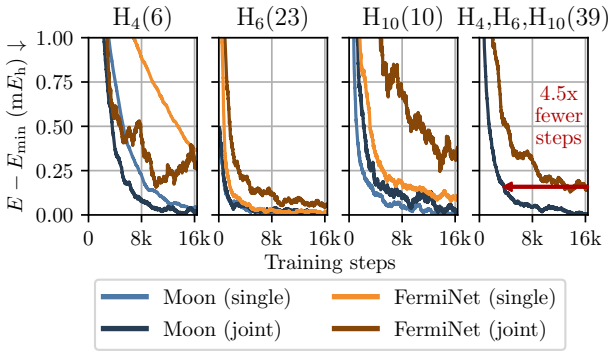


Figure 3. Convergence plots of Globe with Moon and FermiNet. Numbers in brackets show the number of geometries per molecule. In joint training, Moon converges 4.5 times faster and to lower energies.

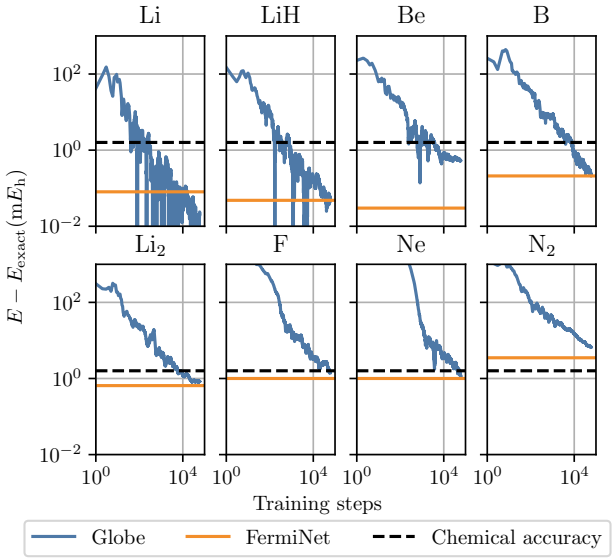


Figure 4. Globe trained on different structures jointly. FermiNet is optimized per molecule, numbers are taken from Pfau et al. (2020). Globe reaches chemical accuracy everywhere except N_2 where the specially optimized FermiNet also fails.

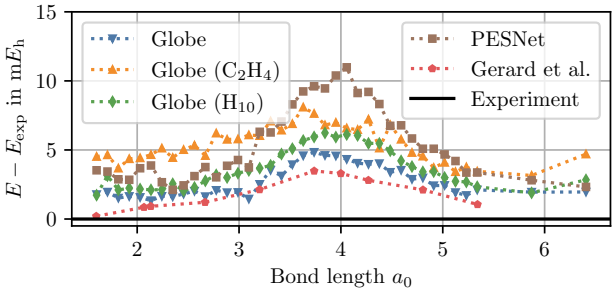


Figure 5. Relative error to Le Roy et al. (2006) on the N_2 potential energy surface. For Globe, the brackets show the additional molecule that the network has trained on. The hydrogen chain H_{10} and ethene result in $0.96\,\mathrm{m}E_\mathrm{h}$ and $2.71\,\mathrm{m}E_\mathrm{h}$ higher energies.

Table 1. Comparison between Moon and FermiNet in solving separated hydrogen chains. H_{10} indicates the energy of a single hydrogen chain, H_{10} + H_{10} for two hydrogen chains $100\,a_0$ apart, and Δ indicates the difference in energy between twice the energy of single chain compared to solving both systems jointly. All values are in Hartree.

	H_{10}	H_{10} + H_{10}	$2 \times H_{10}$	Δ
FermiNet	-5.6631	-10.7564	-11.3262	-0.5698
Moon	-5.6632	-11.3264	-11.3264	0.0000

Learning on similar systems. Training on similar systems jointly may be useful in several settings, e.g., in binding energy computations (Trogolo et al., 2019) or in labeling a diverse dataset (Hoja et al., 2021). Two key aspects are of interest here. Firstly, we investigate the loss in accuracy when training on different molecules and, secondly, we compare Moon to the existing FermiNet (Pfau et al., 2020) to identify potential benefits in convergence and accuracy. For FermiNet, we include all improvements from Gao & Günnemann (2023) and the Jastrow factor from von Glehn et al. (2023). As systems, we choose the hydrogen rectangle H_4, the 6-element hydrogen chain H_6, and the 10-element hydrogen chain H_{10}. Note that each of these systems is a collection of different geometries. We train each wave function model with our reparametrization network on four different settings, one for each molecule and one where we train on all of them jointly. Due to the variational principle, lower energies are better.

Figure 3 shows the average energy during training for both FermiNet and Moon in their two training settings (joint optimization, single molecule only). One can see that Moon converges strictly faster than a similar-sized FermiNet. For small systems, we observe that joint training even accelerates convergence in terms of steps. For FermiNet, we observe a significant gap between single and joint training for the larger hydrogen chain while Moon converges to the same energy in both regimes. Importantly, we observe that Moon behaves significantly more stable in joint optimization compared to FermiNet. We compare the standard deviation of the energy during training in Appendix M.

Size consistency. As discussed in Desiderata 2 in Section 4, for distant molecules the joint wave function is equal to the product of the individual wave functions. In contrast to previous neural wave functions like FermiNet, Globe and Moon adhere to this limit case. We demonstrate this by training Globe with FermiNet and Moon on the hydrogen chain H_{10} and transfer the wave function to two separated hydrogen chains $100\,a_0$ apart. As an optimal result, one expects the energy of the distant hydrogen chains to be equal to twice the energy of a single chain.

In Table 1 we list the difference between these settings. While FermiNet leads to a significant error of $570\,\mathrm{m}E_\mathrm{h}$,

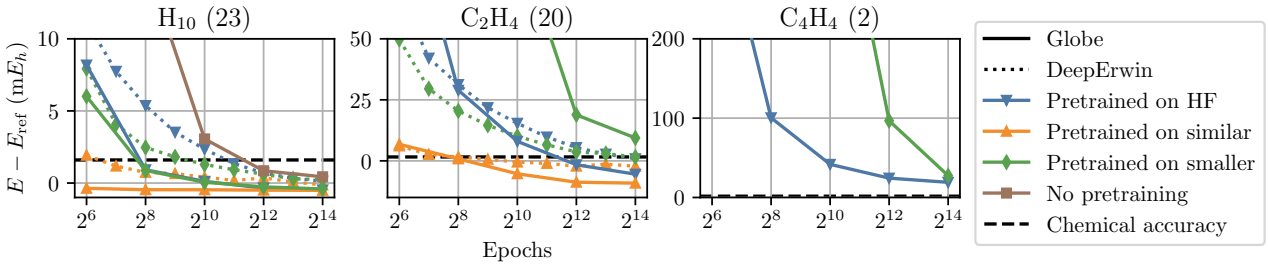


Figure 6. Average energy difference to reference data by training epochs for Globe and DeepErwin across three molecules with different pretraining techniques. Bracketed numbers show the number of geometries. For cyclobutadiene, we neither have DeepErwin reference numbers nor did we run the transferability experiment from similar structures. On ethene and cyclobutadiene, models without any pretraining are not visible due to their high error, see Appendix O.

Moon's result is in perfect agreement with the desiderata. In closer regimes, we analyze the extensivity of FermiNet and Moon in Appendix N.

Learning on dissimilar systems. While optimizing similar molecules has a small impact on the performance of Globe, we now analyze how unrelated molecules affect final energies. We test this by solving for the ground-state energies of various small systems jointly. We pick Li, LiH, Be, B, Li_2, F, Ne, and N_2 from Pfau et al. (2020) as the dataset.

In Figure 4, one sees the energy for each system during training. While, for small systems, Globe with Moon accomplishes lower energies than FermiNet despite training jointly, we observe that this does not carry over to larger systems where the gap between both increases. For all molecules, except for N_2 where FermiNet also fails to reach chemical accuracy, Globe's energies are within the chemical accuracy of the true energy. To close the gap to FermiNet, we found that increasing the size of the wave function Moon has a significant impact on the final energy. We investigate this in Appendix L and argue that this is due to the increased capacity requirement in capturing many wave functions within a single model.

Learning dissimilar energy surfaces. While the previous paragraph analyzed the absolute energies in training on different molecules, here we look at the consistency of energy surfaces when training on different energy surfaces jointly. To test this, we train three Globe on the nitrogen dimer (N_2), once solely on nitrogen, once with the hydrogen chain (H_{10}) as a smaller molecule, and once with ethane (C_2H_4) as a larger molecule. We chose the nitrogen dimer due to its challenging nature (Pfau et al., 2020). Further, we compare Globe to recent neural network-based solutions (Gerard et al., 2022; Gao & Günnemann, 2022).

The potential energy surface in Figure 5 shows that Globe comes close to the performance of current state-of-the-art neural wave functions while being able to model different molecules jointly. We suspect the gap to Gerard et al. (2022) is due to the lower number of determinants, as we use 16 instead of 32 since Pfau et al. (2020) found these to be an important hyperparameter for the nitrogen dimer. Though, one can see that adding unrelated molecules to the optimization worsens the final results depending on the system sizes. For instance, adding the relatively low-energy hydrogen chain worsens the results by $0.96\,\mathrm{m}E_\mathrm{h}$, and the larger ethene structures lead to $2.71\,\mathrm{m}E_\mathrm{h}$ worse energies on average. Note that the models trained with dissimilar structures also have seen fewer nitrogen samples during training as we evenly divide the total batch size of 4096 across all molecules.

Transferibility. Here, we analyze the transferability of Globe across different molecules. Like Scherbela et al. (2022), we train Globe on several molecules and transfer the wave function either to different geometries, or larger molecules. Specifically, we use the 10-element hydrogen chain (H_{10}), ethene (C_2H_4), and cyclobutadiene (C_4H_4). As smaller molecules, we use the 6-element hydrogen chain (H_6), methane (CH_4), and ethene, respectively. We compare our results to DeepErwin (Scherbela et al., 2022). Though, there are key distinctions in our setups. While Globe optimizes all molecules at once, DeepErwin optimizes each molecule independently with weight-sharing applied between the wave functions. Further, DeepErwin performs new CASSCF calculations for each molecule while we apply Globe without any SCF calculations to the new molecule. An epoch for DeepErwin is one optimization step per molecule with a batch size of 2048 per molecule. Since we optimize all geometries jointly, an epoch is a single step for us with a batch size of 4096 shared for all molecules in the batch. Thus, we trained Globe with approximately 10 times fewer samples and 20 times fewer steps.

The convergence plots in Figure 6 show that Globe generally converges quickly to within chemical accuracy in the hydrogen chain and ethene for the standard setting, i.e., pretraining from HF. For cyclobutadiene, we suspect the gap to Gao & Günnemann (2023) is due to our smaller wave function, i.e., we use an embedding dim of 256 and 16 determinants instead of 512 and 32, respectively. Consistent with the results from Scherbela et al. (2022), we find starting from smaller molecules generally worsens convergence for larger systems. Still, pretraining on smaller molecules leads

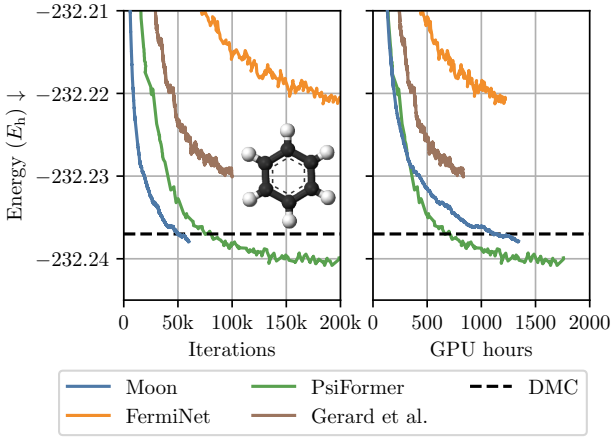


Figure 7. Energy during optimization on benzene (C_6H_6) by iterations and GPU hours. Reference data is taken from von Glehn et al. (2023); Gerard et al. (2022); Ren et al. (2022). Energies are averaged over 4000 iterations. Moon shows similar convergence behavior to the attention-based PsiFormer.

to similar results given sufficient training steps and outperforms the models without pretraining. For a view of the full convergence diagram, see Appendix O. Notably, we observe convergence without performing a single HF calculation on the larger structures, which was essential for previous methods (Pfau et al., 2020; Scherbela et al., 2022). Compared to DeepErwin, Globe results in $0.6\,\mathrm{m}E_\mathrm{h}$ and $6.3\,\mathrm{m}E_\mathrm{h}$ lower energies for the hydrogen chain and ethene, respectively.

Larger systems. As neural-network solutions are interesting thanks to their theoretical scaling, we investigate how Moon scales to larger systems compared to FermiNet (Pfau et al., 2020), recent improvements to FermiNet (Gerard et al., 2022), diffusion Monte Carlo (DMC) calculations (Ren et al., 2022), and the recently proposed attention-based PsiFormer (von Glehn et al., 2023). As system, we use the benzene molecule (C_6H_6), as recent work found a large gap between VMC and the true ground-state energy (Ren et al., 2022; von Glehn et al., 2023).

Figure 7 plots the energy of the system throughout training by iterations and GPU hours. Note that FermiNet, PsiFormer, and Gerard et al. (2022) were optimized with KFAC (Martens & Grosse, 2015) while Moon uses CG-based natural gradient descent. While CG-based natural gradient descent leads to faster early convergence, KFAC results in similar update steps later in training while being two to three times faster per iteration. In energy, we find Moon to behave similarly to PsiFormer in that it approaches lower than DMC energy levels. Compared to the best FermiNet results, we find Moon to reach similar energies in 3 times fewer GPU hours and converge in an identical time to $13.5\,\mathrm{m}E_\mathrm{h}$ lower energies.

6. Conclusion

Solving many Schrödingers jointly with a single system holds the promise of learning generalizing wave functions. Like recent deep learning-based density functional theory (DFT) functionals (Snyder et al., 2012; Kirkpatrick et al., 2021), learning a general neural network solution for quantum mechanical calculations may accelerate material discovery while increasing accuracy. In this work, we introduced Globe, the first systematic approach to performing such a generalization. By embedding orbitals as points in space and using message passing, we can learn dynamic numbers of molecular orbital functions for different molecules. Further, we presented a novel locality-driven wave function, Moon, that shows significant improvements in convergence and extensivity than previous methods when trained on diverse molecules. With Globe and Moon, we are the first to demonstrate the solving of Schrödinger equations of different molecules within a single wave function.

Acknowledgements. We greatly thank Chendi Qian for his valuable early work on designing wave functions and Michael Scherbela for swiftly sharing their data and results with us. Further, we would like to thank Tom Wollschläger and Marten Lienen for their discussion on the localization algorithm, and Jan Schuchardt and Yan Scholten for their invaluable feedback on the paper.

Funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer Normalization, July 2016.

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

Carleo, G. and Troyer, M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, February 2017. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aag2302.

Cassella, G., Sutterud, H., Azadi, S., Drummond, N. D., Pfau, D., Spencer, J. S., and Foulkes, W. M. C. Discovering Quantum Phase Transitions with Fermionic Neural Networks. *Physical Review Letters*, 130(3):036401, January 2023. doi: 10.1103/PhysRevLett.130.036401.

Ceperley, D., Chester, G. V., and Kalos, M. H. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, October 1977. doi: 10.1103/PhysRevB.16.3081.

Chakravorty, S. J., Gwaltney, S. R., Davidson, E. R., Parpia, F. A., and p Fischer, C. F. Ground-state correlation energies for atomic ions with 3 to 18 electrons. *Physical Review A*, 47(5):3649–3670, May 1993. doi: 10.1103/PhysRevA.47.3649.

Feynman, R. P. and Cohen, M. Energy Spectrum of the Excitations in Liquid Helium. *Physical Review*, 102(5): 1189–1204, June 1956. ISSN 0031-899X. doi: 10.1103/PhysRev.102.1189.

Foulkes, W. M. C., Mitas, L., Needs, R. J., and Rajagopal, G. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33–83, January 2001. doi: 10.1103/RevModPhys.73.33.

Gao, N. and Günnemann, S. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, April 2022.

Gao, N. and Günnemann, S. Sampling-free Inference for Ab-Initio Potential Energy Surface Networks. In *The Eleventh International Conference on Learning Representations*, February 2023.

Gasteiger, J., Becker, F., and Günnemann, S. GemNet: Universal Directional Graph Neural Networks for Molecules. In *Advances in Neural Information Processing Systems*, May 2021.

Gerard, L., Scherbela, M., Marquetand, P., and Grohs, P. Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need? *Advances in Neural Information Processing Systems*, May 2022.

Han, J., Zhang, L., and E, W. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, December 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108929.

Hermann, J., Schätzle, Z., and Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y.

Hermann, J., Spencer, J., Choo, K., Mezzacapo, A., Foulkes, W. M. C., Pfau, D., Carleo, G., and Noé, F. Ab-initio quantum chemistry with neural-network wavefunctions, August 2022.

Hoja, J., Medrano Sandonas, L., Ernst, B. G., Vazquez-Mayagoitia, A., DiStasio Jr., R. A., and Tkatchenko, A. QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific Data*, 8:43, February 2021. ISSN 2052-4463. doi: 10.1038/s41597-021-00812-2.

Jastrow, R. Many-body problem with strong forces. *Physical Review*, 98(5):1479, 1955.

Kirkpatrick, J., McMorrow, B., Turban, D. H. P., Gaunt, A. L., Spencer, J. S., Matthews, A. G. D. G., Obika, A., Thiry, L., Fortunato, M., Pfau, D., Castellanos, L. R., Petersen, S., Nelson, A. W. R., Kohli, P., Mori-Sánchez, P., Hassabis, D., and Cohen, A. J. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374(6573):1385–1389, December 2021. doi: 10.1126/science.abj6511.

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Le Roy, R. J., Huang, Y., and Jary, C. An accurate analytic potential function for ground-state N2 from a direct-potential-fit analysis of spectroscopic data. *The Journal of Chemical Physics*, 125(16):164310, October 2006. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2354502.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade*, pp. 9–48. Springer, 2012.

Lennard-Jones, J. E. The electronic structure of some diatomic molecules. *Transactions of the Faraday Society*, 25(0):668–686, January 1929. ISSN 0014-7672. doi: 10.1039/TF9292500668.

Li, X., Fan, C., Ren, W., and Chen, J. Fermionic neural network with effective core potential. *Physical Review Research*, 4(1):013021, January 2022a. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.4.013021.

Li, X., Li, Z., and Chen, J. Ab initio calculation of real solids via neural network ansatz, March 2022b.

Lyakh, D. I., Musiał, M., Lotrich, V. F., and Bartlett, R. J. Multireference Nature of Chemistry: The Coupled-Cluster View. *Chemical Reviews*, 112(1):182–243, January 2012. ISSN 0009-2665, 1520-6890. doi: 10.1021/cr2001417.

Martens, J. and Grosse, R. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pp. 2408–2417, 2015.

Motta, M., Ceperley, D. M., Chan, G. K.-L., Gomez, J. A., Gull, E., Guo, S., Jiménez-Hoyos, C. A., Lan, T. N., Li, J., Ma, F., Millis, A. J., Prokof'ev, N. V., Ray, U., Scuseria, G. E., Sorella, S., Stoudenmire, E. M., Sun, Q., Tupitsyn, I. S., White, S. R., Zgid, D., Zhang, S., and Simons Collaboration on the Many-Electron Problem. 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. *Physical Review X*, 7(3):031059, September 2017. ISSN 2160-3308. doi: 10.1103/PhysRevX.7.031059.

Neuscamman, E., Umrigar, C. J., and Chan, G. K.-L. Optimizing large parameter sets in variational quantum Monte Carlo. *Physical Review B*, 85(4):045103, January 2012. ISSN 1098-0121, 1550-235X. doi: 10.1103/PhysRevB.85.045103.

Nocedal, J. and Wright, S. J. *Numerical Optimization 2nd Edition*. Springer, 2006.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pp. 1310–1318. PMLR, 2013.

Pfau, D., Spencer, J. S., Matthews, A. G. D. G., and Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, September 2020. doi: 10.1103/PhysRevResearch.2.033429.

Ren, W., Fu, W., and Chen, J. Towards the ground state of molecules via diffusion Monte Carlo on neural networks, April 2022.

Scherbela, M., Reisenhofer, R., Gerard, L., Marquetand, P., and Grohs, P. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nature Computational Science*, 2(5):331–341, May 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00228-x.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, June 2018. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.5019779.

Slater, J. C. The Theory of Complex Spectra. *Physical Review*, 34(10):1293–1322, November 1929. doi: 10.1103/PhysRev.34.1293.

Snyder, J. C., Rupp, M., Hansen, K., Müller, K.-R., and Burke, K. Finding Density Functionals with Machine Learning. *Physical Review Letters*, 108(25):253002, June 2012. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.108.253002.

Spencer, J. S., Pfau, D., Botev, A., and Foulkes, W. M. C. Better, Faster Fermionic Neural Networks. *3rd NeurIPS Workshop on Machine Learning and Physical Science*, November 2020.

Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., Wouters, S., and Chan, G. K.-L. PySCF: The Python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1):e1340, 2018. ISSN 1759-0884. doi: 10.1002/wcms.1340.

Trogolo, D., Arey, J. S., and Tentscher, P. R. Gas-Phase Ozone Reactions with a Structurally Diverse Set of Molecules: Barrier Heights and Reaction Energies Evaluated by Coupled Cluster and Density Functional Theory Calculations. *The Journal of Physical Chemistry A*, 123 (2):517–536, January 2019. ISSN 1089-5639, 1520-5215. doi: 10.1021/acs.jpca.8b10323.

von Glehn, I., Spencer, J. S., and Pfau, D. A Self-Attention Ansatz for Ab-initio Quantum Chemistry. In *The Eleventh International Conference on Learning Representations*, February 2023.

Wilson, M., Gao, N., Wudarski, F., Rieffel, E., and Tubman, N. M. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo, March 2021.

Wilson, M., Moroni, S., Holzmann, M., Gao, N., Wudarski, F., Vegge, T., and Bhowmik, A. Wave function Ansatz (but Periodic) Networks and the Homogeneous Electron Gas, February 2022.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *Eighth International Conference on Learning Representations*, April 2020.

Zitnick, C. L., Das, A., Kolluru, A., Lan, J., Shuaibi, M., Sriram, A., Ulissi, Z. W., and Wood, B. M. Spherical Channels for Modeling Atomic Interactions. In *Advances in Neural Information Processing Systems*, October 2022.

A. Size consistency in quantum chemistry

If one is given two distant molecules, one can show that the Hamiltonian $\boldsymbol{H}$ from Equation 1 decomposes into two Hamiltonians $\boldsymbol{H}_1, \boldsymbol{H}_2$ for each of the respective systems.

W.l.o.g., let the electrons be sorted such that $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2] \in \mathbb{R}^N, \mathbf{r}_1 \in \mathbb{R}^{N_1}, \mathbf{r}_2 \in \mathbb{R}^{N_2}$. First, one rewrites the full Hamiltonian

$$\begin{aligned}\boldsymbol{H} = &-\frac{1}{2}\sum_{n=1}^{3N}\nabla_n^2 \\ &+\sum_{n>m=1}^{N}\frac{1}{\|r_n - r_m\|} - \sum_{n=1}^{N}\sum_{m=1}^{M}\frac{Z_m}{\|r_n - R_m\|} \\ &+\sum_{m>n=1}^{M}\frac{Z_m Z_n}{\|R_m - R_n\|},\end{aligned} \tag{26}$$

in terms of the Hamiltonians of the individual systems $\boldsymbol{H}_1, \boldsymbol{H}_2$

$$\begin{aligned}\boldsymbol{H} =&\boldsymbol{H}_1 + \boldsymbol{H}_2 + \sum_{n\in\mathbb{B}_1}\sum_{m\in\mathbb{B}_2}\frac{1}{\|\boldsymbol{r}_n - \boldsymbol{r}_m\|} \\ &- \sum_{n\in\mathbb{B}_1}\sum_{m\in\mathbb{A}_2}\frac{Z_m}{\|\boldsymbol{r}_n - \boldsymbol{R}_m\|} \\ &- \sum_{n\in\mathbb{B}_2}\sum_{m\in\mathbb{A}_1}\frac{Z_m}{\|\boldsymbol{r}_n - \boldsymbol{R}_m\|} \\ &+ \sum_{n\in\mathbb{A}_1}\sum_{m\in\mathbb{A}_2}\frac{Z_m Z_n}{\|\boldsymbol{r}_n - \boldsymbol{R}_m\|}.\end{aligned} \tag{27}$$

where $\mathbb{A}_1, \mathbb{A}_2$ and $\mathbb{B}_1, \mathbb{B}_2$ are the index sets for the nuclei and electrons for both systems, respectively. For distant systems, $\frac{1}{\|\boldsymbol{r}_n - \boldsymbol{r}_m\|} \approx \frac{1}{\|\boldsymbol{r}_n - \boldsymbol{R}_m\|} \approx \frac{1}{\|\boldsymbol{R}_n - \boldsymbol{R}_m\|}$:

$$\begin{aligned}\boldsymbol{H} =&\boldsymbol{H}_1 + \boldsymbol{H}_2 + c(\\ &\underbrace{\sum_{n\in\mathbb{B}_1}\sum_{m\in\mathbb{B}_2} 1 - \sum_{n\in\mathbb{B}_1}\sum_{m\in\mathbb{A}_2} Z_m}_{=0} \\ &\underbrace{- \sum_{n\in\mathbb{B}_2}\sum_{m\in\mathbb{A}_1} Z_m + \sum_{n\in\mathbb{A}_1}\sum_{m\in\mathbb{A}_2} Z_m Z_n)}_{=0}.\end{aligned} \tag{28}$$

where $c = \frac{1}{\|\boldsymbol{R}_n - \boldsymbol{R}_m\|}, n \in \mathbb{A}_1, m \in \mathbb{A}_2$. One may notice that for non-ionic systems $|\mathbb{B}_i| = \sum_{m\in\mathbb{A}_i} Z_m$. Thus, the first two and the last two sums cancel out and one is left with the sum of the individual Hamiltonians.

Given the decomposition of the Hamiltonian, one can show that the lowest eigenvalue of $\boldsymbol{H}$ is $E_1 + E_2$ where E_1 and E_2 are the lowest eigenvalues associated with the eigenfunctions ψ_1 and ψ_2 of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, respectively:

$$\begin{aligned}\boldsymbol{H}(\psi_1\psi_2) &= (\boldsymbol{H}_1 + \boldsymbol{H}_2)(\psi_1\psi_2) && (29)\\ &= \boldsymbol{H}_1\psi_1\psi_2 + \boldsymbol{H}_2\psi_2\psi_1 && (30)\\ &= E_1\psi_1\psi_2 + E_2\psi_2\psi_1 && (31)\\ &= (E_1 + E_2)\psi_1\psi_2. && (32)\end{aligned}$$

Thus, the ground state wave function of the combined Hamiltonian $\boldsymbol{H}$ is the product of the individual ground state wave functions ψ_1 and ψ_2.

To obtain the decomposition $\psi = \psi_1\psi_2$, it remains to show that the molecular orbital functions ϕ must only act on close-by electrons, i.e.,

$$\phi_n(\boldsymbol{r}_m|\mathbf{r}) = \begin{cases}\phi_n(\boldsymbol{r}_m|\mathbf{r}_{\mathbb{B}_1}) & \text{if } n, m \in \mathbb{B}_1, \\ \phi_n(\boldsymbol{r}_m|\mathbf{r}_{\mathbb{B}_2}) & \text{if } n, m \in \mathbb{B}_2, . \\ 0 & \text{else.}\end{cases} \tag{33}$$

where we introduced the shorthand notation $\mathbf{r}_{\mathbb{B}_i} = \{\boldsymbol{r}_n\}_{n\in\mathbb{B}_i}$. For simplicity, we assume a single Slater determinant as wave function ψ:

$$\psi(\mathbf{r}) = \det\begin{pmatrix}\phi_1(\boldsymbol{r}_1|\mathbf{r}) & \dots & \phi_N(\boldsymbol{r}_1|\mathbf{r}) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{r}_N|\mathbf{r}) & \dots & \phi_N(\boldsymbol{r}_N|\mathbf{r})\end{pmatrix}. \tag{34}$$

If we plug in the definition from Equation 33, the matrix factorizes into a block diagonal which then in turn factorizes into the product of wave functions as desired:

$$\begin{aligned}\psi(\mathbf{r}) =& \det\begin{pmatrix}\phi_1(\boldsymbol{r}_1|\mathbf{r}_{\mathbb{B}_1}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \phi_N(\boldsymbol{r}_N|\mathbf{r}_{\mathbb{B}_2})\end{pmatrix} && (35)\\ =& \det\Phi_1(\mathbf{r}_{\mathbb{B}_1})\det\Phi_2(\mathbf{r}_{\mathbb{B}_2}) && (36)\\ =& \psi_1(\mathbf{r}_{\mathbb{B}_1})\psi_2(\mathbf{r}_{\mathbb{B}_2}). && (37)\end{aligned}$$

B. Orbital localization algorithm

In an ideal setting, we would pick the orbital locations such that a) the *local* environment defines them, b) they are *deterministic* and c) they change *smoothly* with arbitrary changes to the atoms. Except for a few edge cases, which we discuss in the next paragraph, we satisfy all three criteria with Algorithm 1. As explained in Section 4.1, we distinguish between core and valence orbitals. Where core orbitals are located at their corresponding nucleus, valence orbitals are located at the center of covalent bonds. To favor the formation of diverse spatially different bonds, higher bond types (e.g., double and triple bonds) are slightly punished such that one favors similar distanced single bonds. Since the distance of an atom to itself is always 0, we replace the self-distances by a cutoff radius c_{self} after which one prefers self-bonds.

Algorithm 1 Orbital localization

Input: nuclei positions $\boldsymbol{R}_i \in \mathbb{R}^3$, charges $Z_i \in \mathbb{N}_+$
Orbital locations $Locs = [\ ]$
Orbital types $Types = [\ ]$
 # Define core orbitals
for $i = 1; i \leq M$ **do**
  Valence orbitals $V_i :=$ Valency(Z_i)
  Core orbitals $C_i := \lceil \frac{Z_i - V_i}{2} \rceil$
  for $j = 1; j \leq C_m$ **do**
    $Locs$.append(R_i)
    $Types$.append$((Z_i, j))$
  end for
end for
Distances $D_{m,n} := \begin{cases} \|R_m - R_n\| & \text{, if } n \neq m \\ c_{\text{self}} & \text{, else} \end{cases}$
 # Define valence orbitals
Bond type $T_{m,n} := 0$
for $i = 1; i \leq \lceil \sum_m^M V_m/2 \rceil$ **do**
  Scores $S_{m,n} := \frac{\mathbb{1}[V_m > 0 \wedge V_n > 0]}{D_{m,n} + T_{m,n}/2}$
  Indices $m, n := \arg\max_{m,n} \boldsymbol{S}$ (w.l.o.g. $m \leq n$)
  $V_m := V_m - 1; V_n := V_n - 1$
  $T_{m,n} := T_{n,m} := T_{m,n} + 1$
  $Locs$.append$(\frac{R_m + R_n}{2})$
  $Types$.append$(T_{m,n})$
end for
return $Locs, Types$

Edge cases. While the orbital localization fulfills our free desiderata: locality, determinism, and smoothness most of the time, there are edge cases we would like to highlight here in which we cannot guarantee smoothness. First, discrete changes occur when the nearest neighbors between atoms change. Second, if multiple pairs have identical distances, the algorithm would not be deterministic. In such cases, we rely on a series of 'tie-breakers', i.e., further criteria. In particular, we prefer edges furthest from the center of mass. If a tie remains, we start comparing the polar angle and, lastly, the azimuthal angle to break ties. While this formulation allows us to localize orbitals deterministically it also breaks the smoothness, e.g., if the order in any of the tie-breakers changes. Further, it relies on a smoothly changing equivariant coordinate frame which cannot exist (Gao & Günnemann, 2022).

Considering these discrete jumps in our orbital localization, one may ask why we decided on this particular algorithm. To answer this, one first has to consider why these discrete changes happen within Algorithm 1. The $\arg\max$ function introduces these discrete changes. While replacing the $\arg\max$ by a smooth approximation, e.g., via a softmax would resolve all discrete jumps, it would greatly deteriorate the locality. For instance, in larger systems, the softmax will always surely converge to the center of mass rather than any local bond structure. The examples given above are in extreme situations in symmetric molecules, e.g., the transition state of cyclobutadiene. In our experience, even established classical approximative methods such as the Hartree-Fock method struggle in such situations. One may see localizing orbitals as an instantiation of the greater problem of how one should break symmetries in neural wave functions such that one can model the ground state accurately while keeping sufficient inductive bias to generalize to new structures.

As a final point of discussion, one should ask whether these discontinuities are harmful in practice. As for equilibrium structures, the cases we listed will generally not happen as every atom will be closely surrounded by as many atoms as its valency with longer distances to other atoms. Considering these aspects, we believe our orbital localization algorithm to be sufficient for the current state of neural wave functions while we encourage future work to approach the problem of discontinuities.

C. Molecular Orbital Network details

As Section 4.2 focuses on novel aspects of the wave function, we want to provide some implementation and minor details here.

Rescaling. To limit the input magnitude for distanced particles, we adopt the logarithmic rescaling from von Glehn et al. (2023), i.e.,

$$\boldsymbol{g}_{ij} = \frac{\log\left(1 + g_{ij}^{(4)}\right)}{g_{ij}^{(4)}} \boldsymbol{g}_{ij}. \tag{38}$$

Normalization. Like the reparametrization network, we use learnable normalization factors within the wave functions for the four message-passing steps Equation (18), (21), (20) and (24). Specifically, we normalize the electron embeddings after the electron-electron message passing in Equation (18) by the expected number of close electrons via

$$\hat{\boldsymbol{h}}_i^{\mathrm{e}(0)} = \frac{1}{\mu(\boldsymbol{r}_i)} \boldsymbol{h}^{\mathrm{e}(0)}, \tag{39}$$

$$\mu(\boldsymbol{r}) = 1 + \sum_{m=1}^{M} \frac{Z_m}{2} \exp\left(-\frac{\|\boldsymbol{r} - \boldsymbol{R}_m\|^2}{\sigma_{\text{norm}}^2}\right) \tag{40}$$

where $\hat{\boldsymbol{h}}_i^{\mathrm{e}(0)}$ are the electron embeddings passed to further layers. Note that this formulation is similar to Equation (13) but here we multiply by half of the charge of the nucleus to account for the expected number of electrons per spin close to the nucleus. For the electron-nuclei message-passing steps in Equation (21), (20) and (24), we use $\frac{1}{\nu^{\mathbf{R}}_{\boldsymbol{r}_i}}$, $\frac{1}{\nu^{\mathbf{R}}_{\boldsymbol{R}_i}}$, and $\frac{1}{\nu^{\mathbf{R}}_{\boldsymbol{r}_i}}$, respectively.

Reparametrized filters. In the message-passing steps in Equation (21), (20) and (24), we use reparametrized version of the spatial filters from Equation (14). Concretely, these reparametrized versions take the form

$$\bar{\Gamma}_m^{(l)}(\boldsymbol{x}) = \boldsymbol{W}^{(l)} \bar{\beta}_m(\boldsymbol{x}), \tag{41}$$

$$\begin{aligned}\bar{\beta}_m(\boldsymbol{x}) =& \boldsymbol{W}^{\text{env}} \left[\exp\left(-\left(\frac{\|\boldsymbol{x}\|}{\bar{\varsigma}_{mi}} \right)^2 \right) \right]_{i=1}^{D} \\ & \circ \left(\sigma\left(\boldsymbol{x} \bar{\boldsymbol{W}}_m^{(1)} + \bar{\boldsymbol{b}}_m^{(1)} \right) \boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)} \right).\end{aligned} \tag{42}$$

Here, we replaced the envelope ranges ς_i and the first linear layer in the MLP by atom-parameterized versions. The remaining parameters are shared across all m.

Residual connections. We add residual connections between each update layer and renormalize the embeddings, i.e.,

$$\hat{\boldsymbol{h}}_m^{\text{n}(L+1)} = \frac{1}{\sqrt{2}} \left(\boldsymbol{h}_m^{\text{n}(L)} + \boldsymbol{h}_m^{\text{n}(L+1)} \right) \tag{43}$$

where $\hat{\boldsymbol{h}}_m^{\text{n}(L+1)}$ are the nuclei embeddings used in subsequent layers. For the electron embeddings, we add a skip connection after the diffusion step

$$\hat{\boldsymbol{h}}_i^{\text{e}(L)} = \frac{1}{\sqrt{2}} \left(\boldsymbol{h}_i^{\text{e}(L)} + \boldsymbol{h}_i^{\text{e}(0)} \right). \tag{44}$$

Jastrow factor. As Jastrow factor we additively combine the Jastrow factors from Gao & Günnemann (2023) and von Glehn et al. (2023)

$$\begin{aligned}J(\mathbf{r}) =& \sum_{i=1}^{N} \text{MLP}(\boldsymbol{h}_i^{\text{e}(L)}) \\ &+ \beta_{\text{par}} \sum_{i,j;\alpha_i=\alpha_j} -\frac{1}{4} \frac{\alpha_{\text{par}}^2}{\alpha_{\text{par}} + \|\boldsymbol{r}_i - \boldsymbol{r}_j\|} \\ &+ \beta_{\text{anti}} \sum_{i,j;\alpha_i\neq\alpha_j} -\frac{1}{2} \frac{\alpha_{\text{anti}}^2}{\alpha_{\text{anti}} + \|\boldsymbol{r}_i - \boldsymbol{r}_j\|}.\end{aligned} \tag{45}$$

where $\alpha_{\text{par}}, \alpha_{\text{anti}}, \beta_{\text{par}}, \beta_{\text{anti}}$ are learnable parameters.

Parameter domain. While most of the reparametrized parameters are weight matrices without clear restrictions, some are used in numerically critical situations, e.g., as a divisor. In such cases, we apply a softplus function $f(x) = \log(1 + \exp(x))$ to avoid division by zero. Concretely, we use this domain restriction for the envelope ranges $\tilde{\bar{\varsigma}}_m$ and envelope parameters $\tilde{\bar{\sigma}}$. To obtain local orbitals, $\tilde{\bar{\pi}}$ decay to zero if the distance between an atom and an orbital increases. We accomplish this by defining $\tilde{\bar{\pi}} = \tanh(\tilde{\bar{\pi}}_1) f(\tilde{\bar{\pi}}_1)$ where $\tilde{\bar{\pi}}_1$ and $\tilde{\bar{\pi}}_2$ are two different outputs of the reparametrization network and f is the softplus function. As any direct atom-orbital parameter decays to 0 if the distance between the atom and the orbital increases, this parametrization gives the desired effect. While one could also drop any transformation on $\tilde{\bar{\pi}}$ to accomplish the decaying effect, Gao & Günnemann (2022) found a softplus on π to help in convergence which we confirmed in early experiments. Thus, we define $\tilde{\bar{\pi}}$ as a product where the $\tanh$ accounts for the decay and sign, and the softplus for the granularity. In Appendix A, we show that this parametrization results in the desired product of wave functions for distant systems.

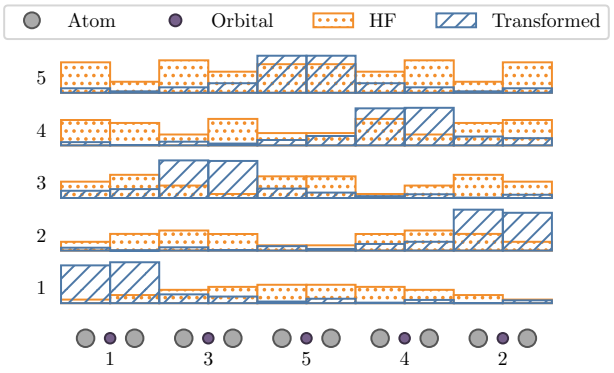


Figure 8. Illustration of our orbital canonicalization. The bottom row illustrates the hydrogen chain with our localized orbitals. Rows 1 to 5 illustrate the contribution of each atom to each of the five orbitals for the Hartree-Fock solution obtained via PySCF (Sun et al., 2018) and our transformed solution. We plot the sum of absolute values of Ω belonging to atom i as the height of the ith bar. Our transformed solution (blue) localizes the contribution close to the nuclei closest to its location.

D. Canonicalizing Hartree-Fock solutions

As discussed in Section 4.3, the solution obtained from a Hartree-Fock calculation may not align with our assumptions about the locality of orbitals. Since learning global orbital functions based on our localized orbital embeddings presents a difficult challenge, we seek to canonicalize our Hartree-Fock solutions such that they align with our localized orbitals. As explained in Section 2, in Hartree-Fock one optimizes coefficients of linear combinations of atomic orbital functions to construct molecular orbital functions. Figure 8 shows these coefficients Ω^T. Each Hartree-Fock molecular orbital, 1 to 5, exhibits a non-local structure. In the following, we will describe how we transform obtained Hartree-Fock solutions such that they obey a local structure and canonicalize their coefficients to avoid disagreements between similar molecules.

If one considers the HF electronic wave function $\psi(\mathbf{r}) = \det \Phi = \det \Phi\Omega^T$, one can see that the wave function does not change if one multiplies the coefficient matrix Ω^T with a matrix with unit determinant $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ on the right, i.e., $\psi(\mathbf{r}) = \det \Phi\Omega^T \boldsymbol{A}$. As the sign of the wave function is arbitrary, $\boldsymbol{A}$ may also have a determinant of -1. Since we know

Algorithm 2 Mask construction

Input: Atom pairs (n_i, m_i), Orbital types $T_i \in \mathbb{N}_+$, Atomic orbitals O_m
 # $\mathbb{P}$ is a dictionary of dictionaries of lists where the first level is an atom, the second level are orbital types, and the list keeps track of the orbitals.
Priority $\mathbb{P} := \{\}$
for $i = 1; i \leq N_{\text{orb}}$ **do**
  $\mathbb{P}[n_i][T_i]$.append(i)
  if $n_i \neq m_i$ **then**
    $\mathbb{P}[m_i][T_i]$.append(i)
  end if
end for
Offsets $o_i := \sum_{m=1}^{i-1} O_m$
Result $\boldsymbol{M} := \mathbf{0}^{\sum_{m=1}^{M} \times N}$
 # Iterate through all atoms.
for $n, \mathbb{T}_n \in \mathbb{P}$ **do**
  # Offset that indicates the lowest free orbital.
  Offset $o := o_n$
  # Iterate through all orbitals types of atom n.
  for $T, \boldsymbol{m} \in \mathbb{T}_n$ **do**
    # Iterate through the cardinality orbital type T.
    for $i = 1; i \leq \dim(\boldsymbol{m})$ **do**
      $\boldsymbol{M}_{o+i,m_i} := 1$
    end for
    # Since $\dim(\boldsymbol{m})$ many orbitals have been assigned, we must increase the offset.
    $o := o + \dim(\boldsymbol{m})$
  end for
end for
return $\boldsymbol{M}$

the atoms involved in the localization of the ith molecular orbital, we can formulate an optimization problem as

$$\min_{\boldsymbol{A}} \|\Omega^T \boldsymbol{A} \circ (1 - \boldsymbol{M})\|_2^2 + \sum_{i=1}^{N} (1 - \|(\Omega^T \boldsymbol{A} \circ \boldsymbol{M})_i\|_2)^2 \tag{46}$$

$$\text{s.t.} |\det \boldsymbol{A}| = 1 \tag{47}$$

where $\boldsymbol{M} \in \{0,1\}^{N \times \eta}, \eta = \sum_{m=1}^{M} O_m$ is a binary mask indicating our desired relation between atomic and molecular orbitals. Note, that the first term in Equation (46) encourages zero interaction with non-involved atoms, and the second term aids in keeping the wave function normalized.

In our matching mask $\boldsymbol{M}$, we want to preserve the order of energy levels. Atomic orbitals are typically sorted by energy level, i.e., the ith orbital has lower energies than the jth atomic orbital iff $i < j$. To get a canonical ordering, we translate these energy levels to our localized orbitals. Specifically, we want our localized core orbitals to match the atomic orbitals in the same order, i.e., our ith core orbital of an atom should match the ith atomic orbital. For valence orbitals, we enforce the same for bonds of higher order (double bonds, triple bonds, ...) where we match the first valence orbital of that bond to the lowest free orbitals of both atoms and the second bond to the second lowest free orbitals of both, etc. If an atom has bonds to k different atoms, we cannot easily define an order between bonds and, thus, distribute the k free orbitals of both atoms equally to the k different bonds.

We illustrate this in an example of H_2O. Hydrogen has 1 atomic orbital and oxygen 5 (the number of atomic orbitals is determined by the period of the element), i.e., $\eta = 6$. The number of molecular orbitals is $\lceil \frac{\sum_{m=1}^{M} Z_m}{2} \rceil = 5$. Now, assuming an equilibrium structure, our molecular orbitals are distributed as follows: three core orbitals associated with oxygen and one valence orbital for each of the O-H bonds. Given our mask construction, we get the following mask

$$\boldsymbol{M}_{\text{H2O}} = \left[\begin{array}{ccccc|c|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right] \tag{48}$$

where the first three rows are core-orbitals and the last two rows are valence-orbitals. The vertical lines group the atomic orbitals by the associated atom, the first five columns belong to the oxygen atom while the last two belong to each of the hydrogen atoms. A formal definition of our mask construction is given in Algorithm 2.

Finally, we solve Equation (46) with an alternating optimization algorithm. In the first step, we optimize $\boldsymbol{A}$ with the Broyden-Fletcher-Goldfarb-Shannon (BFGS) algorithm (Nocedal & Wright, 2006) where we parametrize $\hat{\boldsymbol{A}}$ with real numbers but normalize it with $\boldsymbol{A} = \frac{1}{\sqrt[N]{|\det \boldsymbol{A}|}} \hat{\boldsymbol{A}}$ before computing the loss. Because this restriction cannot change the sign of the determinant $\boldsymbol{A}$ and permutation represent local minima, we use the Hungarian algorithm (Kuhn, 1955) to optimize over all permutations given a fixed $\boldsymbol{A}$. We compute the cost matrix for the Hungarian algorithm by evaluating the Equation (46) for all possible pairwise permutations. After finding the optimal permutation matrix $\boldsymbol{P}^{(t)}$, we merge it into $\boldsymbol{A}^{(t+1)} = \boldsymbol{A}^{(t)} \boldsymbol{P}^{(t)}$. We either stop after a fixed number of iterations or after the loss does not change. Typically, this method converges within 2 iterations. Since we have to do this only once as preprocessing before pretraining, one can neglect the computational cost, which is in the order of a second per molecule.

As the sign of the wave function is arbitrary, we should decide on a canonical sign for each molecular orbital to avoid mismatches between different Hartree-Fock solutions of similar structures. We implement this by multiplying $\boldsymbol{M}$ with a diagonal matrix $\boldsymbol{D}$ where the diagonal elements are

defined as

$$\boldsymbol{D}_{ii} = \begin{cases} -1 & \text{if } \sum_{j=1}^{\eta} (\Omega^T A \circ \boldsymbol{M})_{ij} < 0, \\ 1 & \text{else.} \end{cases} \tag{49}$$

E. Pretraining regularization

Since we found the output of the reparametrization network to be unstable during pretraining, we add a small regularization to its output. Specifically, we define for each parameter matrix it outputs a target normal distribution, i.e., a mean and variance. For instance, for weight matrices $\bar{\boldsymbol{W}} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, such the ones in Equation (42) or Equation (19), we follow the standard initialization and define the mean to be zero and the standard deviation to be $\frac{1}{\sqrt{d_{\text{in}}}}$ (LeCun et al., 2012). In our regularization loss, we then enforce that the outputted distribution follows our target normal distribution by matching the first p_{max} moments of the output distribution with the moments of the target distribution. Concretely, for the ith outputted parameters we add the following loss

$$\mathcal{L}_{\text{pre}}(\theta_i) = \sum_{p=1}^{p_{\text{max}}} \left(\frac{1}{|\hat{\theta}_i|} \left(\sum_{j=1}^{|\hat{\theta}_i|} \hat{\theta}_{ij}^p \right) - m_p \right)^2, \tag{50}$$

$$\hat{\theta}_i = \frac{(\theta_i - \mu_i)}{s_i}, \tag{51}$$

$$m_p = \begin{cases} 0 & \text{if p is odd,} \\ (p-1)!! & \text{if p is even} \end{cases} \tag{52}$$

where θ_i is ith outputted parameter, μ_i is its target mean, s_i its target standard deviation, !! the double factorial and m_p is the pth central moment of a standard normal distribution.

F. Rescaling gradients

In the following, we discuss a gradient rescaling technique on a per-molecule basis to obtain a stable optimization if one optimizes molecules of different sizes jointly. As the norm of gradients in Equation (10) is proportional to the expected deviation from the mean, the standard deviation of the energy functions as a proxy for the gradient's norm. We rescale the gradients based on this proxy rather than the actual gradient norm as acquiring the latter is inherently expensive as it requires one to compute the full Jacobian of the network rather than a Jacobian vector product. Given the standard deviations $s_i = \sqrt{\mathbb{E}_{x \sim \psi^2_{\theta_i}} \left[E_{\theta_i}(x) - \mathbb{E}_{x \sim \psi^2_{\theta_i}} \left[E_{\theta_i}(x) \right] \right]^2}$ where θ_i indicates the parameters outputted by the reparametrization network for the ith molecule. We rescale the gradients of the ith molecule with $\min\left(1, \frac{1}{s_i}\right)$. This way small gradients are not scaled up but large gradients are scaled down.

G. VMC optimization

A VMC step consists of three substeps: 1) sampling the square of the wave function ψ_θ^2, 2) Computing the local energy $E_\theta(\mathbf{r})$ and gradients $\nabla_\theta E$ and 3) preconditioning the gradient with natural gradient descent $\boldsymbol{F}^{-1} \nabla_\theta E$. To sample the electronic wave function ψ_θ, we use Metropolis-Hastings, i.e., in multiple iterations we perturb the electron positions from the last step with gaussian noise and perform rejection sampling based on the square of the wave function ψ_θ. Next, we compute the local energies for each electron configuration as in Equation (9) and use these samples to approximate the gradients with Equation (10). Finally, we precondition the gradient with the inverse of the Fisher information matrix (FIM). As the FIM scales quadratically with the number of parameters, realizing it and computing its inverse is infeasible. Instead, we use the conjugate-gradient (CG) method to approximate its inverse (Neuscamman et al., 2012). For efficiency reasons, we compute the output of the reparametrization network once for the first two steps as it's constant throughout sampling and energy calculations. Before applying the update, we clip the norm of the gradient to 1 (Pascanu et al., 2013) such that different system sizes do not require different choices of learning rates (Gao & Günnemann, 2022).

H. Experimental setup

We implemented all experiments and methods in JAX (Bradbury et al., 2018). As we cannot rely on fixed tensor shapes like in previous work where only the spatial arrangements varied within a batch, we implemented everything with masking operations. We generally parallelize all operations where possible over all molecules within a batch. Exceptions are determinant calculations and the computation of the local energy. While one can parallelize the determinant operation if one pads smaller matrices, we found this parallelization to be slower than performing the determinant calculations sequentially. For computing the local energy, one needs to compute the Laplacian, i.e., the trace of the Hessian, of the log wave function. This is a computationally demanding task where higher memory efficiency can be achieved by serializing across molecules and electrons.

For pretraining, we use the LAMB optimizer (You et al., 2020) while for VMC we use gradient descent with a maximal gradient norm of 1. During VMC, we apply the gradient clipping from von Glehn et al. (2023), i.e., we clip all deviations from the median larger than 5 times the mean absolute deviation before computing the mean of the local energies.

All experiments ran on 1 to 4 Nvidia A100 GPUs depending on the system size. If not otherwise specified, we use the hyperparameters from Table 2.

Table 2. Default hyperparameters.

	Hyperparameter	Value
Pretraining	Steps	1e4
	Basis	STO-6G
	Method	RHF
Optimization	Steps	6e4
	Learning rate	$\frac{0.1}{1+\frac{t}{100}}$
	Batch size	4096
	Damping	1e-4 $\sigma[E_L]$
	Local energy clipping	5
	Max grad norm	1
	CG max steps	100
MCMC	Target pmove	0.5
	# Steps	40
Moon	Hidden dim	256
	E-E int dim	32
	Layers	4
	Activation	SiLU
	Determinants	16
	Jastrow layers	3
	Filter hidden dims	[16, 8]
Reparametrization	Embedding dim	128
	MLP layers	4
	Message dim	64
	Layers	3
	Activation	SiLU
	Filter hidden dims	[64, 16]

Table 3. Forward pass timings of FermiNet, Moon, and Globe.

(# nuclei / #electrons)	FermiNet	Moon	Globe
1 / 10	1.8 µs	1.6 µs	1.1 ms
1 / 40	10.1 µs	8.2 µs	1.2 ms
1 / 80	32.0 µs	25.5 µs	1.4 ms
10 / 10	2.2 µs	4.5 µs	2.3 ms
10 / 40	12.1 µs	14.7 µs	2.5 ms
10 / 80	36.7 µs	39.0 µs	2.6 ms
20 / 40	15.2 µs	21.7 µs	3.1 ms
20 / 80	44.3 µs	55.6 µs	3.1 ms

I. Molecular structures

Here, we list for each of our experiments the molecular structures and reference calculations.

For testing training on similar structures, we use the hydrogen rectangle from Pfau et al. (2020). For the six-element hydrogen chain, we use the pretraining geometries from Scherbela et al. (2022), and for the ten-element hydrogen chain, we use the geometries from Motta et al. (2017).

The extended hydrogen chains for the extensivity experiment are generated by having a n-element chain of hydrogen atoms with interatomic distances of $1.8\,\mathrm{a_0}$.

For testing dissimilar structures, we use the same distances for nitrogen as in Pfau et al. (2020). As reference energy, we use twice the atomic energy of nitrogen from Chakravorty et al. (1993) plus the experimental dissociation energy from Le Roy et al. (2006). For the additional hydrogen chain, we reuse the geometries from Motta et al. (2017). For ethene, we use the evaluation structures from Scherbela et al. (2022).

In our transferability experiment, we take the six-element and ten-element hydrogen chain as well as the methane, and ethene structures and energies from Scherbela et al. (2022). The cyclobutadiene structures are from Lyakh et al. (2012) with the final VMC energies of Gao & Günnemann (2023) as reference.

For benzene, we reuse the same geometry from Ren et al. (2022) as previous works.

J. Timings

Table 3 lists the timings for the forward pass of FermiNet, Moon, and Globe. For systems with few nuclei, we find Moon to perform faster than FermiNet while reaching higher accuracies. Though, this advantage reverses with an increasing number of nuclei due to the focus on electron-nuclei interactions.

While Globe's forward pass is significantly slower than FermiNet's or Moon's it must only be executed once per step, i.e., sampling and energy computations do not require

Table 4. Parameter counts for FermiNet, PsiFormer, Moon, and Globe.

FermiNet	PsiFormer	Moon	Globe
0.7M	1.6M	1M	13M

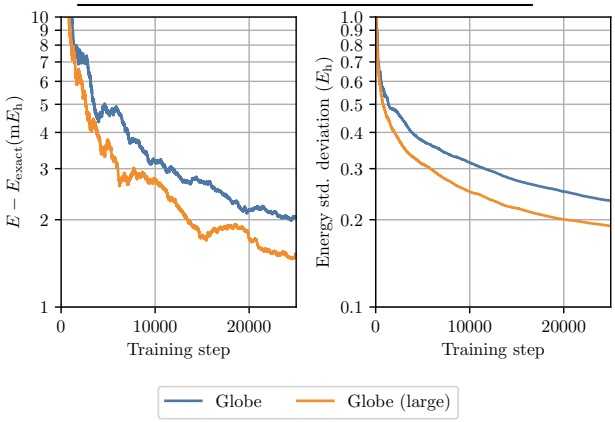


Figure 9. Abalation of Moon where the hidden dimension has been increased to 512 and the number of determinants to 32. In agreement with previous work, we find that increasing the size of the wave function improves variational results accordingly (Spencer et al., 2020; von Glehn et al., 2023).

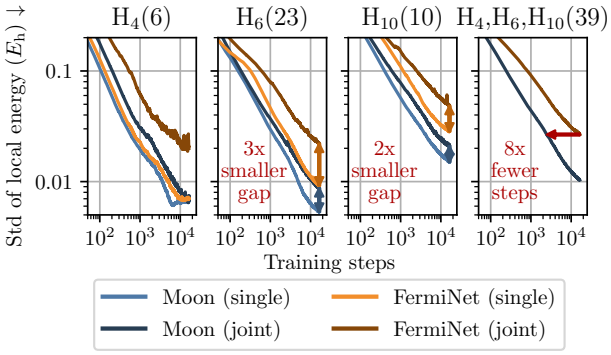


Figure 10. Convergence plots of Globe with Moon and FermiNet. Numbers in brackets show the number of geometries per molecule. In joint training, Moon converges 8 times faster and closes the gap to individual training by a factor of at least 2.

the reparametrization network but just the wave function.

K. Parameters

In Table 4, we list the number of parameters for FermiNet (Pfau et al., 2020), PsiFormer (von Glehn et al., 2023), Moon, and Globe. With its 300k more parameters, we found Moon to outperform FermiNet significantly in various benchmarks. Compared to PsiFormer, we find Moon to perform similarly with 600k fewer parameters.

Globe's large number of parameters is mostly due to the lage output space, e.g., 10M are concentrated in a dense layer to predict the 8192-dimensional output space for the 32 $\tilde{w}_i^{k\delta} \in \mathbb{R}^{256}$ orbital embeddings, see Equation 25.

L. Moon size ablation

While jointly training on diverse molecules seems to decrease the accuracy of neural wave functions, here we want to investigate the effect of the size of the network on training. We perform the common augmentation of increasing the hidden dimension to 512 and the number of determinants to 32 (Spencer et al., 2020; von Glehn et al., 2023) and compare the average energy on the diverse small molecule dataset from Section 5.

The energy during training in Figure 9 agrees with previous results on neural network wave functions that increasing the network size increases accuracy (Spencer et al., 2020; von Glehn et al., 2023). As learning diverse molecular wave functions within a single neural network may require more parameters than learning a single wave function, we already see such improvements in small molecules.

M. Standard deviation on hydrogen systems

While the energy of a system is a good indicator of convergence, any ground-state wave function will have no standard deviation in its local energy. Thus, we can take a look at the standard deviation of the local energy as a proxy for the convergence of a wave function. In Figure 10, we plot the standard deviation during the training on similar hydrogen systems. We observe that Moon converges 8 times faster in joint training while also closing the gap to the individual trainings by a factor of at least 2.

N. Extensivity

We analyze the behavior of Globe with Moon and FermiNet on increasing lengthy hydrogen chains. We first train Globe on the same hydrogen structures as in the previous experiment, i.e., H_4, H_6, and H_{10}. After training, we evaluate Globe on smaller and larger n-element hydrogen chains.

Figure 12 depicts the scaling behavior of Moon and FermiNet depending on the system size. As both are upper bounds to the true energy, lower energies are better. Considering that no finetuning is done, neither FermiNet nor Moon diverges far from the trained energies per atom. Interestingly, Moon performs better to smaller substructures like the hydrogen dimer H_2 but results in higher energies for moderately larger chains. Thanks to its locality, Moon's energy per atom is lower than FermiNet's with further increasing system sizes.

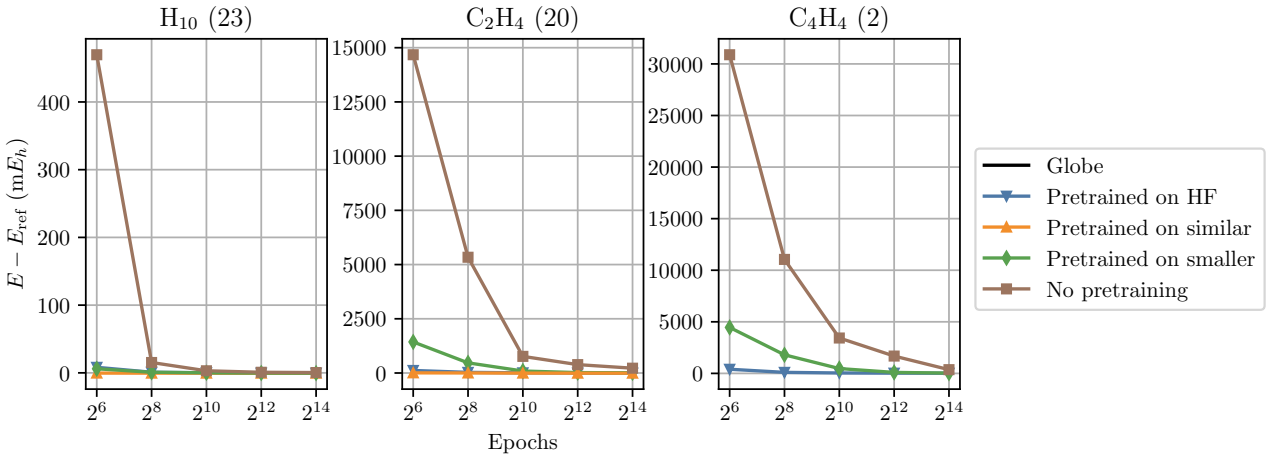


Figure 11. Extended view of Figure 6 without a limit on the y-axis. Convergence of Globe suffers

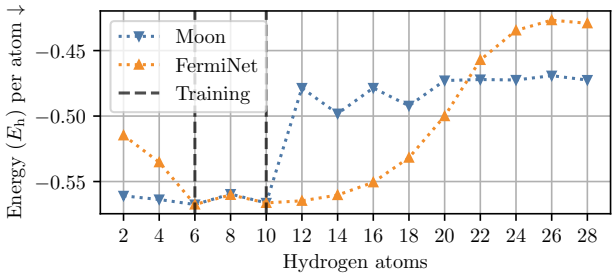


Figure 12. Evaluation of Globe with Moon and FermiNet, trained on small hydrogen clusters, on n-elment hydrogen chains. Thanks to Globe's localized orbitals, the energy per atom converges to a constant value for longer chains.

O. Extended view on transferibility

As Figure 6 in Section 5 does not include the training curves for ethene and cyclobutadiene without pretraining, we present an extended version in Figure 11. One can see that dropping pretraining impedes convergences. Meanwhile, thanks to its graph-learned approach to molecular orbitals, Globe, with pretraining on smaller molecules, is the first method to reach convergence on larger structures without performing a SCF calculation first.

C *Publication 3*: Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations

Publication: Nicholas Gao and Stephan Günnemann. Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a

Location: Vancouver, Canada

License: The rights of the publication are held by the authors; from the respective web page `https://neurips.cc/FAQ/Copyright`, June 9th 2025:

> Authors do not transfer the copyright of their papers to NeurIPS. Instead, they grant NeurIPS a non-exclusive, perpetual, royalty-free, fully-paid, fully-assignable license to copy, distribute and publicly display all or part of the paper.

Therefore, no explicit permission is required to include the publications in this thesis.

Neural Pfaffians: Solving Many Many-Electron Schrödinger Equations

Nicholas Gao, Stephan Günnemann
{n.gao,s.guennemann}@tum.de
Department of Computer Science & Munich Data Science Institute
Technical University of Munich

Abstract

Neural wave functions accomplished unprecedented accuracies in approximating the ground state of many-electron systems, though at a high computational cost. Recent works proposed amortizing the cost by learning generalized wave functions across different structures and compounds instead of solving each problem independently. Enforcing the permutation antisymmetry of electrons in such generalized neural wave functions remained challenging as existing methods require discrete orbital selection via non-learnable hand-crafted algorithms. This work tackles the problem by defining overparametrized, fully learnable neural wave functions suitable for generalization across molecules. We achieve this by relying on Pfaffians rather than Slater determinants. The Pfaffian allows us to enforce the antisymmetry on arbitrary electronic systems without any constraint on electronic spin configurations or molecular structure. Our empirical evaluation finds that a single neural Pfaffian calculates the ground state and ionization energies with chemical accuracy across various systems. On the TinyMol dataset, we outperform the 'gold-standard' CCSD(T) CBS reference energies by $1.9\,\mathrm{m}E_\mathrm{h}$ and reduce energy errors compared to previous generalized neural wave functions by up to an order of magnitude.

1 Introduction

Solving the electronic Schrödinger equation is at the heart of computational chemistry and drug discovery. Its solution provides a molecule's or material's electronic structure and energy (Zhang et al., 2023). While the exact solution is infeasible, neural networks have recently shown unprecedentedly accurate approximations (Hermann et al., 2023). These neural networks approximate the system's ground-state wave function $\Psi : \mathbb{R}^{N_\mathrm{e}\times 3} \to \mathbb{R}$, the lowest energy state, by minimizing the energy $\langle\Psi|\,\hat{H}\,|\Psi\rangle$, where $\hat{H}$ is the Hamiltonian operator, a mathematical description of the system. While such neural wave functions are highly accurate, training has proven computationally intensive.

Gao & Günnemann (2022) have shown that training a generalized neural wave function on a large class of systems amortizes the cost. However, their approach is limited to different geometric arrangements of the same molecule. Subsequent works eliminated this limitation by introducing hand-crafted algorithms (Gao & Günnemann, 2023a) or heavily relying on classical Hartree-Fock calculations (Scherbela et al., 2023). Both impose strict, non-learnable mathematical constraints and prior assumptions that may not always hold, limiting their generalization and accuracies. Hand-crafted algorithms only work for a limited set of molecules, in particular organic molecules near equilibrium, while the reliance on Hartree-Fock empirically results in degraded accuracies.

In this work, we propose the Neural Pfaffian (NeurPf) to overcome these limitations. As suggested by its name, NeurPf uses Pfaffians to define a superset of the previously used Slater determinants to enforce the fermionic antisymmetry. The Pfaffian lifts the constraint on the number of molecular orbitals from Slater determinants (Szabo & Ostlund, 2012), enabling overparametrized wave functions

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

with simpler and more accurate generalization. Compared to Globe (Gao & Günnemann, 2023a), the absence of hand-crafted algorithms enables the modeling of non-equilibrium, ionized, or excited systems. By being fully learnable without fixed Hartree-Fock calculations like TAO (Scherbela et al., 2024), NeurPf achieves significantly lower variational energies. Our empirical results show that NeurPf can learn all second-row elements' ground-state, ionization, and electron affinity potentials with a single wave function. Further, we demonstrate that NeurPf's accuracy surpasses Globe on the challenging nitrogen dimer with seven times fewer parameters while not suffering from performance degradations when adding structures to the training set. On the TinyMol dataset, NeurPf surpasses the highly accurate reference CCSD(T) CBS energies on the small structures by $1.9\,\mathrm{m}E_\mathrm{h}$ and reduces errors compared to TAO by factors of 10 and 6 on the small and large structures, respectively.

2 Quantum chemistry

Quantum chemistry aims to solve the time-independent Schrödinger equation (Foulkes et al., 2001)

$$\hat{H}\left|\Psi\right\rangle = E\left|\Psi\right\rangle \tag{1}$$

where $\Psi : \mathbb{R}^{N_\uparrow \times 3} \times \mathbb{R}^{N_\downarrow \times 3} \to \mathbb{R}$ is the electronic wave function for $N_\uparrow$ spin-up and $N_\downarrow$ spin-down electrons, $\hat{H}$ is the Hamiltonian operator, and E is the system's energy. To ease notation, if not necessary, we omit spins in Ψ and treat it as $\Psi : \mathbb{R}^{N_\mathrm{e} \times 3} \to \mathbb{R}$ where $N_\mathrm{e} = N_\uparrow + N_\downarrow$. The Hamiltonian $\hat{H}$ for molecular systems, which we are concerned with in this work, is given by

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{N_\mathrm{e}}\sum_{k=1}^{3}\frac{\partial^2}{\partial \vec{r}_{ik}^{\,2}} + \sum_{j>i}^{N_\mathrm{e}}\frac{1}{\|\vec{r}_i - \vec{r}_j\|} - \sum_{i=1}^{N_\mathrm{e}}\sum_{m=1}^{N_\mathrm{n}}\frac{Z_m}{\|\vec{r}_i - \vec{R}_m\|} + \sum_{n>m}^{N_\mathrm{n}}\frac{Z_m Z_n}{\|\vec{R}_m - \vec{R}_n\|} \tag{2}$$

with $\vec{r}_i \in \mathbb{R}^3$ being the ith electron's position, and $\vec{R}_m \in \mathbb{R}^3, Z_m \in \mathbb{N}_+$ being the mth nucleus' position and charge. The wave function Ψ describes the behavior of electrons in the system defined by the Hamiltonian $\hat{H}$. As the square of the wave function Ψ^2 is proportional to the probability density $p(\vec{\mathbf{r}}) \propto \Psi^2(\vec{\mathbf{r}})$ of finding the electrons at positions $\vec{\mathbf{r}} \in \mathbb{R}^{N_\mathrm{e} \times 3}$, its integral must be finite:

$$\int \Psi(\vec{\mathbf{r}})^2 \mathrm{d}\vec{\mathbf{r}} < \infty. \tag{3}$$

Further, as electrons are indistinguishable half-spin fermionic particles, the wave function must be antisymmetric under any same-spin electron permutation τ:

$$\Psi\left(\tau^\uparrow(\vec{\mathbf{r}}^\uparrow), \tau^\downarrow(\vec{\mathbf{r}}^\downarrow)\right) = \mathrm{sgn}(\tau^\uparrow)\mathrm{sgn}(\tau^\downarrow)\Psi(\vec{\mathbf{r}}). \tag{4}$$

To enforce this constraint, the wave function is typically defined as a so-called Slater determinant of $N_\uparrow + N_\downarrow$ integrable so-called orbital functions $\phi_i : \mathbb{R}^3 \to \mathbb{R}$:

$$\Psi_\text{Slater}(\vec{\mathbf{r}}) = \det\left[\phi_j^\uparrow(\vec{r}_i^{\,\uparrow})\right]\det\left[\phi_j^\uparrow(\vec{r}_i^{\,\downarrow})\right] = \det\Phi^\uparrow(\vec{\mathbf{r}}^\uparrow)\det\Phi^\downarrow(\vec{\mathbf{r}}^\downarrow). \tag{5}$$

Note that for the determinant to exist, one needs exactly $N_\uparrow$ up and $N_\downarrow$ down orbitals $\phi_j^\uparrow$ and $\phi_j^\downarrow$.

In linear algebra, Eq. (1) is an eigenvalue problem, where we look for the eigenfunction Ψ_0 with the lowest eigenvalue E_0. In Variational Monte Carlo (VMC), this is solved by applying the variational principle, which states that the energy of any trial wave function Ψ upper bounds E_0:

$$E_0 \leq \frac{\left\langle\Psi\right|\hat{H}\left|\Psi\right\rangle}{\left\langle\Psi^2\right\rangle} = \frac{\int \Psi(\vec{\mathbf{r}})\hat{H}\Psi(\vec{\mathbf{r}})\mathrm{d}\vec{\mathbf{r}}}{\int \Psi^2(\vec{\mathbf{r}})\mathrm{d}\vec{\mathbf{r}}}. \tag{6}$$

By plugging in the probability distribution from Eq. (3), we can rewrite Eq. (6) as

$$E_0 \leq \mathbb{E}_{p(\vec{\mathbf{r}})}\left[\Psi^{-1}(\vec{\mathbf{r}})\hat{H}\Psi(\vec{\mathbf{r}})\right] = \mathbb{E}_{p(\vec{\mathbf{r}})}\left[E_L(\vec{\mathbf{r}})\right], \tag{7}$$

with $E_L(\vec{\mathbf{r}}) = \Psi(\vec{\mathbf{r}})^{-1}\hat{H}\Psi(\vec{\mathbf{r}})$ being the so-called local energy. The right-hand side of Eq. (7) is known as the variational energy. As Eq. (7) does not require Ψ to be an analytic function, we can approximate the energy of any valid wave function Ψ with samples drawn from $p(\vec{\mathbf{r}})$. If we pick a

parametrized family of wave functions Ψ_θ, we can optimize the parameters θ to minimize the VMC energy by following the gradient of the variational energy

$$\nabla_\theta = \mathbb{E}_{p(\vec{\mathbf{r}})}\left[(E_L(\vec{\mathbf{r}}) - \mathbb{E}_{p(\vec{\mathbf{r}})}\left[E_L(\vec{\mathbf{r}})\right])\nabla_\theta \log \Psi_\theta(\vec{\mathbf{r}})\right], \tag{8}$$

where we approximate all expectations by Monte Carlo sampling (Ceperley et al., 1977).

Neural wave functions typically keep the functional form of Eq. (5) but replace the orbitals ϕ_i with learned many-electron orbitals $\phi_i^{\text{NN}} : \mathbb{R}^3 \times \mathbb{R}^{N_\text{e}\times 3} \to \mathbb{R}$ (Hermann et al., 2023). These many-electron orbitals ϕ_i^{NN} are implemented as different readouts of the same permutation-equivariant neural network. Multiplying each orbital by an envelope function $\chi_i : \mathbb{R}^3 \to \mathbb{R}$ that decays exponentially to zero at large distances enforces the finite integral requirement in Eq. (3).

Generalized wave functions solve the more general problem where the nucleus positions $\vec{\mathbf{R}}$ and charges $\mathbf{Z}$ are not fixed. Since the Hamiltonian $\hat{H}_{\vec{\mathbf{R}},\mathbf{Z}}$ depends on the molecular structure $(\vec{\mathbf{R}}, \mathbf{Z})$, so does the corresponding ground state wave function $\Psi_{\vec{\mathbf{R}},\mathbf{Z}}$. Note that we still work in the Born-Oppenheimer approximation, i.e., we treat the nuclei as classical point charges (Zhang et al., 2023). Given a dataset of molecular structures $\mathcal{D} = \{(\vec{\mathbf{R}}_1, \mathbf{Z}_1), ...\}$, the total energy $\sum_{(\vec{\mathbf{R}},\mathbf{Z})\in\mathcal{D}} \frac{\langle\Psi_{\vec{\mathbf{R}},\mathbf{Z}}|\hat{H}_{\vec{\mathbf{R}},\mathbf{Z}}|\Psi_{\vec{\mathbf{R}},\mathbf{Z}}\rangle}{\langle\Psi^2_{\vec{\mathbf{R}},\mathbf{Z}}\rangle}$ is minimized to approximate the ground state for each structure. Typically, the dependence on $\vec{\mathbf{R}}, \mathbf{Z}$ is implemented by using a meta network that takes $\vec{\mathbf{R}}, \mathbf{Z}$ as inputs and outputs the parameters of the electronic wave function (Gao & Günnemann, 2022).

3 Related work

While attempts to enforce the fermionic antisymmetry in neural wave functions in less than $O({N_\text{e}}^3)$ operations promise faster runtime than Slater determinants, the accuracy of these methods is limited (Han et al., 2019; Acevedo et al., 2020; Richter-Powell et al., 2023). Pfau et al. (2020) and Hermann et al. (2020) established Slater determinants for neural wave functions by demonstrating chemical accuracy on small molecules. Note, Eq. (5) may also be written via a block-diagonal matrix, i.e., $\Psi(\vec{\mathbf{r}}) = \det\left(\text{diag}(\Phi^\uparrow, \Phi^\downarrow)\right)$. Spencer et al. (2020)'s implementation further increased accuracies by parametrizing the off diagonals that were implicitly set to 0 before, with additional orbitals $\tilde{\Phi}$:

$$\Psi_{\text{Slater}}(\vec{\mathbf{r}}) = \det(\hat{\Phi}(\vec{\mathbf{r}})) = \det\begin{bmatrix}\Phi^\uparrow(\vec{\mathbf{r}}^\uparrow) & \tilde{\Phi}^\uparrow(\vec{\mathbf{r}}^\uparrow)\\ \tilde{\Phi}^\downarrow(\vec{\mathbf{r}}^\downarrow) & \Phi^\downarrow(\vec{\mathbf{r}}^\downarrow)\end{bmatrix}. \tag{9}$$

Several works confirmed the improved empirical accuracy of this approach (Gerard et al., 2022; Lin et al., 2021; Ren et al., 2023; Gao & Günnemann, 2023b, 2024). While later works refined the architecture to increase accuracy (von Glehn et al., 2023; Wilson et al., 2021, 2023), the use of Slater determinants mostly remained a constant, with two notable exceptions: Firstly, Lou et al. (2023) use AGP wave functions (Casula & Sorella, 2003; Casula et al., 2004) to formulate the wave function as $\Psi(\vec{\mathbf{r}}) = \det(\Phi^\uparrow)\det(\Phi^\downarrow) = \det(\Phi^\uparrow\Phi^{\downarrow T})$. This avoids picking exactly $N_\uparrow/N_\downarrow$ orbitals as $\Phi^\uparrow$ and $\Phi^\downarrow$ may be non-square but fails to generalize Eq. (9), we empirically verify the impact of this limitation in App. I. Secondly, Kim et al. (2023) introduced the combination of neural networks and Pfaffians, who demonstrated its performance on the ultra-cold Fermi gas. Though universal in theory, their parametrization yields no trivial adaption to molecular systems. In classical quantum chemistry, Bajdich et al. (2006, 2008) reported promising early results with Pfaffians in single-structure calculations for small molecules. In this work, we generalize Eq. (9) to Pfaffian wave functions that permit pretraining with Hartree-Fock calculations and generalization across molecules.

Generalized wave functions. Scherbela et al. (2022) started this research with a weight-sharing scheme between wave functions. These still had to be reoptimized for each structure. Later, Gao & Günnemann (2022, 2023b) proposed PESNet, a generalized wave function for energy surfaces allowing joint training without reoptimization. Subsequent works extended PESNet to different compounds where the main challenge is parametrizing exactly $N_\uparrow + N_\downarrow$ orbitals, such that the orbital matrix in Eq. (9) stays square. The problem of finding these orbitals was formulated into a discrete orbital selection problem. Gao & Günnemann (2023a)'s hand-crafted algorithm accomplishes this by selecting orbitals via a greedy nearest neighbor search. In contrast, Scherbela et al. (2024, 2023) use the lowest eigenvalues of the Fock matrix as selection criteria. Both introduce non-learnable constraints, limiting generalization or sacrificing accuracy. NeurPf avoids the selection problem by introducing an overparametrization when enforcing the exchange antisymmetry.

4 Neural Pfaffian

Previous generalized wave functions build on Slater wave functions and attempt to adjust the orbitals ϕ_i to the molecule. Slater determinants were chosen due to their previously demonstrated high accuracy. However, they require exactly $N_\uparrow + N_\downarrow$ orbitals. While the nuclei allow inferring the total number of electrons N_e of any stable, singlet state system, the spin distribution into $N_\uparrow$ and $N_\downarrow$ orbitals per atom is not readily available. Previous works implement this via a discrete selection of orbitals via non-learnable prior assumptions and constraints on the wave function; see Sec. 3.

Here, we present the Neural Pfaffian (NeurPf), a superset of Slater wave functions that preserves accuracy while relaxing the orbital number constraint. By not enforcing an exact number of orbitals, NeurPf is overparametrized with $N_\mathrm{o} \geq \max\{N_\uparrow, N_\downarrow\}$ orbitals, avoiding discrete selections and making it a natural choice for generalized wave functions. Importantly, NeurPf can be pretrained with Hartree-Fock, which accounts for $> 99\%$ of the total energy (Szabo & Ostlund, 2012). We introduce NeurPf in four steps: (1) We introduce the Pfaffian and use it to define a superset of Slater wave functions. (2) We present memory-efficient envelopes that additionally accelerate convergence. (3) We introduce a new pretraining scheme for matching Pfaffian and Slater wave functions. (4) We discuss combining our developments to build a generalized wave function.

4.1 Pfaffian wave function

The Pfaffian of a skew-symmetric $2n \times 2n$ matrix A, i.e., $A = -A^T$, is defined as

$$\mathrm{Pf}(A) = \frac{1}{2^n n!} \sum_{\tau \in S_{2n}} \mathrm{sgn}(\tau) \prod_{i=1}^{n} A_{\tau(2i-1),\tau(2i)} \tag{10}$$

where S_{2n} is the symmetric group of $2n$ elements. One may consider it a square root of the determinant of A since $\mathrm{Pf}(A)^2 = \det(A)$. An important property of the Pfaffian is $\mathrm{Pf}(BAB^T) = \det(B)\mathrm{Pf}(A)$ for any invertible matrix B and skew-symmetric matrix A. In the context of neural wave functions, this means that if A is an along both dimensions permutation equivariant function of the electron positions $\vec{\mathbf{r}}$, $A(\tau(\vec{\mathbf{r}})) = P_\tau A(\vec{\mathbf{r}}) P_\tau^T$, the Pfaffian of A is a valid wave function that fulfills the antisymmetry requirement from Eq. (4):

$$\Psi(\tau(\vec{\mathbf{r}})) = \mathrm{Pf}(A(\tau(\vec{\mathbf{r}}))) = \mathrm{Pf}(P_\tau A(\vec{\mathbf{r}}) P_\tau^T) = \det(P_\tau)\mathrm{Pf}(A(\vec{\mathbf{r}})) = \mathrm{sign}(\tau)\Psi(\vec{\mathbf{r}}). \tag{11}$$

To compute the Pfaffian without evaluating the $2n!$ terms in Eq. (10), we implement the Pfaffian via a tridiagonalization with the Householder transformation as in Wimmer (2012).

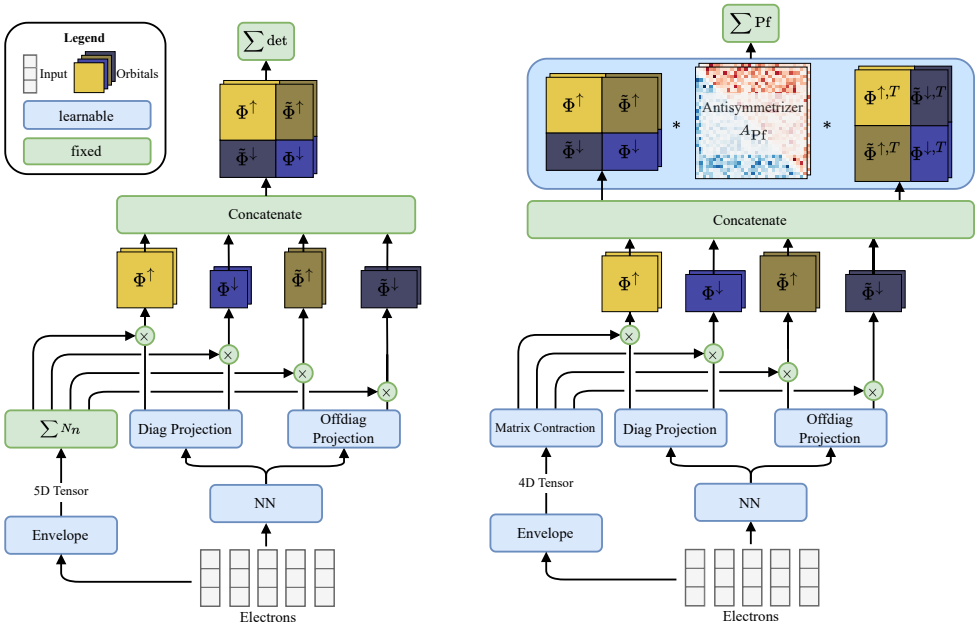
There are various ways to construct A (Bajdich et al., 2006, 2008; Kim et al., 2023). Here, we introduce a superset of Slater wave functions, enabling high accuracy on molecular systems. If A is a skew-symmetric matrix, so is BAB^T for any arbitrary matrix B. Thus, we can construct Ψ_{Pfaffian} as

$$\Psi_{\text{Pfaffian}}(\vec{\mathbf{r}}) = \frac{1}{\mathrm{Pf}(A_{\text{Pf}})} \mathrm{Pf}\left(\hat{\Phi}_{\text{Pf}}(\vec{\mathbf{r}}) A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}(\vec{\mathbf{r}})^T\right) \tag{12}$$

where $A_{\text{Pf}} \in \mathbb{R}^{N_\mathrm{o} \times N_\mathrm{o}}$ is a learnable skew-symmetric matrix and $\hat{\Phi}_{\text{Pf}} : \mathbb{R}^{N_\mathrm{e} \times 3} \to \mathbb{R}^{N_\mathrm{e} \times N_\mathrm{o}}$ is a permutation equivariant function like in Eq. (9). This construction elevates the need for having exactly $N_\uparrow / N_\downarrow$ orbitals as in Slater determinants. We may now overparametrize the wave function with $N_\mathrm{o} \geq \max\{N_\uparrow, N_\downarrow\}$ orbitals, allowing for a more flexible and simpler implementation without needing discrete orbital selection. By choosing $\hat{\Phi}_{\text{Pf}} = \hat{\Phi}$, it is straightforward to see that Eq. (12) is a superset of the Slater determinant wave function in Eq. (9). Note that, like in Eq. (9), we parametrize two sets of orbital functions Φ_{Pf} and $\tilde{\Phi}_{\text{Pf}}$ and change their order for spin-down electrons to not enforce the exchange antisymmetry between different-spin electrons. As the normalizer $\mathrm{Pf}(A_{\text{Pf}})$ is constant, we drop it going forward. As it is common in quantum chemistry (Szabo & Ostlund, 2012; Hermann et al., 2020), we use linear combinations of wave functions to increase expressiveness:

$$\Psi_{\text{Pfaffian}}(\vec{\mathbf{r}}) = \sum_{k=1}^{N_\mathrm{k}} c_k \Psi_{\text{Pfaffian},k}(\vec{\mathbf{r}}). \tag{13}$$

We visually compare the schematic of the Slater determinant and Pfaffian wave functions in Fig. 1. In App. A, we discuss how to handle odd numbers of electrons such that $\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T$ has even



(a) Slater wave function (b) Neural Pfaffian

Fig. 1: Schematic of the Slater determinant (1a) and our NeurPf (1b). Where the Slater formulation requires exactly N_e orbital functions, the Pfaffian formulation works for any number $N_\mathrm{o} \geq \max\{N_\uparrow, N_\downarrow\}$ of orbital functions, indicated by the rectangular orbital blocks.

dimensions. Like previous work (Pfau et al., 2020), we parametrize the orbital functions ϕ_i as a product of a permutation equivariant neural network $\boldsymbol{h} : \mathbb{R}^3 \times \mathbb{R}^{N\times 3} \to \mathbb{R}^{N_\mathrm{f}}$ and an envelope function $\chi : \mathbb{R}^3 \to \mathbb{R}$:

$$\phi_{ki}(\vec{r}_j|\vec{\mathbf{r}}) = \chi_{ki}(\vec{r}_j) \cdot \boldsymbol{h}(\vec{r}_j|\vec{\mathbf{r}})^T \boldsymbol{w}_{ki} \cdot \eta_{ki}^{N_\uparrow - N_\downarrow} \tag{14}$$

with $\boldsymbol{w}_{ki} \in \mathbb{R}^{N_\mathrm{f}}$ being a learnable weight vector, and $\eta_{ki}^{N_\uparrow - N_\downarrow} \in \mathbb{R}$ being a scalar depending on the spin state of the system, i.e., the difference between the number of up and down electrons. The envelope function χ ensures that the integral of the squared wave function is finite. For $\boldsymbol{h}$, we use Moon from Gao & Günnemann (2023a) thanks to its size consistency.

4.2 Memory-efficient envelopes

To satisfy the finite integral requirement on the square of Ψ in Eq. (3), the orbitals ϕ are multiplied by an envelope function $\chi : \mathbb{R}^3 \to \mathbb{R}$ that exponentially decays to zero at large distances. We do not split spins here and work with $N_\mathrm{e} = N_\uparrow + N_\downarrow$ to simplify the discussion, but, in practice, we would split the envelopes into two sets, one for Φ_Pf and one for $\tilde{\Phi}_\mathrm{Pf}$. The envelope function is typically a sum of exponentials centered on the nuclei (Spencer et al., 2020). In Einstein's summation notation, the envelope function can be written as

$$\chi_{ki}(\vec{r}_{bj}) = \underbrace{\boldsymbol{\pi}_{kmi}}_{N_\mathrm{k}\times N_\mathrm{n}\times N_\mathrm{o}} \underbrace{\cdot \exp(-\boldsymbol{\sigma}_{kmi}\|\vec{\mathbf{r}}_{bj} - \vec{\mathbf{R}}_m\|)_{bkmji}}_{N_\mathrm{b}\times N_\mathrm{k}\times N_\mathrm{n}\times N_\mathrm{e}\times N_\mathrm{o}} \tag{15}$$

where N_b denotes the batch size. Empirically, we found the tensor on the right side containing many redundant entries. Further, due to the nonlinearity of the exponential function, one cannot implement the envelope in a simple matrix contraction but has to materialize the full five-dimensional tensor. NeurPf amplifies this problem as $N_\mathrm{o} \geq N_\mathrm{e}$ whereas Slater determinants constraint $N_\mathrm{o} = N_\mathrm{e}$.

We use a single set of exponentials per nucleus instead of having one for each combination of orbital and nucleus. This reduces the number of envelopes per electron from $N_\mathrm{k} \times N_\mathrm{n} \times N_\mathrm{o}$ to $N_\mathrm{k} \times N_\mathrm{env}$, where $N_\mathrm{env} = N_\mathrm{n} \times N_\mathrm{env/nuc}$ is the number of envelope functions. In general, we pick $N_\mathrm{env/nuc}$

such that $N_{\text{env}} \approx N_{\text{o}}$. These atomic envelopes are linearly recombined into molecular envelopes, effectively enlarging $\boldsymbol{\pi}$ to a $N_{\text{k}} \times N_{\text{o}} \times N_{\text{env}}$ tensor. Thanks to these rearrangements, we avoid constructing a five-dimensional tensor. Instead, we define the envelopes as

$$\chi_{ki}(\vec{r}_{bj}) = \underbrace{\boldsymbol{\pi}_{kni}}_{N_{\text{k}} \times N_{\text{env}} \times N_{\text{o}}} \cdot \underbrace{\exp(-\boldsymbol{\sigma}_{kn} \| \vec{r}_{bj} - \vec{\mathbf{R}}_n \|)_{kbnj}}_{N_{\text{b}} \times N_{\text{k}} \times N_{\text{env}} \times N_{\text{e}}}. \tag{16}$$

Concurrently, Pfau et al. (2024) presented similar bottleneck envelopes. However, we found ours to converge faster and not yield numerical instabilities. We discuss this further in App. B and I.

4.3 Pretraining Pfaffian wave functions

Pretraining is essential in training neural wave functions and has frequently been observed to critically affect final energies (Gao & Günnemann, 2023a; von Glehn et al., 2023; Gerard et al., 2022). The pretraining aims to find orbital functions close to the ground state to stabilize the optimization. Traditionally, this is done by matching the orbitals of the neural wave function to the orbitals of a baseline wave function, typically a Hartree-Fock wave function $\Psi_{\text{HF}} = \det(\Phi_{\text{HF}})$, by solving

$$\min_{\boldsymbol{\theta}} \|\Phi_{\boldsymbol{\theta}} - \Phi_{\text{HF}}\|_2^2, \tag{17}$$

for the neural network parameters $\boldsymbol{\theta}$ (Pfau et al., 2020). Since our Pfaffian has N_{o} orbitals while Hartree-Fock has N_{e}, we cannot directly apply this to our Pfaffian wave function. Further, as we predict orbitals per nucleus, our arbitrary orbital order may not align with Hartree-Fock.

We propose two alternative pretraining schemes for neural Pfaffian wave functions: one based on matching single-electron orbitals and one based on matching geminals, effectively two-electron orbitals. We need to expand the Hartree-Fock orbitals Φ^{HF} to N_{o} orbitals to match the single-electron orbitals directly. We construct $\bar{\Phi}^{\text{HF}}$ by padding the extra $N_{\text{o}} - N_{\text{e}}$ orbitals with zeros. It can easily be verified that the wave function $\Psi_{\text{HF-Pf}} = \frac{1}{\operatorname{Pf} A_{\text{HF}}} \operatorname{Pf}(\bar{\Phi}_{\text{HF}} A_{\text{HF}} \bar{\Phi}_{\text{HF}}^T)$, is equivalent to the original Hartree-Fock wave function, i.e., $\Psi_{\text{HF-Pf}} = \Psi_{\text{HF}} = \det(\Phi_{\text{HF}})$ for any invertible skew-symmetric A_{HF}. Further, note that the multiplication of $\bar{\Phi}_{\text{HF}}$ with any matrix $T \in SO(N_{\text{o}})$ from the special orthogonal group does not change $\Psi_{\text{HF-Pf}}$. Thus, it suffices to match the single electron orbitals of $\hat{\Phi}_{\text{Pf}}$ and $\bar{\Phi}_{\text{HF}}$ up to a rotation $T \in SO(N_{\text{o}})$, yielding the following optimization problem:

$$\min_{\boldsymbol{\theta}} \min_{T \in SO(N_{\text{o}})} \|\hat{\Phi}_{\text{Pf}} - \bar{\Phi}_{\text{HF}} T\|_2^2. \tag{18}$$

We solve this alternatingly for T and $\boldsymbol{\theta}$. To match the geminals $\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T$ and $\Phi_{\text{HF}} A_{\text{HF}} \Phi_{\text{HF}}^T$, we have to account for the fact that the choice of A_{HF} is arbitrary as long as it is skew-symmetric and invertible. Again, we solve this optimization problem alternatingly by solving for $A_{\text{HF}} \in \mathbb{S} = \{A \in SO(N_{\text{e}}) : A = -A^T\}$ and $\boldsymbol{\theta}$:

$$\min_{\boldsymbol{\theta}} \min_{A_{\text{HF}} \in \mathbb{S}} \|\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T - \Phi_{\text{HF}} A_{\text{HF}} \Phi_{\text{HF}}^T\|_2^2. \tag{19}$$

While both formulations share the same minimizer, combining both yields the most stable results. We hypothesize that this is because the single-electron orbitals are more stable than the geminals and thus provide a better starting point for the optimization. In contrast, the latter provides a closer formulation of the neural network orbitals. Thus, we pretrain our neural Pfaffian wave functions by solving the optimization problem

$$\min_{\boldsymbol{\theta}} \left(\alpha \min_{T \in SO(N_{\text{o}})} \|\hat{\Phi}_{\text{Pf}} - \bar{\Phi}_{\text{HF}} T\|_2^2 + \beta \min_{A_{\text{HF}} \in \mathbb{S}} \|\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T - \Phi_{\text{HF}} A_{\text{HF}} \Phi_{\text{HF}}^T\|_2^2 \right) \tag{20}$$

with weights $\alpha, \beta \in [0, 1]$. To optimize over the special orthogonal group $SO(N_{\text{o}})$, we use the Cayley transform (Gallier, 2013). App. C further details the procedure.

4.4 Generalizing over systems

We now focus on generalizing the construction of our Pfaffian wave function for different systems. We accomplish the generalization similar to PESNet (Gao & Günnemann, 2022) by introducing a second neural network, the MetaGNN $\mathcal{M} : (\mathbb{R}^3 \times \mathbb{N}_+)^{N_{\text{n}}} \to \Theta$ that acts upon the molecular structure, i.e., nuclei positions and charges, and parametrizes the electronic wave function $\Psi_{\text{Pfaffian}} : \mathbb{R}^{N_{\text{e}} \times 3} \times \Theta \to \mathbb{R}$ for the system of interest. As architecture for the wave function and MetaGNN, we use the same architecture as in Gao et al. (2023a) with the exception being that we replace the Slater determinant with the Pfaffian as described in Sec. 4 and minor tweaks highlighted in App. D.4.

Pfaffian. To represent wave functions of different systems within a single NeurPf, we need to adapt the orbitals $\hat{\Phi}_{\text{Pf}}$ and antisymmetrizer A_{Pf} from Eq. (12) to the molecule. In doing so, we must ensure $N_{\text{o}} \geq \max\{N_\uparrow, N_\downarrow\}$. Otherwise, $\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T$ is singular, and the wave function is zero. One may solve this by picking N_{o} large enough that $N_{\text{o}} \geq \max\{N_\uparrow, N_\downarrow\}$ for all molecules in the dataset. However, this is computationally expensive, does not reuse known orbitals in the problem, and simply moves the problem to even larger systems. Instead, we grow the number of orbitals N_{o} with the system size by defining $N_{\text{orb/nuc}}$ orbitals per nucleus, as depicted in Fig. 2. This allows us to transfer orbitals from smaller systems to larger systems. We only need to ensure that $N_{\text{orb/nuc}}$ is larger than half the maximum number of electrons in a period, e.g., for the first period $N_{\text{orb/nuc}} \geq 1$, for the second period $N_{\text{orb/nuc}} \geq 5$.

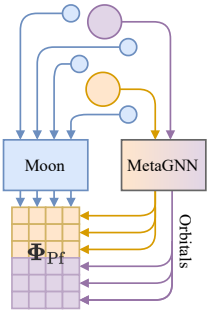


Fig. 2: Orbital parametrization per nucleus. ○, ●/● indicate electrons and nuclei, respectively.

The projection $\boldsymbol{W}$ from Eq. (14) and the envelope decays $\boldsymbol{\sigma}$ are parametrized by node embeddings, while the envelope weights $\boldsymbol{\pi}$ and the antisymmetrizer A_{Pf} are derived from edge embeddings. We predict a $N_{\text{orb/nuc}} \times N_{\text{f}}$ matrix per nucleus for $\boldsymbol{W}$ and a $N_{\text{env/nuc}}$ vector per nucleus for $\boldsymbol{\sigma}$. For the edge parameters $\boldsymbol{\pi}$ and A_{Pf}, we predict a $N_{\text{env/nuc}} \times N_{\text{orb/nuc}}$ and a $N_{\text{orb/nuc}} \times N_{\text{orb/nuc}}$ matrix per edge, respectively. These are concatenated into the $N_{\text{env}} \times N_{\text{o}}$ and $N_{\text{o}} \times N_{\text{o}}$ matrices $\boldsymbol{\pi}$ and $\hat{A}_{\text{Pf}}$. The latter is antisymmetrized to get $A_{\text{Pf}} = \frac{1}{2}(\hat{A}_{\text{Pf}} - \hat{A}_{\text{Pf}}^T)$. We parametrize the spin-dependent scalars η as node outputs for a fixed number of spin configurations N_{s}. Because the change in spin configuration does not grow with system size, N_{s} is fixed. We generate two sets of these parameters, on for Φ_{Pf} and on for $\tilde{\Phi}_{\text{Pf}}$. App. D provides definitions for the wave function, the MetaGNN, and the parametrization.

Pretraining. Previous work like Gao & Günnemann (2023a) needed to canonicalize the Hartree-Fock solutions for different systems before pretraining to ensure that the orbitals fit the neural network. Alternatively, Scherbela et al. (2023) relied on traditional quantum chemistry methods like Foster & Boys (1960)'s localization to canonicalize their orbitals in conjunction with sign equivariant neural networks. In contrast, we ensure that the transformed Hartree-Fock orbitals are similar across structures as we optimize $T \in SO(N_{\text{o}})$ and $A_{\text{HF}} \in \mathbb{S}$ for each structure separately, which simultaneously also accounts for arbitrary rotations in the orbitals produced by Hartree-Fock.

Limitations. While our Pfaffian-based generalized wave function significantly improves accuracy on organic chemistry, we leave the transfer to periodic systems for future work (Kosmala et al., 2023). Further, due to the lac of low-level hardware/software support for the Pfaffian and the increased number of orbitals $N_{\text{o}} \geq \max\{N_\uparrow, N_\downarrow\}$, our Pfaffian is slower than a comparably-sized Slater determinant. While we solve the issue of enforcing the fermionic antisymmetry, our neural wave functions are still unaware of any symmetries of the wave function itself. These are challenging to describe and largely unknown, but their integration may improve generalization performance (Schütt et al., 2018). Finally, in classical single-structure calculations, NeurPf may not improve accuracies. App. P discusses the broader impact of our work.

5 Experiments

In the following, we evaluate NeurPf on several atomic and molecular systems by comparing it to Globe (Gao & Günnemann, 2023a) and TAO (Scherbela et al., 2024). Concretely, we investigate the following: (1) Second-row elements and their ionization potentials and electron affinities. Globe cannot compute these due to its restriction to singlet state systems. (2) The challenging nitrogen potential energy surface where Globe significantly degraded performance when enlarging their training set with additional molecules. (3) The TinyMol dataset (Scherbela et al., 2024) to evaluate NeurPf's generalization capabilities across biochemical molecules. In interpreting the following results, one should mind the variational principle, i.e., lower energies are better for neural wave functions. Further, $1\,\text{kcal}\,\text{mol}^{-1} \approx 1.6\,\text{m}E_{\text{h}}$ is the typical threshold for chemical accuracy.

Like previous work, we optimize the neural wave function using the VMC framework from Sec. 2. We precondition the gradient with the Spring optimizer (Goldshlager et al., 2024). App. E details the setup further. App. F,I and J show an experiment on extensity and additional ablations.

Atomic systems and spin configurations. We evaluate NeurPf on second-row elements and their ionization potentials and electron affinities. These systems are particularly interesting as they represent

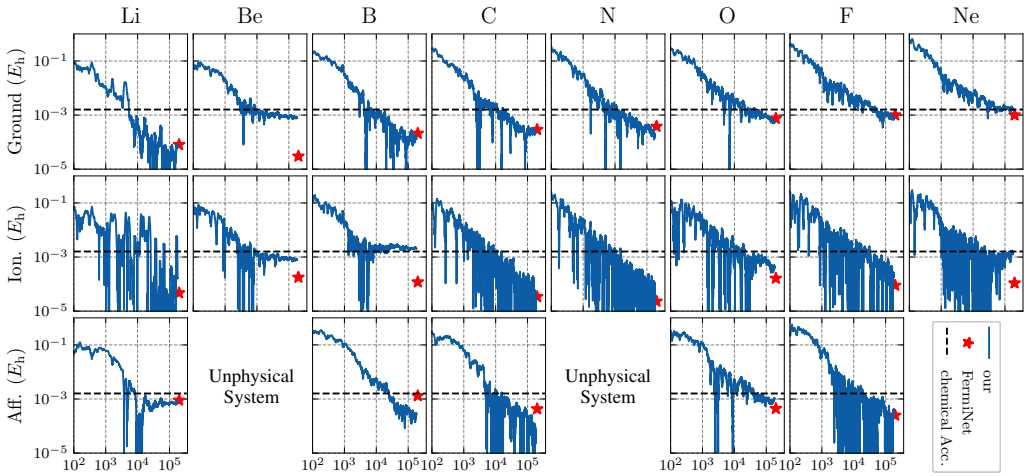


Fig. 3: Ground state, electron affinity, and ionization potential errors of second-row elements during training. A single NeurPf has been trained on all systems jointly while references (Pfau et al., 2020) were calculated separately for each system. Energies are averaged over the last 10% of steps.

a wide range of spin configurations. We cannot use Globe on such systems because they differ from the singlet state assumption. Instead, we compare our results to the single-structure calculations from Pfau et al. (2020)'s FermiNet and the exact results from Chakravorty et al. (1993); Klopper et al. (2010). In App. G, we repeat this experiment for metals.

Fig. 3 displays the ground state energy, electron affinity, and ionization potential errors of NeurPf during training compared to the reference energies from Pfau et al. (2020); Chakravorty et al. (1993); Klopper et al. (2010). It is apparent that NeurPf reaches chemical accuracy relative to the exact results while only training a single neural network for all systems. While separately optimized FermiNets may achieve lower errors, Pfau et al. (2020) trained 21 neural networks for 200k steps each compared to a single NeurPf trained for 200k steps, i.e., 21 times fewer steps and samples. Whereas Gao & Günnemann (2023a); Scherbela et al. (2023) focus on singlet state systems or stable biochemical molecules, NeurPf demonstrates that a generalized wave function need not be restricted to such simple systems and can even generalize to a wide range of electronic configurations.

Effect of uncorrelated data. Next, we evaluate NeurPf on the nitrogen potential energy surface, a traditionally challenging system due to its high electron correlation effects (Lyakh et al., 2012). This is particularly interesting as Gao & Günnemann (2023a) observed a significant accuracy degradation when reformulating their wave function to generalize over different systems. In particular, they found that training only on the nitrogen dimer leads to significantly lower errors than training with an ethene-augmented dataset, indicating an accuracy penalty in generalization. We replicate their setup and compare the performance of NeurPf trained on the nitrogen energy surface with and without additional ethene structures. Like Gao & Günnemann (2023a), the nitrogen structures are taken from Pfau et al. (2020) and the ethene structures from Scherbela et al. (2022). As additional references, we plot Gao & Günnemann (2022)'s PESNet and Fu et al. (2023)'s FermiNet results.

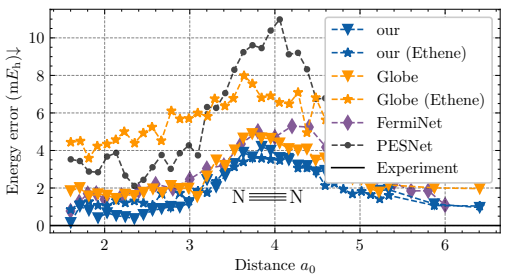


Fig. 4: Potential energy surface of nitrogen. Energies are relative to Le Roy et al. (2006).

Fig. 4 shows the error potential energy surface relative to the experimental results from Le Roy et al. (2006). NeurPf reduces the average error on the energy surface from Globe's 2.7 mE_h to 2 mE_h when training solely on nitrogen structures. When adding the ethene structures, Globe's error increases to 5.3 mE_h while NeurPf's error stays constant at 2 mE_h, a lower error than the Globe without the augmented dataset. These results indicate NeurPf's strong capabilities in approximating ground states while allowing for generalization across different systems without a significant loss in accuracy.

TinyMol dataset. Finally, we look at learning a generalized wave function over different molecules and structures. We use the TinyMol dataset (Scherbela et al., 2024), consisting of a small and large dataset. The dataset includes 'gold-standard' CCSD(T) CBS energies. The small set consists of 3 molecules with 2 heavy atoms, while the large set covers 4 molecules with 3 heavy atoms. For each molecule, 10 structures are provided. Here, we compare again both Globe (+Moon) and TAO to NeurPf. All models are directly trained on the small and large test sets.

Fig. 5 shows the mean energy difference to CCSD(T) at different stages of the training. We refer to App. K for a per molecule error attribution. It is apparent that NeurPf yields lower errors than the TAO and Globe after at least 500 steps. On the small structures, NeurPf even matches the CCSD(T) baseline after 16k steps and achieves $1.9\,\mathrm{m}E_\mathrm{h}$ lower energies after 32k steps. Since VMC methods are variational, i.e., lower energies are always better, NeurPf is more accurate than the CCSD(T) CBS reference. Compared to TAO and Globe, NeurPf reports $5.9\,\mathrm{m}E_\mathrm{h}$ and $11.3\,\mathrm{m}E_\mathrm{h}$ lower energies, respectively. On the large structures, we observe a similar pattern where we find NeurPf having a 25 times smaller error than TAO during the early stages of training and reaching $21.1\,\mathrm{m}E_\mathrm{h}$ lower energies after 32k steps – a 6 times lower error compared to the CCSD(T) baseline. Note that since the CCSD(T) (CBS) energies are neither exact nor variational, the true error to the ground state is unknown. Still, we provide additional numbers for a NeurPf trained for 128k steps in App. K. There, we find NeurPf yielding $4.4\,\mathrm{m}E_\mathrm{h}$ lower energies on the large structures. These results show that a generalized wave function can achieve high accuracy on various molecular structures without pretraining when not relying on hand-crafted algorithms or Hartree-Fock calculations. For additional experiments, we refer the reader to App. L where we first pretrain TAO and NeurPf on a separate training set and, then, finetune on the small and large test sets and App. M for a comparison of joint and separate optimization.

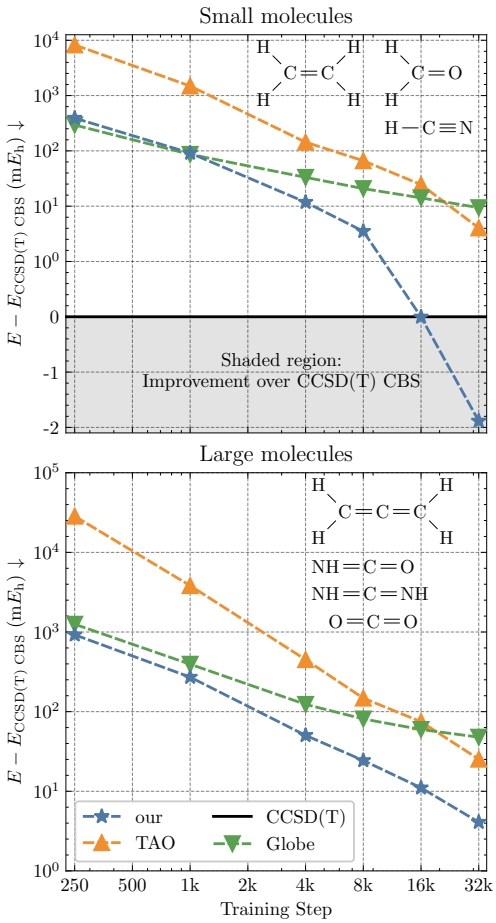


Fig. 5: Convergence of mean energy difference on the TinyMol dataset from Scherbela et al. (2024). The y-axis is linear < 1 and logarithmic ≥ 1. Due to the variational principle, NeurPf is better than the reference CCSD(T) on the small molecules.

6 Conclusion

In this work, we established a new way of parametrizing neural network wave functions for generalization across molecules via overparametrization with Pfaffians. Our Neural Pfaffian is more accurate, simpler to implement, fully learnable, and applicable to any molecular system compared to previous work. The wave function changes smoothly with the structure, avoiding the discrete orbital selection problem previously solved via hand-crafted algorithms or Hartree-Fock. Additionally, we introduced a memory-efficient implementation of the exponential envelopes, reducing memory requirements while accelerating convergence. Further, we presented a pretraining scheme for Pfaffians enabling initialization with Hartree-Fock – a crucial step for molecular systems. Our experimental evaluation demonstrated that our Neural Pfaffian can generalize across different ionizations of various systems, stay accurate when enlarging datasets, and set a new state of the art by outperforming previous neural wave functions and the reference CCSD(T) CBS on the TinyMol dataset. These developments open the door for new neural wave functions applications, e.g., to generate reference data for machine-learning force fields or density functional theory (Cheng et al., 2024; Gao et al., 2024).

Acknowledgments. We greatly thank Simon Geisler for our valuable discussions. Further, we thank Valerie Engelmayer, Leo Schwinn, and Aman Saxena for their invaluable feedback on the manuscript. Funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

References

Acevedo, A., Curry, M., Joshi, S. H., Leroux, B., and Malaya, N. Vandermonde Wave Function Ansatz for Improved Variational Monte Carlo. In *2020 IEEE/ACM Fourth Workshop on Deep Learning on Supercomputers (DLS)*, pp. 40–47, November 2020. doi: 10.1109/DLS51937.2020.00010.

Bajdich, M., Mitas, L., Drobný, G., Wagner, L. K., and Schmidt, K. E. Pfaffian Pairing Wave Functions in Electronic-Structure Quantum Monte Carlo Simulations. *Physical Review Letters*, 96 (13):130201, April 2006. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.96.130201.

Bajdich, M., Mitas, L., Wagner, L. K., and Schmidt, K. E. Pfaffian pairing and backflow wavefunctions for electronic structure quantum Monte Carlo methods. *Physical Review B*, 77(11):115112, March 2008. doi: 10.1103/PhysRevB.77.115112.

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

Casula, M. and Sorella, S. Geminal wavefunctions with Jastrow correlation: A first application to atoms. *The Journal of Chemical Physics*, 119(13):6500–6511, October 2003. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.1604379.

Casula, M., Attaccalite, C., and Sorella, S. Correlated geminal wave function for molecules: An efficient resonating valence bond approach. *The Journal of Chemical Physics*, 121(15):7110–7126, October 2004. ISSN 0021-9606. doi: 10.1063/1.1794632.

Ceperley, D., Chester, G. V., and Kalos, M. H. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, October 1977. doi: 10.1103/PhysRevB.16.3081.

Chakravorty, S. J., Gwaltney, S. R., Davidson, E. R., Parpia, F. A., and p Fischer, C. F. Ground-state correlation energies for atomic ions with 3 to 18 electrons. *Physical Review A*, 47(5):3649–3670, May 1993. doi: 10.1103/PhysRevA.47.3649.

Cheng, L., Szabó, P. B., Schätzle, Z., Kooi, D., Köhler, J., Giesbertz, K. J. H., Noé, F., Hermann, J., Gori-Giorgi, P., and Foster, A. Highly Accurate Real-space Electron Densities with Neural Networks, September 2024.

Foster, J. M. and Boys, S. F. Canonical Configurational Interaction Procedure. *Reviews of Modern Physics*, 32(2):300–302, April 1960. ISSN 0034-6861. doi: 10.1103/RevModPhys.32.300.

Foulkes, W. M. C., Mitas, L., Needs, R. J., and Rajagopal, G. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33–83, January 2001. doi: 10.1103/RevModPhys.73.33.

Fu, W., Ren, W., and Chen, J. Variance extrapolation method for neural-network variational Monte Carlo, August 2023.

Gallier, J. Remarks on the Cayley Representation of Orthogonal Matrices and on Perturbing the Diagonal of a Matrix to Make it Invertible, November 2013.

Gao, N. and Günnemann, S. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, April 2022.

Gao, N. and Günnemann, S. Generalizing Neural Wave Functions. In *International Conference on Machine Learning*, February 2023a. doi: 10.48550/arXiv.2302.04168.

Gao, N. and Günnemann, S. Sampling-free Inference for Ab-Initio Potential Energy Surface Networks. In *The Eleventh International Conference on Learning Representations*, February 2023b.

Gao, N. and Günnemann, S. On Representing Electronic Wave Functions with Sign Equivariant Neural Networks. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, March 2024.

Gao, N., Köhler, J., and Foster, A. Folx - Forward Laplacian for JAX, 2023.

Gao, N., Eberhard, E., and Günnemann, S. Learning Equivariant Non-Local Electron Density Functionals, October 2024.

Gasteiger, J., Groß, J., and Günnemann, S. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, September 2019.

Gerard, L., Scherbela, M., Marquetand, P., and Grohs, P. Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need? *Advances in Neural Information Processing Systems*, May 2022.

Goldshlager, G., Abrahamsen, N., and Lin, L. A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions, January 2024.

Han, J., Zhang, L., and E, W. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, December 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108929.

Hermann, J., Schätzle, Z., and Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y.

Hermann, J., Spencer, J., Choo, K., Mezzacapo, A., Foulkes, W. M. C., Pfau, D., Carleo, G., and Noé, F. Ab initio quantum chemistry with neural-network wavefunctions. *Nature Reviews Chemistry*, 7 (10):692–709, October 2023. ISSN 2397-3358. doi: 10.1038/s41570-023-00516-8.

Kim, J., Pescia, G., Fore, B., Nys, J., Carleo, G., Gandolfi, S., Hjorth-Jensen, M., and Lovato, A. Neural-network quantum states for ultra-cold Fermi gases, May 2023.

Klopper, W., Bachorz, R. A., Tew, D. P., and Hättig, C. Sub-meV accuracy in first-principles computations of the ionization potentials and electron affinities of the atoms H to Ne. *Physical Review A*, 81(2):022503, February 2010. ISSN 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.81.022503.

Kosmala, A., Gasteiger, J., Gao, N., and Günnemann, S. Ewald-based Long-Range Message Passing for Molecular Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 17544–17563. PMLR, July 2023.

Le Roy, R. J., Huang, Y., and Jary, C. An accurate analytic potential function for ground-state N2 from a direct-potential-fit analysis of spectroscopic data. *The Journal of Chemical Physics*, 125 (16):164310, October 2006. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2354502.

Li, R., Ye, H., Jiang, D., Wen, X., Wang, C., Li, Z., Li, X., He, D., Chen, J., Ren, W., and Wang, L. A computational framework for neural network-based variational Monte Carlo with Forward Laplacian. *Nature Machine Intelligence*, 6(2):209–219, February 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00794-x.

Lin, J., Goldshlager, G., and Lin, L. Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation. *arXiv:2112.03491 [physics]*, December 2021.

Lou, W. T., Sutterud, H., Cassella, G., Foulkes, W. M. C., Knolle, J., Pfau, D., and Spencer, J. S. Neural Wave Functions for Superfluids, July 2023.

Lyakh, D. I., Musiał, M., Lotrich, V. F., and Bartlett, R. J. Multireference Nature of Chemistry: The Coupled-Cluster View. *Chemical Reviews*, 112(1):182–243, January 2012. ISSN 0009-2665, 1520-6890. doi: 10.1021/cr2001417.

Martin, W. C. and Musgrove, A. Ground levels and ionization energies for the neutral atoms. 1998.

Mishchenko, K. and Defazio, A. Prodigy: An Expeditiously Adaptive Parameter-Free Learner, October 2023.

Motta, M., Ceperley, D. M., Chan, G. K.-L., Gomez, J. A., Gull, E., Guo, S., Jiménez-Hoyos, C. A., Lan, T. N., Li, J., Ma, F., Millis, A. J., Prokof'ev, N. V., Ray, U., Scuseria, G. E., Sorella, S., Stoudenmire, E. M., Sun, Q., Tupitsyn, I. S., White, S. R., Zgid, D., Zhang, S., and Simons Collaboration on the Many-Electron Problem. 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. *Physical Review X*, 7(3):031059, September 2017. ISSN 2160-3308. doi: 10.1103/PhysRevX.7.031059.

Pfau, D., Spencer, J. S., Matthews, A. G. D. G., and Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3): 033429, September 2020. doi: 10.1103/PhysRevResearch.2.033429.

Pfau, D., Axelrod, S., Sutterud, H., von Glehn, I., and Spencer, J. S. Accurate computation of quantum excited states with neural networks. *Science*, 385(6711):eadn0137, August 2024. doi: 10.1126/science.adn0137.

Ren, W., Fu, W., Wu, X., and Chen, J. Towards the ground state of molecules via diffusion Monte Carlo on neural networks. *Nature Communications*, 14(1):1860, April 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-37609-3.

Richter-Powell, J., Thiede, L., Asparu-Guzik, A., and Duvenaud, D. Sorting Out Quantum Monte Carlo, November 2023.

Scherbela, M., Reisenhofer, R., Gerard, L., Marquetand, P., and Grohs, P. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nature Computational Science*, 2(5):331–341, May 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00228-x.

Scherbela, M., Gerard, L., and Grohs, P. Variational Monte Carlo on a Budget — Fine-tuning pre-trained Neural Wavefunctions. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Scherbela, M., Gerard, L., and Grohs, P. Towards a transferable fermionic neural wavefunction for molecules. *Nature Communications*, 15(1):120, January 2024. ISSN 2041-1723. doi: 10.1038/s41467-023-44216-9.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24): 241722, June 2018. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.5019779.

Shazeer, N. GLU Variants Improve Transformer, February 2020.

Spencer, J. S., Pfau, D., Botev, A., and Foulkes, W. M. C. Better, Faster Fermionic Neural Networks. *3rd NeurIPS Workshop on Machine Learning and Physical Science*, November 2020.

Szabo, A. and Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, 2012.

von Glehn, I., Spencer, J. S., and Pfau, D. A Self-Attention Ansatz for Ab-initio Quantum Chemistry. In *The Eleventh International Conference on Learning Representations*, February 2023.

Wilson, M., Gao, N., Wudarski, F., Rieffel, E., and Tubman, N. M. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo, March 2021.

Wilson, M., Moroni, S., Holzmann, M., Gao, N., Wudarski, F., Vegge, T., and Bhowmik, A. Neural network ansatz for periodic wave functions and the homogeneous electron gas. *Physical Review B*, 107(23):235139, June 2023. doi: 10.1103/PhysRevB.107.235139.

Wimmer, M. Algorithm 923: Efficient Numerical Computation of the Pfaffian for Dense and Banded Skew-Symmetric Matrices. *ACM Transactions on Mathematical Software*, 38(4):30:1–30:17, August 2012. ISSN 0098-3500. doi: 10.1145/2331130.2331138.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *Eighth International Conference on Learning Representations*, April 2020.

Zhang, X., Wang, L., Helwig, J., Luo, Y., Fu, C., Xie, Y., Liu, M., Lin, Y., Xu, Z., Yan, K., Adams, K., Weiler, M., Li, X., Fu, T., Wang, Y., Yu, H., Xie, Y., Fu, X., Strasser, A., Xu, S., Liu, Y., Du, Y., Saxton, A., Ling, H., Lawrence, H., Stärk, H., Gui, S., Edwards, C., Gao, N., Ladera, A., Wu, T., Hofgard, E. F., Tehrani, A. M., Wang, R., Daigavane, A., Bohde, M., Kurtin, J., Huang, Q., Phung, T., Xu, M., Joshi, C. K., Mathis, S. V., Azizzadenesheli, K., Fang, A., Aspuru-Guzik, A., Bekkers, E., Bronstein, M., Zitnik, M., Anandkumar, A., Ermon, S., Liò, P., Yu, R., Günnemann, S., Leskovec, J., Ji, H., Sun, J., Barzilay, R., Jaakkola, T., Coley, C. W., Qian, X., Qian, X., Smidt, T., and Ji, S. Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems, November 2023.

Table 1: Number of envelope parameters for the full envelope and our memory efficient envelopes for an explanatory system.

	σ	π	Total
full	1600	1600	3200
our	640	12800	13400

A Odd numbers of electrons

To handle odd numbers of electrons, we extend the electron pair matrix $\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T$ to even dimensions. We accomplish this by augmenting $\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T$ with a learnable single-electron orbital ϕ_{odd} to

$$\widehat{\hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T} = \begin{pmatrix} \hat{\Phi}_{\text{Pf}} A_{\text{Pf}} \hat{\Phi}_{\text{Pf}}^T & \phi_{\text{odd}} \\ -\phi_{\text{odd}}^T & 0 \end{pmatrix}. \tag{21}$$

To obtain a single additional orbital for the whole molecule, we parameterize one orbital $\phi_{\text{odd},m}$ for each nucleus as in Eq. (14) and sum them up to obtain $\phi_{\text{odd}} = \sum_{m=1}^{N_\text{n}} \phi_{\text{odd},m}$.

B Difference to bottleneck envelopes

Similar to the bottleneck envelope from Pfau et al. (2024), our efficient envelopes aim at reducing memory requirements. The bottleneck envelopes are defined as

$$\chi^k_{\text{bottleneck}}(r_{bi})_j = \sum_{l=1}^{L} w^k_{jl} \sum_{m=1}^{N_\text{n}} \pi_{lm} \exp\left(-\sigma_{lm} \|\boldsymbol{r}_i - \boldsymbol{R}_m\|\right) \tag{22}$$

While both methods share the idea of reducing the number of parameters, they differ in their implementation. Whereas the bottleneck envelopes construct a full set of L many-nuclei envelopes and then linearly recombine these to the final envelopes for each of $K \times N_o$ orbitals, our efficient envelopes construct the final envelopes directly from a set of single-nuclei exponentials. Further, we use a different set of basis functions for each of the K determinants. In terms of computational complexity, the bottleneck envelopes require $O(N_\text{e} N_\text{n} L) + O(KLN_\text{e}N_\text{o})$ operations to compute the envelopes, while our efficient envelopes require $O(KN_\text{env}N_\text{e}N_\text{o})$ operations. In practice, we found our efficient envelopes to be faster and converge better on all systems we tested. An ablation study is presented in App. I. Further, we observed no numerical instabilities in our envelopes as reported by Pfau et al. (2024).

Compared to the full envelopes, we find our memory efficient ones to be slower but yielding better performance. This is likely due to the increased number of wave function parameters. The number of parameters for the full envelopes and our memory efficient envelopes is shown in Tab. 1 for an example with $N_e = N_o = 20, N_n = 5, N_d = 16, N_{\frac{\text{env}}{\text{atom}}} = 8$. The full envelopes' σ, π scale both $O(N_d N_n N_o)$ while our memory efficient envelopes' σ scales $O(N_d N_n N_{\text{env/nuc}})$ and π scales $O(N_d N_n N_{\text{env/nuc}} N_o)$. In runtime, the full envelopes require $O(N_d N_n N_e N_o)$ operations, while our memory efficient envelopes require $O(N_d N_n N_{\frac{\text{env}}{\text{atom}}} N_e N_o)$ operations. In memory complexity, the full envelopes require $O(N_d N_n N_e^2)$, while our memory efficient envelopes require $O(N_d N_n N_{\frac{\text{env}}{\text{atom}}} N_e)$.

C Pretraining

To pretrain NeurPf, we solve the optimization problem from Eq. (20). The nested optimization problems are solved iteratively, where we first solve for $T \in SO(N)$ and $A_\text{HF} \in \mathbb{S}$ and then for the parameters of the wave function θ. We describe how we parametrize the special orthogonal group $SO(N)$ and the antisymmetric special orthogonal group $\mathbb{S}$ and then how we solve the optimization problems.

To optimize over the special orthogonal group $SO(N)$, we parametrize T via some arbitrary matrix $\tilde{T} \in \mathbb{R}^{N \times N}$. Next, we obtain an antisymmetrized version of $\tilde{T}$ via

$$\hat{T} = \frac{1}{2}\left(\tilde{T} - \tilde{T}^T\right). \tag{23}$$

We now may use $\hat{T}$ with the Cayley transform to obtain a special orthogonal matrix

$$\bar{T} = \left(\hat{T} - I\right)^{-1}\left(\hat{T} + I\right) \tag{24}$$

where I is the identity matrix. $\bar{T}$ is now a special orthogonal matrix where all eigenvalues are 1. To parametrize matrices with an even number of eigenvalues -1 as well, we simply multiply $\bar{T}$ with itself:

$$T = \bar{T}\bar{T} \tag{25}$$

which gives us our final parametrization of the special orthogonal group $SO(N)$ (Gallier, 2013).

We follow Gallier (2013), to parametrize antisymmetric special orthogonal matrices $\mathbb{S}$. In particular, we parametrize some T using the procedure outlined above. To parametrize A_{HF}, it remains to antisymmetrize T while preserving the special orthogonal property. We accomplish this by defining

$$A_{\text{HF}} = T\tilde{I}T^T \tag{26}$$

where

$$\tilde{I} = \text{diag}\left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \dots\right) = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ -1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & \\ 0 & 0 & -1 & 0 & \\ \vdots & & & & \ddots \end{bmatrix} \tag{27}$$

is the antisymmetric identity matrix. Since the product of special orthogonal matrices is special orthogonal and BAB^T yielding an antisymmetric matrix for any special orthogonal matrix B, we have that $A_{\text{HF}} \in \mathbb{S}$ is an antisymmetric special orthogonal matrix.

Now that we can parametrize both groups with real matrices, we can simplify the optimization problem by performing gradient optimization for both T, A_{HF}, and θ. We solve this problem alternatively, where we first solve for T and A_{HF} by doing N_{pre} steps of gradient optimization with the prodigy optimizer (Mishchenko & Defazio, 2023) and then perform a single outer step on θ with the lamb optimizer (You et al., 2020) like previous works (Gao & Günnemann, 2022; von Glehn et al., 2023).

D Model architectures

We largly reuse the same architecture for the MetaGNN $\mathcal{M} : (\mathbb{R}^3 \times \mathbb{N}_+)^{N_{\text{n}}} \to \Theta$ and wave function $\Psi_{\text{Pfaffian}} : \mathbb{R}^{N_{\text{e}} \times 3} \times \Theta \to \mathbb{R}$ as Gao & Günnemann (2023a). We canonicalize all molecular structures using the equivariant coordinate frame from Gao & Günnemann (2022).

D.1 Wave function

Similar to Gao & Günnemann (2023a), we use bars above functions and parameters to indicate that the MetaGNN $\mathcal{M}$ parameterizes these and that they vary by structure. We define our wave function as a Jastrow-Pfaffian wave function like Kim et al. (2023):

$$\Psi(\mathbf{r}) = \exp\left(J(\mathbf{r})\right) \sum_{k=1}^{K} c_k \text{Pf}\left(\hat{\Phi}^k_{\text{Pf}}(\vec{\mathbf{r}}) A^k_{\text{Pf}} \hat{\Phi}^k_{\text{Pf}}(\vec{\mathbf{r}})^T\right). \tag{28}$$

As Jastrow factor $J : \mathbb{R}^{N \times 3} \to \mathbb{R}$ we use a linear combination of a learnable MLP of electron embeddings and the fixed electronic cusp Jastrow from von Glehn et al. (2023):

$$\begin{aligned} J(\mathbf{r}) = &\sum_{i=1}^{N} \text{MLP}(\boldsymbol{h}(\vec{r}_i|\vec{\mathbf{r}})) \\ &+ \beta_{\text{par}} \sum_{i,j;\alpha_i=\alpha_j} -\frac{1}{4} \frac{\alpha^2_{\text{par}}}{\alpha_{\text{par}} + \|\boldsymbol{r}_i - \boldsymbol{r}_j\|} \\ &+ \beta_{\text{anti}} \sum_{i,j;\alpha_i\neq\alpha_j} -\frac{1}{2} \frac{\alpha^2_{\text{anti}}}{\alpha_{\text{anti}} + \|\boldsymbol{r}_i - \boldsymbol{r}_j\|} \end{aligned} \tag{29}$$

where $\boldsymbol{h} : \mathbb{R}^3 \times \mathbb{R}^{N\times 3} \to \mathbb{R}^{N_\mathrm{f}}$ is the ith output of the permutation equivariant neural network, implemented via the Molecular orbital network (Moon) (Gao & Günnemann, 2023a), $\beta_{\text{par}}, \beta_{\text{anti}}, \alpha_{\text{par}}, \alpha_{\text{anti}} \in \mathbb{R}$ are learnable scalars, and α_i is the spin of the ith electron.

The orbitals $\hat{\Phi}_{\text{Pf}}$ are defined as in Eq. (14) with Moon performing the following steps: We start with constructing electron embeddings based on electron-electron distances and then proceed to aggregate these embeddings to the orbitals. The nuclei are updated through MLPs and finally diffused to the electrons, yielding the final electron embeddings.

The initial embedding $\boldsymbol{h}_i^{(0)}$ of the ith electron is constructed as

$$\boldsymbol{h}_i^{(0)} = \frac{1}{\mu(\boldsymbol{r}_i)} \left(\sum_{j=1}^{N} \sigma\left(\boldsymbol{g}_{ij}^{\text{e-e}} \boldsymbol{W}^{\delta_{\alpha_i}^{\alpha_j}} \right) \circ \Gamma^{\delta_{\alpha_i}^{\alpha_j}} (\|\vec{r}_i - \vec{r}_j\|) \right) \boldsymbol{W} \tag{30}$$

where $\circ$ denotes the Hadamard product and the Kronecker delta $\delta_{\alpha_i}^{\alpha_j}$ as superscript indicates different parameters depending on the identity between spin α_i and α_j. $\Gamma : \mathbb{R}^I \to \mathbb{R}^D$ is a learnable radial filter function, and σ is the activation function. $\boldsymbol{g}_{ij}^{\text{e-e}} \in \mathbb{R}^4$ are the rescaled electron-electron distances (von Glehn et al., 2023):

$$\boldsymbol{g}_{ij} = \frac{\log\left(1 + \|\vec{r}_i - \vec{r}_j\|\right)}{\|\vec{r}_i - \vec{r}_j\|} \left[\vec{r}_i - \vec{r}_j, \|\vec{r}_i - \vec{r}_j\|\right]. \tag{31}$$

μ is a normalization factor:

$$\mu(\vec{r}) = 1 + \sum_{m=1}^{M} \frac{Z_m}{2} \exp\left(-\frac{\|\vec{r} - \vec{R}_m\|^2}{\sigma_{\text{norm}}^2} \right). \tag{32}$$

We use the initial electron embeddings with nuclei embeddings and electron-nuclei distances to construct pairwise nuclei-electron embeddings representing edges in a fully connected graph:

$$\boldsymbol{h}_{im}^{\text{e-n}} = \sigma\left(\boldsymbol{h}_i^{(0)} + \bar{\boldsymbol{z}}_m + \boldsymbol{g}_{im}^{\text{e-n}} \bar{\boldsymbol{W}}_m \right). \tag{33}$$

where $\bar{\boldsymbol{z}}_m$ is the mth nucleus embedding, $\boldsymbol{g}_{im}^{\text{e-n}} \in \mathbb{R}^4$ are the rescaled electron-nuclei distances like in Eq. (31). These embeddings are then aggregated with spatial filters twice: once towards the nuclei and once towards the electrons:

$$\boldsymbol{h}_m^{\text{n}\alpha(1)} = \frac{1}{\mu(\vec{R}_m)} \sum_{i \in \mathbb{A}^\alpha} \boldsymbol{h}_{i,m}^{\text{e-n}} \circ \bar{\Gamma}_m^{\text{n}}(\vec{r}_i - \vec{R}_m), \tag{34}$$

$$\boldsymbol{m}_i^{(1)} = \frac{1}{\mu(\vec{r}_i)} \sum_{m=1}^{M} \boldsymbol{h}_{i,m}^{\text{e-n}} \circ \bar{\Gamma}_m^{\text{e}}(\vec{r}_i - \vec{R}_m), \tag{35}$$

$$\boldsymbol{h}_i^{(1)} = \sigma(\boldsymbol{m}_i^{(1)} \boldsymbol{W} + \boldsymbol{b}). \tag{36}$$

We update the nuclei embeddings with L update layers:

$$\boldsymbol{h}_m^{\text{n}\alpha(l+1)} = \boldsymbol{h}_m^{\text{n}\alpha(l)} + \sigma([\boldsymbol{h}_m^{\text{n}\alpha(l)}, \boldsymbol{h}_m^{\text{n}\hat{\alpha}(l)}] \boldsymbol{W}^{(l)} + \boldsymbol{b}^{(l)}), \tag{37}$$

where $\hat{\alpha}$ denotes the opposite spin of α, to obtain the final nuclei embeddings $\boldsymbol{h}_m^{\text{n}\alpha(L)}$. The final electron embeddings $\boldsymbol{h}_i^{\text{e}(L)}$ are constructed by combining the message from the nuclei and the previous electron embedding:

$$\boldsymbol{h}_i^{\text{e}(L)} = \sigma\left(\sigma\left(\boldsymbol{h}_i^{(1)} \boldsymbol{W} + \boldsymbol{m}_i^{(L)} + \boldsymbol{b}_1 \right) \boldsymbol{W} + \boldsymbol{b}_2 \right) + \boldsymbol{h}_i^{(1)} \tag{38}$$

where $\boldsymbol{m}_i$ is the message from the nuclei to the ith electron:

$$\boldsymbol{m}_i^{(L)} = \frac{1}{\mu(\vec{r}_i)} \sum_{m=1}^{M} \sigma\left(\left[\boldsymbol{h}_m^{\text{n}\alpha_i(L)}, \boldsymbol{h}_m^{\text{n}\hat{\alpha}_i(L)} \right] \boldsymbol{W} + \boldsymbol{b} \right) \circ \bar{\Gamma}_m^{\text{diff}}(\vec{r}_i - \vec{R}_m). \tag{39}$$

The spatial filters Γ are defined as:

$$\bar{\Gamma}_m^{(l)}(\boldsymbol{x}) = \bar{\beta}_m(\boldsymbol{x})\boldsymbol{W}^{(l)}, \tag{40}$$

$$\bar{\beta}_m(\boldsymbol{x}) = \left[\exp\left(-\left(\frac{\|\boldsymbol{x}\|}{\bar{\varsigma}_{mi}}\right)^2\right)\right]_{i=1}^{D} \boldsymbol{W}^{\text{env}} \circ \left(\sigma\left(\boldsymbol{x}\bar{\boldsymbol{W}}_m^{(1)} + \bar{\boldsymbol{b}}_m^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}\right). \tag{41}$$

Note that $\bar{\beta}$ is shared across all instances of $\bar{\Gamma}$. Γ is defined analogously to $\bar{\Gamma}$ but with fixed learnable parameters instead of MetaGNN parametrized ones.

D.2 MetaGNN

The MetaGNN $\mathcal{M} : (\mathbb{R}^3 \times \mathbb{N}_+)^{N_\text{n}} \to \Theta$ takes the nucleus position $\vec{\mathbf{R}}$ and charges $\mathbf{Z}$ as input and outputs parameters of the electronic wave function to adapt the solution to the system of interest. We follow Gao & Günnemann (2022, 2023a) and implement it as a graph neural network (GNN) where nuclei are represented as nodes and edges are constructed based on inter-particle distances. The charge of the nucleus determines the initial node embeddings:

$$\boldsymbol{k}_i^{(0)} = \boldsymbol{E}_{Z_i} \tag{42}$$

where $\boldsymbol{E}$ is an embedding matrix and Z_i is the charge of the ith nucleus. These embeddings are iteratively updated via message passing in the following way:

$$\boldsymbol{k}_i^{(l+1)} = f^{(l)}(\boldsymbol{k}_i^{(l)}, \boldsymbol{t}_i^{(l)}), \tag{43}$$

$$\boldsymbol{t}_i^{(l)} = \frac{1}{\nu_{\vec{R}_i}^{\vec{\mathbf{R}}}} \sum_{j=1}^{M} g^{(l)}(\boldsymbol{k}_i^{(l)}, \boldsymbol{k}_j^{(l)}) \circ \Gamma^{(l)}(\vec{R}_i - \vec{R}_j), \tag{44}$$

$$\nu_x^{\mathcal{N}} = 1 + \sum_{y \in \mathcal{N}} \exp\left(-\frac{\|x-y\|^2}{\sigma_\text{norm}^2}\right) \tag{45}$$

where Eq. (43) describes the update function, Eq. (44) the message construction, and Eq. (45) a learnable normalization coefficient. We implement the functions f and g via Gated Linear Units (GLU) (Shazeer, 2020). As spatial filters, we use the same as in the wave function but additionally multiply the filters with radial Bessel functions from Gasteiger et al. (2019):

$$\Gamma^{(l)}(\boldsymbol{x}) = \beta(\boldsymbol{x})\boldsymbol{W}^{(l)}, \tag{46}$$

$$\beta(\boldsymbol{x}) = \left[\sqrt{\frac{2}{c}}\frac{\sin\left(\frac{f_i x}{c}\right)}{x}\exp\left(-\left(\frac{\|\boldsymbol{x}\|}{\varsigma_i}\right)^2\right)\right]_{i=1}^{D} \boldsymbol{W}^{\text{env}} \circ \left(\sigma\left(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}\right) \tag{47}$$

where f_i are learnable frequencies, and c is a smooth cutoff for the Bessel functions.

After L layers, we take the final node embeddings, pass them through another GLU, and then use a different GLU as head for each distinct parameter tensor of the wave function we want to predict. For edge-dependent parameters, like π or A, we first construct edge embeddings by concatenating all combinations of node embeddings. We pass these through a GLU and then proceed like for node embeddings. For all outputs, we add a default charge-dependent parameter tensor such that the MetaGNN only learns a delta to an initial guess depending on the charge of the nucleus.

D.3 Orbital parametrization

Our Pfaffian wave function enables us to simply parametrize a $N_\text{o} \geq \max\{N_\uparrow, N_\downarrow\}$ orbitals rather than parametrizing exactly $N_\uparrow/N_\downarrow$. As discussed in Sec. 4.4, we accomplish this by associating a fixed number of orbitals with each nucleus. Here, we provide detailed construction for all parameters of the orbital construction. For simplicity, we do not explicitly show the dependence on the kth Pfaffian. Note that we simply extend the readout by an N_k sized dimension for each of the N_k Pfaffians from Eq. (13). Further, we predict two sets of parameters, one for Φ_Pf and one for $\tilde{\Phi}_\text{Pf}$ in

Eq. (9). To parametrize the orbitals, we predict $N_{\text{orb/nuc}}$ orbital parameters for each of the N_{n} nuclei. Concretely, the linear projection to $\boldsymbol{W}_k$ from Eq. (14) are constructed as

$$\boldsymbol{W} = \begin{bmatrix} \omega_1(\boldsymbol{k}_1) \\ \vdots \\ \omega_{N_{\text{orb/nuc}}}(\boldsymbol{k}_1) \\ \omega_1(\boldsymbol{k}_2) \\ \vdots \\ \omega_{N_{\text{orb/nuc}}}(\boldsymbol{k}_{N_{\text{n}}}) \end{bmatrix} \in \mathbb{R}^{N_{\text{o}} \times N_{\text{f}}} \tag{48}$$

where $\omega_i : \mathbb{R}^D \to \mathbb{R}^{N_{\text{f}}}$ learnable readouts of our MetaGNN. Similarly, we parametrize the envelope coefficients $\boldsymbol{\sigma}_k$ from Eq. (16):

$$\boldsymbol{\sigma} = \begin{bmatrix} \varsigma_1(\boldsymbol{k}_1) \\ \vdots \\ \varsigma_{N_{\text{env/nuc}}}(\boldsymbol{k}_1) \\ \varsigma_1(\boldsymbol{k}_2) \\ \vdots \\ \varsigma_{N_{\text{env/nuc}}}(\boldsymbol{k}_{N_{\text{n}}}) \end{bmatrix} \in \mathbb{R}_+^{N_{\text{env}}} \tag{49}$$

where $\varsigma_i : \mathbb{R}^D \to \mathbb{R}_+$ are learnable readouts of our MetaGNN. The linear orbital weights $\boldsymbol{\pi}$ connect each nuclei-centered envelope to the non-atom-centered orbitals. For this, we need to find a mapping from each of the N_{env} envelopes to each of the N_{o} orbitals. Since $N_{\text{env}} = N_{\text{env/nuc}} \times N_{\text{n}}$ and $N_{\text{o}} = N_{\text{orb/nuc}} \times N_{\text{n}}$ are predicted per nuclei, a natural connection is established via a pair-wise atom function:

$$\boldsymbol{\pi} = \begin{bmatrix} \varpi_{1,1}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \dots & \varpi_{1,N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \varpi_{1,1}(\boldsymbol{k}_2, \boldsymbol{k}_1) & \dots \\ \vdots & \ddots & \vdots & & \\ \varpi_{N_{\text{env/nuc}},1}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \dots & \varpi_{N_{\text{env/nuc}},N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \varpi_{N_{\text{env/nuc}},1}(\boldsymbol{k}_2, \boldsymbol{k}_1) & \dots \\ \varpi_{1,1}(\boldsymbol{k}_1, \boldsymbol{k}_2) & & \varpi_{1,N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_2) & \varpi_{1,1}(\boldsymbol{k}_2, \boldsymbol{k}_2) & \dots \\ \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{N_{\text{env}} \times N_{\text{o}}} \tag{50}$$

where $\varpi_{i,j} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ are learnable readouts of our MetaGNN. Similarly, we establish the orbital correlations A from Eq. (14) by connecting each of the N_{o} orbitals to each other:

$$\hat{A}_{\text{Pf}} = \begin{bmatrix} \alpha_{1,1}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \dots & \alpha_{1,N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_2) & \alpha_{1,1}(\boldsymbol{k}_2, \boldsymbol{k}_1) & \dots \\ \vdots & \ddots & \vdots & & \\ \alpha_{N_{\text{orb/nuc}},1}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \dots & \alpha_{N_{\text{orb/nuc}},N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_1) & \alpha_{N_{\text{orb/nuc}},1}(\boldsymbol{k}_2, \boldsymbol{k}_1) & \dots \\ \alpha_{1,1}(\boldsymbol{k}_1, \boldsymbol{k}_2) & & \alpha_{1,N_{\text{orb/nuc}}}(\boldsymbol{k}_1, \boldsymbol{k}_2) & \alpha_{1,1}(\boldsymbol{k}_2, \boldsymbol{k}_2) & \dots \\ \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{N_{\text{o}} \times N_{\text{o}}} \tag{51}$$

$$A_{\text{Pf}} = \frac{1}{2}(\hat{A}_{\text{Pf}} - \hat{A}_{\text{Pf}}^T) \tag{52}$$

where $\alpha_{i,j} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ are learnable readouts of our MetaGNN and Eq. (52) enforcing the antisymmetry requirements on A.

D.4 Changes to the MetaGNN

We performed several optimizations on the MetaGNN from Gao & Günnemann (2023a) that primarily reduce the number of parameters while keeping accuracy. In particular, we changed the following:

- We replace all MLPs with gated linear units (GLU) (Shazeer, 2020).
- We reduced the hidden dimension from 128 to 64.

Table 2: Hyperparameters used for the experiments.

	Hyperparameter	Value
	Structure batch size	full batch
	Total electron samples	4096
Pretraining	Epochs	10000
	Learning rate	$10^{-3} * (1 + t * 10^{-4})^{-1}$
	Optimizer	Lamb
	MCMC steps	5
	Basis	STO-6G
	Subproblem steps	50
	Subproblem optimizer	Prodigy
	Subproblem α	1.0
	Subproblem β	10^{-4}
Optimization	Steps	60000
	Learning rate	$0.02 * (1 + t * 10^{-4})^{-1}$
	Optimizer	Spring
	MCMC steps	20
	Norm constraint	10^{-3}
	Damping	0.001
	Momentum	0.99
	Energy clipping	5 times mean deviation from median
Ansatz	Hidden dim	256
	E-E int dim	32
	Layers	4
	Activation	SiLU
	Determinants/Pfaffians	16
	Jastrow layers	3
	Filter hidden dims	$[16, 8]$
Pfaffian	$N_{\mathrm{orb/nuc}}$ (H, He)	2
	$N_{\mathrm{orb/nuc}}$ (Li, Be)	6
	$N_{\mathrm{orb/nuc}}$ (B, C)	7
	$N_{\mathrm{orb/nuc}}$ (N, O)	8
	$N_{\mathrm{orb/nuc}}$ (F, Ne)	10
	$N_{\mathrm{env/nuc}}$	8
MetaGNN	Embedding dim	64
	Message dim	32
	Layers	3
	Activation	SiLU
	Filter hidden dims	$[32, 16]$

- We reduced the message dimension from 64 to 32.
- We use bessel basis functions (Gasteiger et al., 2019) on the radius for edge filters.
- We remove the hand-crafted orbital locations and the associated network.
- We added a LayerNorm before every GLU.

Together, these changes reduce the number of parameters from 13M to 1M for the MetaGNN while outperforming Gao & Günnemann (2023a) as demonstrated in Sec. 5.

E Experimental setup

Table 3: Compute time per experiment measured in Nvidia A100 GPU hours.

Experiment	Time (GPU hours)
Ionization & affinity	224
N2	116
N2 + Ethene	124
TinyMol small	78
TinyMol large	96

E.1 Hyperparameters

We list the default parameters used for the experiments in Tab. 2. Most of them were taken directly from Gao & Günnemann (2023a). We may have used different parameters for the experiments in Sec. 5 if explicitly stated so. We implement everything in JAX (Bradbury et al., 2018). To compute the laplacian $\nabla^2\Psi$, we use the forward laplacian algorithm (Li et al., 2024) implemented in the folx library (Gao et al., 2023).

E.2 Source code

We provide the source code publicly on GitHub [1].

E.3 Compute time

Tab. 3 lists the compute times required for conducting our experiments measured in Nvidia A100 GPU hours. Depending on the experiment, we use between 1 and 4 GPUs per experiment via data parallelism. We typically allocated 32GB of system memory and 16 CPU cores per experiment. In terms of the number of parameters, the Moon wave function is as large as in Gao & Günnemann (2023a) at 1M parameters, and the MetaGNN shrank from 13M parameters to just 1M parameters.

E.4 Preconditioning

The Spring optimizer (Goldshlager et al., 2024) is a natural gradient descent optimizer for electronic wave functions Ψ with the following update rule

$$\theta^t = \theta^t - \eta\delta^t \tag{53}$$

$$\delta^t = (\bar{O}^T\bar{O} + \lambda I)^{-1}(\nabla\theta^t + \lambda\mu\delta^{t-1}) \tag{54}$$

where λ is the damping factor, μ is the momentum, η is the learning rate, and $\bar{O}$ is the zero-centered Jacobian:

$$\bar{O} = O - \frac{1}{N}\sum_{i=1}^{N} O_i, \tag{55}$$

$$O = \begin{bmatrix} \frac{\partial \log\psi(x_1)}{\partial\theta} \\ \vdots \\ \frac{\partial \log\psi(x_N)}{\partial\theta} \end{bmatrix}. \tag{56}$$

Since $\bar{O} \in \mathbb{R}^{N\times P}$ where N is the batch size and P the number of parameters, the update in Eq. (54) can be efficiently computed using the Woodbury matrix identity, which after some simplifications yields

$$\delta^t = \bar{O}(\bar{O}\bar{O}^T + \lambda I)^{-1}(\epsilon + \mu\bar{O}\delta^{t-1}) + \mu\delta^{t-1}. \tag{57}$$

Our early experiment found it necessary to center the jacobian $\bar{O}$ per molecule rather than once for all. In single-structure VMC, the centering eliminates the gradient of the wave function along the direction where the amplitude of the wave function increases for all inputs. This direction does not

[1] `https://github.com/n-gao/neural-pfaffian`

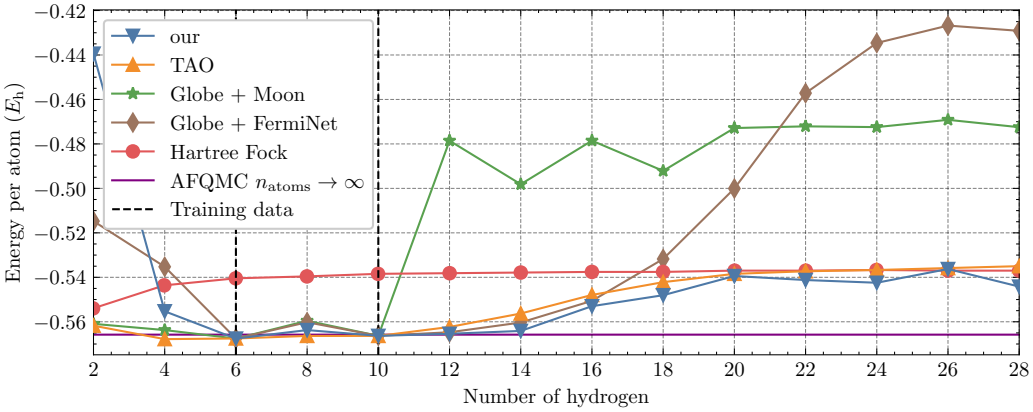


Fig. 6: Energy per atom of hydrogen chains with different lengths. The energy is computed with a single NeurPf trained on the hydrogen chains with 6 and 10 atoms.

affect energies. Thus, instead of restricting the gradient from increasing in magnitude for all samples, we constrain it to not increase in magnitude for each molecule separately N.ote that the latter implies the first but not vice versa. For multi-structure VMC, we compute $\bar{O}$ as

$$\bar{O} = O - \begin{bmatrix} \frac{1}{N_1} \sum_{i=1}^{N_1} O_i \\ \vdots \\ \frac{1}{N_M} \sum_{i=N-N_M}^{N_M} O_i \end{bmatrix} \tag{58}$$

where $N_1, ..., N_M$ are the index limits between molecular structures.

To stabilize computations, we performed preconditioning in float64.

F Extensivity on hydrogen chains

Gao & Günnemann (2023a) and Scherbela et al. (2024) analyzed the behavior of their wave functions on hydrogen chains to investigate the extensivity of their wave functions. They did so by training the generalized wave functions on a set of hydrogen chains with 6 and 10 elements. Then, they evaluated the energy per atom on hydrogen chains with different lengths. We replicated their experiment and trained a single NeurPf on the hydrogen chains with 6 and 10 atoms and evaluated the energy per atom on hydrogen chains of increasing lengths.

Fig. 6 shows the energy per atom of hydrogen chains with different lengths for various methods, Globe+Moon and Globe+FermiNet from Gao & Günnemann (2023a), Scherbela et al. (2024), Hartree-Fock (CBS), the AFQMC limit for an infinitely long chain (Motta et al., 2017), and NeurPf. It is apparent that NeurPf outperforms Globe+Moon and Globe+FermiNet significantly by achieving significantly lower energies outside of the training regime. Compared to Scherbela et al. (2024), NeurPf generally performs better on longer chains, achieving errors below the Hartree-Fock baseline. However, we observe significantly higher errors in the shortest chains in NeurPf.

These results indicate that NeurPf is better at generalizing to longer chains than previous works despite not including additional Hartree-Fock calculations like Scherbela et al. (2024).

G Metal ionization energies

In addition to the results in Sec. 5, where we train on all second-row elements and their ionization and affinity potentials, we here train a single NeurPf on a set of metals and their ionization energies. This demonstrates that Neural Pfaffians also scale to heavier 3rd and 4th row elements. Fig. 7 shows the ionization energy during training. It is apparent, that NeurPf can learn a solution for all states simultaneously.

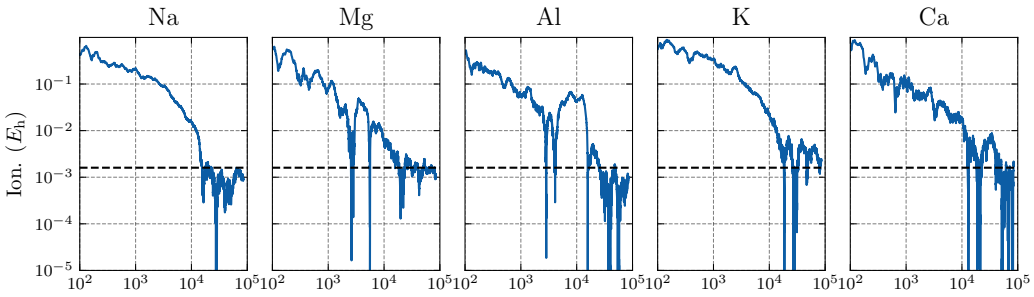


Fig. 7: Ionization energies of metal atoms. The ionization energies are computed with a single NeurPf trained on the neutral and ionized atoms. Reference energies are taken from Martin & Musgrove (1998).

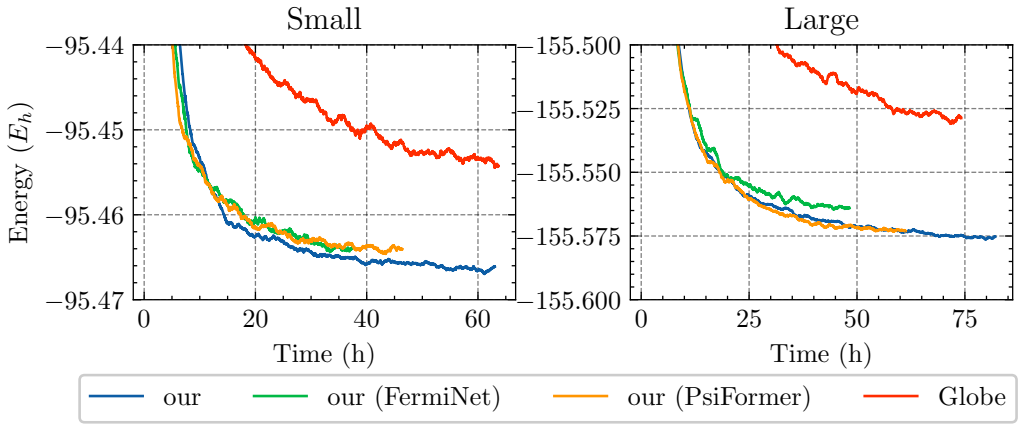


Fig. 8: Energy convergence as a function of time.

H TinyMol convergence in time

In Fig. 8, we show the runtime effect of choosing different embedding and antisymmetrizer. We test our default model, our (FermiNet), our (PsiFormer) and Globe + Moon on both TinyMol datasets. For any time budget, all variants of NeurPf converge to lower energies than Globe.

I Convergence ablation studies

Here, we provide additional ablation studies to further investigate the performance of NeurPf and our efficient envelopes. In particular, we train four different models on the small TinyMol dataset: NeurPf, NeurPf with the envelopes from Spencer et al. (2020), NeurPf with the envelopes from Pfau et al. (2024), and an AGP-based generalized wave function.

The total energy during training is shown in Fig. 9. The left plot shows the convergence regarding the number of steps, and the right plot shows the convergence in terms of time. We observe that NeurPf convergence is consistently faster than the other methods in terms of the number of steps and time. One further sees the importance of generalizing Eq. (9) via the Pfaffian as the AGP-based wave function does not converge to the same accuracy as NeurPf. The bottleneck envelopes from Pfau et al. (2024) do not only converge to worse energies but are also slower per step than our efficient envelopes from Sec. 4.2.

J Model ablation studies

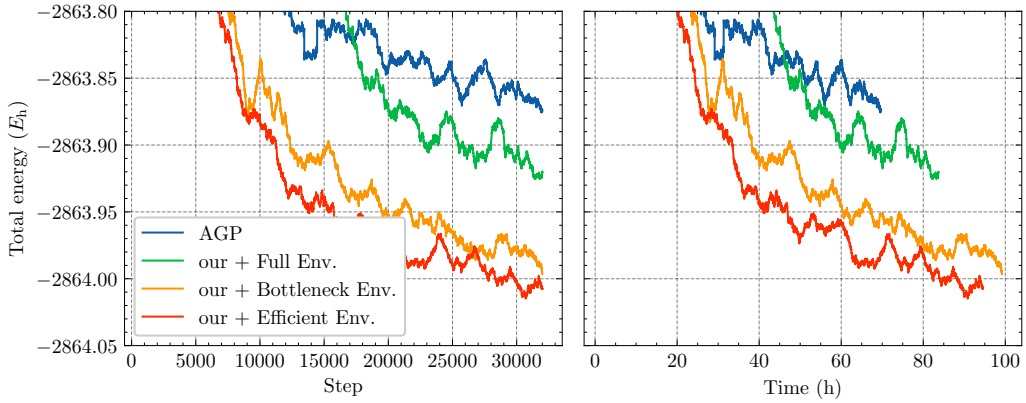


Fig. 9: Ablation study on the small TinyMol dataset. The y-axis shows the sum of all energies in the dataset. The left plot shows the convergence in terms of the number of steps. The right plot shows the convergence in terms of time. our + Full Env. shows a NeurPf with the envelopes from Spencer et al. (2020) and our + Bottleneck Env. uses the bottleneck envelopes from Pfau et al. (2024).

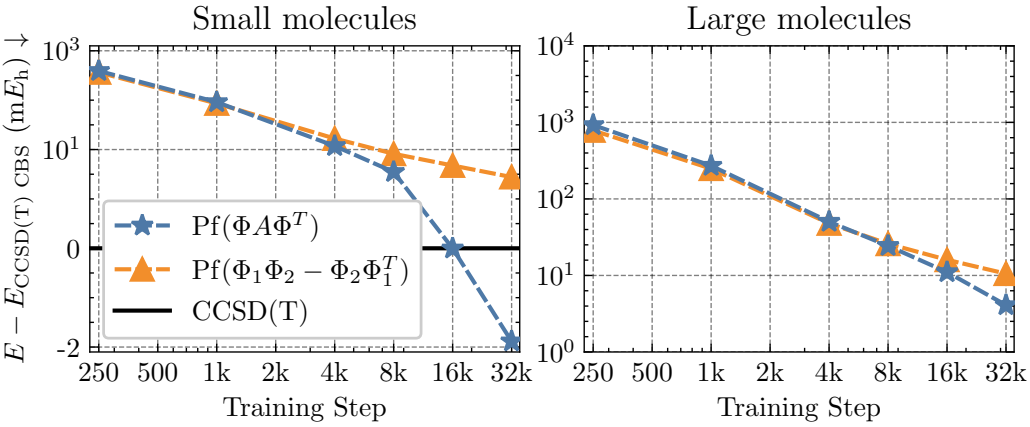


Fig. 10: TinyMol ablation with fixed and learnable antisymmetrizer.

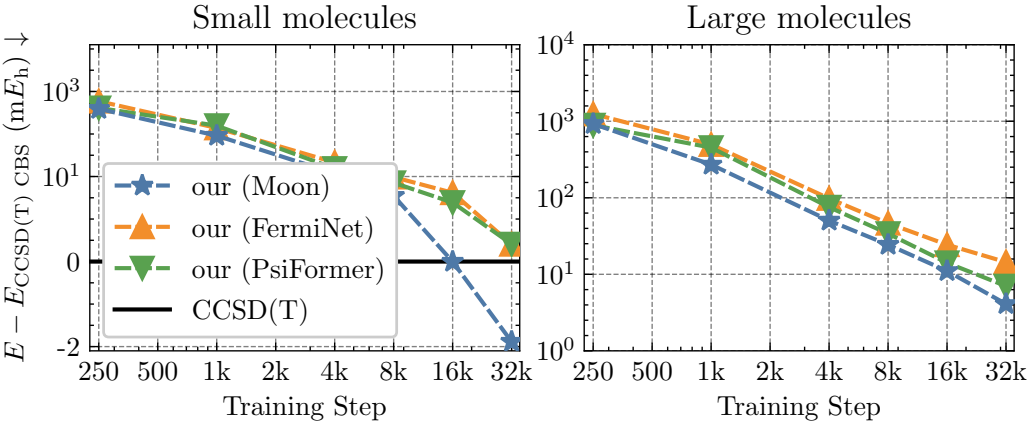


Fig. 11: Ablation study on the small TinyMol dataset with different embedding networks.

Table 4: TinyMol energies compared to CCSD(T) in $\mathrm{m}E_\mathrm{h}$.

Method (Steps)	Small CNH	Small C_2H_4	Small COH_2	Large C_3H_4	Large CN_2H_2	Large CNOH	Large CO_2
Globe (32k)	5.2	12.3	10.7	62.3	45.8	40.4	42.7
TAO (32k)	1.1	4.5	6.6	18.7	21.0	41.9	19.6
our (32k)	**-3.7**	**0.1**	**-2.1**	**12.7**	**5.5**	**3.1**	**5.0**
our (128k)	-4.2	-1.5	-3.7	1.4	-3.8	-6.9	-8.2

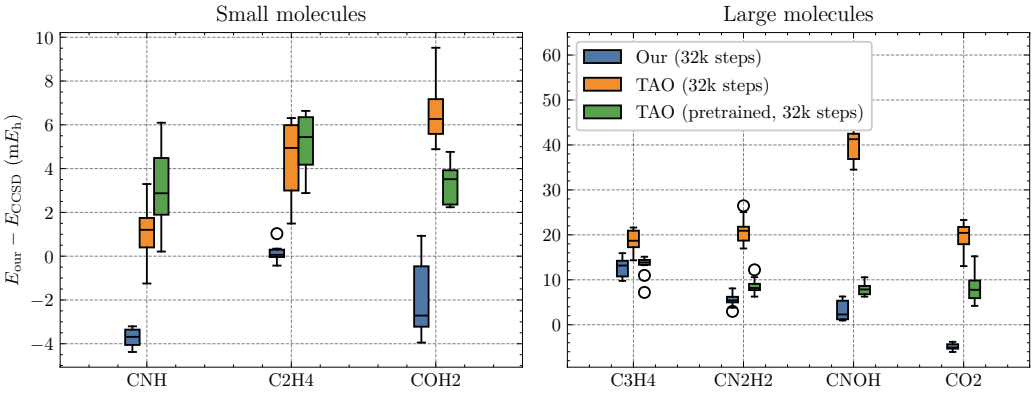


Fig. 12: Boxplot of the energy per molecule on both TinyMol small and large datasets for NeurPf, TAO, and the pretrained TAO from Scherbela et al. (2024). Each boxplot contains results from 10 structures for the given molecule. The line indicates the mean, the box the interquartile range, and the whiskers the 1.5 times the interquartile range.

J.1 Learnable antisymmetrizer

We picked $\mathrm{Pf}(\Phi A \Phi^T)$ as parametrization because it generalizes Slater determinants and many alternative parametrizations. For instance, by choosing $A = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$ and $\Phi = (\Phi_1 \ \Phi_2) = \mathrm{Pf}(\Phi A \Phi^T) \implies \mathrm{Pf}(\Phi_1 \Phi_2^T - \Phi_2 \Phi_1^T)$. We investigate the impact of having A being fixed/learnable in Fig. 10. The results suggest that having A being learnable is a significant factor in our Neural Pfaffian's accuracy.

J.2 Embedding network

Since NeurPf is not limited to Moon, we performed additional ablations with FermiNet (Pfau et al., 2020) and PsiFormer (von Glehn et al., 2023) as the embedding. The results in Fig. 11 show Neural Pfaffians outperforming Globe and TAO with any of the three equivariant embedding models. Consistent with Gao & Günnemann (2023a), Moon is the best choice for generalized wave functions.

K TinyMol results

Here, we provide additional data analysis and error metrics for the TinyMol dataset. First, we show in Table 4 the energy per molecule for the small and large TinyMol datasets for NeurPf, Globe, and TAO. To estimate the remaining error, we also train another NeurPf for 128k steps. The results show that NeurPf consistently outperforms TAO and Globe on all molecules in both datasets.

Second, we show the error per molecule for both the small and large TinyMol datasets in Fig. 12. We plot all models after 32k steps of training. It is apparent that NeurPf consistently results in lower, i.e., better, energies than TAO on all molecules in both datasets. Even the pretrained TAO is outperformed by NeurPf on all but four structures of C3H4 in the large TinyMol dataset.

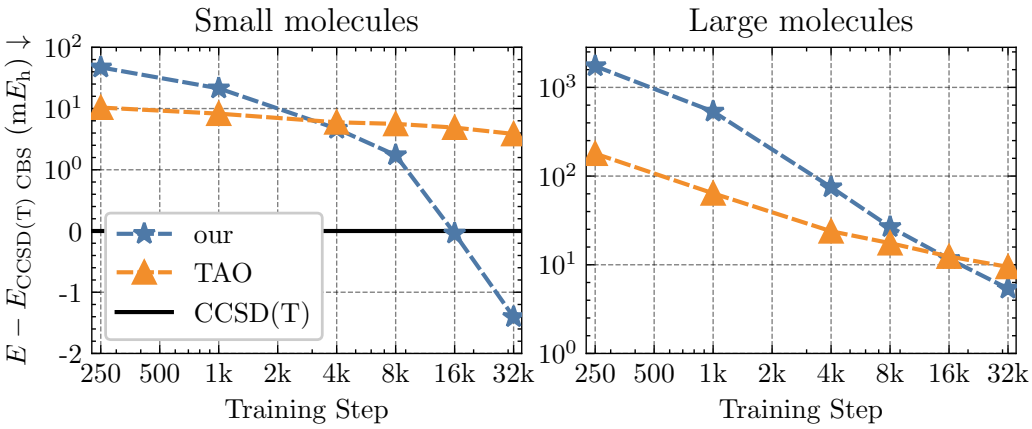


Fig. 13: TinyMol results with pretraining on the training set.

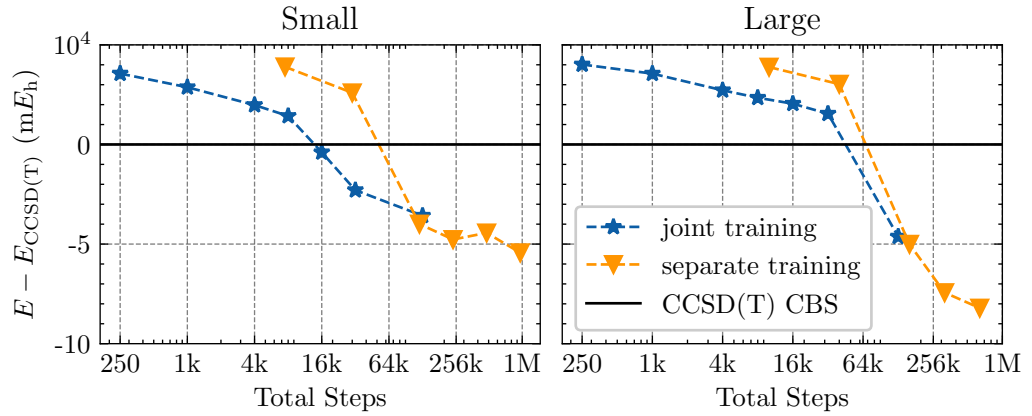


Fig. 14: Comparison of the energy per molecule on the TinyMol dataset for training jointly on all structures vs training a model per structure.

L Pretraining on the TinyMol dataset

The TinyMol provides an additional pretraining set of 360 structures (18 molecules, 20 structures each). Like Scherbela et al. (2024), we pretrain our model on the training set of the TinyMol dataset and then finetune on the two test sets. Interestingly, we find the Spring optimizer to be unstable when swapping molecules from step to step and, thus, use CG-preconditioning like Gao & Günnemann (2023a) during pretraining. While yielding a small benefit on the small molecules, we find no notable difference to the Hartree-Fock pretrained model on the large molecules as shown in Fig. 13. On the small structures, the unpretrained NeurPf's energies are $5.7\,\mathrm{m}E_\mathrm{h}$ lower. NeurPf also surpasses the pretrained TAO after just 8k steps. Compared to the pretrained TAO on the large structures, NeurPf surpasses TAO after 16k steps and achieves $5.4\,\mathrm{m}E_\mathrm{h}$ lower energies after 32k steps.

M Joint vs separate training

To estimate the benefit of training a generalized wave function compared to training a model per molecule, we compare the convergence of the total energy on the TinyMol dataset for both approaches depending on the total number of training steps. As training a separate model for each of the 70 TinyMole test molecules is computationally beyond the scope of this work, we select on structure per molecules and train a model for each of the 7 molecules. We use the same NeurPf with MetaGNN for both approaches. The results are shown in Fig. 14. We observe that for lower step numbers, it is quite beneficial to train a generalized model. Though, this benefit vanishes for higher step numbers, and

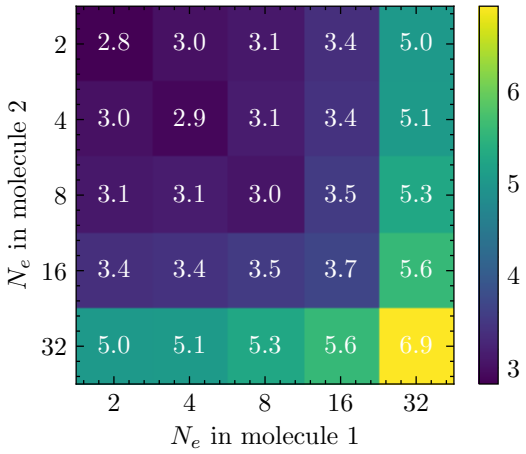


Fig. 15: Time per training step depending on the number of electrons in two molecules.

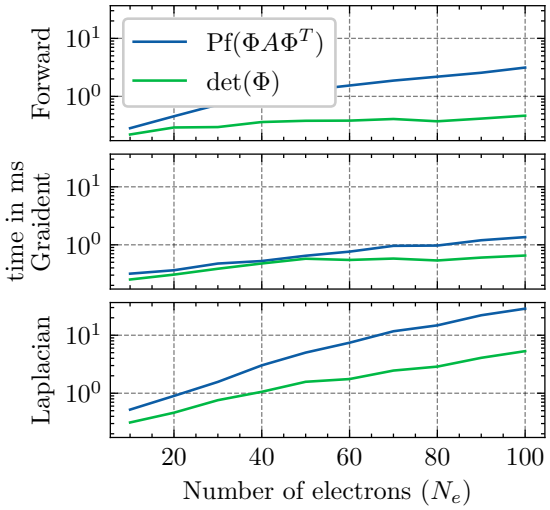


Fig. 16: Time for the forward pass, gradient and Laplacian computation of determinant vs our Pfaffian implementation.

training a model per molecule yields lower energies. We attribute this to the fact that the generalized model has to learn a more complex representation that is not necessary for each molecule individually. Further, the per-molecule energy estimates are quite unstable due to the small shared batch size. Developments like Scherbela et al. (2023) may improve NeurPf training as well.

N Training time by batch composition

Here, we benchmark the total time per step for a two-molecule batch. We test all combinations of two molecules with $N_e^1, N_e^2 \in 2, 4, 8, 16, 32$. While we find a small runtime increase when processing small molecules jointly in Fig. 15, for larger systems, we see the runtime per step converge to the geometric mean of the individual runtimes.

O Pfaffian runtime

In Fig. 16, we benchmark our implementation for $\mathrm{Pf}(\Phi A \Phi^T)$ (incl. the matrix multiplications) against the standard operation of $\det \Phi$ for 10 to 100 electrons. We implement the Pfaffian in JAX

while highly optimized CUDA kernels are available for the determinant. In summary, both share the same complexity of $O(N^3)$, but the Pfaffian is approximately 5 times slower.

P Broader impact

Highly accurate quantum chemical calculations are essential for understanding chemical reactions and materials properties. Our work contributes to this development by providing accurate neural network quantum Monte Carlo calculations at broader scales thanks to generalized wave functions. While this may be used to distill more accurate force fields or exchange-correlation functionals for DFT, the societal impact of our work is primarily in the scientific domain due to the high computational cost of neural network VMC. To the best of our knowledge, our work does not promote any negative societal impact more than general theoretical chemistry research does.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim the following in our abstract and introduction: (1) Neural Pfaffian are applicable to any molecular system. As outlined in Section 4.4, we ensure this by parametrizing the orbitals to be always larger than the number of electrons. (2) Neural Pfaffian can learn all second-row element systems' ground state, ionization, and electron affinity energies. We demonstrate this in our first experiment in Section 5. (3) Neural Pfaffian outperforms Globe on the nitrogen dimer. See the second experiment in Section 5.(4) We outperform CCSD(T) CBS on the small structures in TinyMol and TAO by factors of 10 and 6 on small and large structures, respectively. See the third experiment in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: At the end of Section 4.4, we list the limitations of our work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper does not contain theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 gives the mathematical definition of our new contribution. Appendix D details the exact model definitions. Appendix E lists all hyperparameters, and as we explain in Appendix E.2, we provide the source code to reviewers and publish it publicly upon publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the source code via OpenReview to the reviewers as mentioned in Appendix E.2. The code will be made publicly available upon publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: At the experimental setups (Section 5), we list the original references for structures and energies. Hyperparameters and additional details are listed in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: As common in deep learning-based quantum Monte Carlo literature, we do not repeat experiments for different seeds due to their computational cost, see Appendix E.3, and their generally low deviations across runs. We omit error bars due to numerical integration as these are typically below the readability threshold $\approx 0.1\,\mathrm{m}E_\mathrm{h}$.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We list compute resources and time in Appendix E.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work respects the NeurIPS Code of Ethics in every aspect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We discuss this in App. P.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We strongly believe that there is no higher danger of misuse for our work than for traditional methods in computational chemistry, especially not at the scale where neural wave functions are applicable.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We list all sources for our molecular structures and reference energies at the appropriate places in Section 5. Further, we cite other codes we base our implementation of in Appendix E.2 and E.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Upon publication, we will publish our source code as a new asset publically under the MIT license. Before that, we provide an early version to the reviewers via OpenReview, see Appendix E.2.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: Our research does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

D *Publication 4*: Sampling-free Inference for Ab-Initio Potential Energy Surface Networks

Publication: Nicholas Gao and Stephan Günnemann. Generalizing Neural Wave Functions. In *International Conference on Machine Learning*, 2023a

Location: Honolulu, USA

License: The rights of the publication are held by the authors; from the respective web page `https://iclr.cc/FAQ/Copyright`, June 9th 2025:

> Authors do not transfer the copyright of their paper to ICLR, instead they grant ICLR a non-exclusive, perpetual, royalty-free, fully-paid, fully-assignable license to copy, distribute and publicly display all or part of the paper.

Therefore, no explicit permission is required to include the publications in this thesis.

SAMPLING-FREE INFERENCE FOR AB-INITIO POTENTIAL ENERGY SURFACE NETWORKS

Nicholas Gao, Stephan Günnemann
Department of Computer Science & Munich Data Science Institute
Technical University of Munich, Germany
{n.gao,s.guennemann}@tum.de

ABSTRACT

Recently, it has been shown that neural networks not only approximate the ground-state wave functions of a single molecular system well but can also generalize to multiple geometries. While such generalization significantly speeds up training, each energy evaluation still requires Monte Carlo integration which limits the evaluation to a few geometries. In this work, we address the inference shortcomings by proposing the **P**otential **l**earning from **a**b-initio **Net**works (PlaNet) framework, in which we simultaneously train a surrogate model in addition to the neural wave function. At inference time, the surrogate avoids expensive Monte-Carlo integration by directly estimating the energy, accelerating the process from hours to milliseconds. In this way, we can accurately model high-resolution multi-dimensional energy surfaces for larger systems that previously were unobtainable via neural wave functions. Finally, we explore an additional inductive bias by introducing physically-motivated restricted neural wave function models. We implement such a function with several additional improvements in the new PESNet++ model. In our experimental evaluation, PlaNet accelerates inference by 7 orders of magnitude for larger molecules like ethanol while preserving accuracy. Compared to previous energy surface networks, PESNet++ reduces energy errors by up to $74\,\%$.

1 INTRODUCTION

Solving the Schrödinger equation is key to assessing a molecular system's quantum mechanical (QM) properties. As analytical solutions are unavailable, one must rely on expensive approximate solutions. Such methods are called *ab-initio* if they do not rely on empirical data. Recently, neural networks succeeded in such ab-initio calculations within the variational Monte-Carlo (VMC) framework (Carleo & Troyer, 2017). Although they lead to accurate energies, training such neural wave functions proved to be computationally intensive (Pfau et al., 2020).

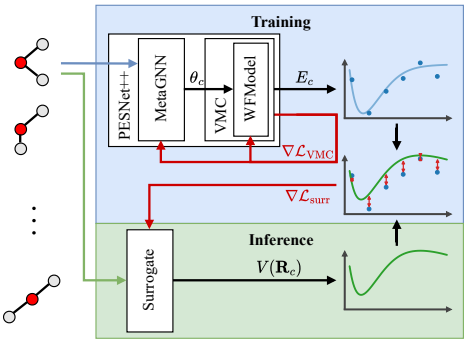


Figure 1: PlaNet framework. During training, we use the noisy energies obtained during the VMC optimization to fit our surrogate. At inference time we only query the surrogate to avoid costly numerical integration.

To reduce the computational burden, Gao & Günnemann (2022) proposed the potential energy surface network (PESNet) to simultaneously solve many Schrödinger equations, i.e., for different spatial arrangements of the nuclei in $\mathbb{R}^3$. They use a GNN to reparametrize the wave function model based on the molecular structure. While training significantly faster, afterward one only obtains a neural wave function that generalizes over a domain of geometries, but not the associated energy surface. Obtaining the energy of a geometry remains costly requiring a Monte-Carlo integration with complexity scaling as $O(N^4)$ in the number of electrons N. This high inference time prohibits many applications for neural wave functions. For instance, geometry optimization, free energy calculation, potential energy surface scans, or molecular dynamics simulations typically involve hundreds of thousands of energy evaluations (Jensen, 2010; Hoja et al., 2021).

To address the inference shortcomings, we propose the **P**otential **l**earning from **a**b-initio **Net**works (PlaNet) framework, where we utilize intermediate results from the PESNet optimization to train a surrogate graph neural network (GNN) as illustrated in Figure 1. In optimizing PESNet, one must compute approximate energy values to evaluate the loss. We propose to use these noisy intermediate energies, which otherwise would be discarded, as training labels for the surrogate. After training, the surrogate accelerates inference by directly estimating energies bypassing Monte Carlo integration.

As a second contribution, we adapt neural wave functions for closed-shell systems, i.e., systems without unpaired electrons, by introducing restricted neural wave functions. Specifically, we use doubly-occupied orbitals as inductive bias like restricted Hartree-Fock theory (Szabo & Ostlund, 2012). Together with several other architectural improvements, we present the improved PESNet++.

In our experiments, PESNet++ significantly improves energy estimates on challenging molecules such as the nitrogen dimer where PESNet++ reduces errors by $74\,\%$ compared to PESNet. When analyzing PlaNet we find it to accurately reproduce complex energy surfaces well within chemical accuracy across a range of systems while accelerating inference by 7 orders of magnitude for larger molecules such as ethanol. To summarize our contributions:

- **PlaNet**: an orders of magnitude faster inference method for PESNet(++), enabling exploration of higher dimensional energy surfaces at no loss in accuracy.
- **PESNet++**: an improved neural wave function for multiple geometries setting new state-of-the-art results on several energy surfaces while training only a single model.

2 Related Work

Neural wave functions. Variational Monte Carlo calculations rely on expressive function forms to approximate the ground-state wave function. While early single determinants with linear combinations of basis functions without any explicit electron-electron interaction were used (Slater, 1929), later backflow (Feynman & Cohen, 1956) and Jastrow factors (Jastrow, 1955) directly introduced electron correlation effects. While the combination of both has shown success (Brown et al., 2007), Carleo & Troyer (2017) were the first to systematically exploit the expressiveness of neural networks as wave functions. While initial works targeted small systems (Kessler et al., 2021; Han et al., 2019; Choo et al., 2020), FermiNet (Pfau et al., 2020; Spencer et al., 2020), and PauliNet (Hermann et al., 2020) introduced scalable approaches. Building on their success, recent works proposed integration into diffusion Monte-Carlo (DMC) (Wilson et al., 2021; Ren et al., 2022), introduction of pseudopotentials (Li et al., 2022a), architectural improvements (Gerard et al., 2022), and extension to periodic systems (Wilson et al., 2022; Li et al., 2022b; Cassella et al., 2022). Finally, two approaches to scaling neural wave functions to multiple geometries have been explored. Scherbela et al. (2022) proposed a weight-sharing scheme across geometries to reduce the number of iterations per geometry, while Gao & Günnemann (2022) directly reparametrize the wave function with an additional neural network eliminating the need for retraining.

Machine learning potentials. The use of machine learning models as surrogates for quantum mechanical calculations has a rich history, e.g., one of the first force fields has been the Merck Molecular Force Field (MMFF94) (Halgren, 1996). Later, kernel methods were used (Behler, 2011; Bartók et al., 2013; Christensen et al., 2020), while graph neural networks currently obtain state-of-the-art results (Schütt et al., 2018; Gasteiger et al., 2019; 2021). Despite significant progress in the field by proving universality for certain model classes (Thomas et al., 2018; Gasteiger et al., 2021), the need for data and label quality remain limiting factors. Thus, the line between ab-initio methods and ML potentials has blurred in recent years, either by the integration of QM calculations into ML models (Qiao et al., 2020; 2021), Δ-ML methods (Wengert et al., 2021), or by integrating ML models into QM calculations (Snyder et al., 2012; Kirkpatrick et al., 2021).

3 Background

Notation For consistency with previous work, we largely follow the notation by Gao & Günnemann (2022). We use 'geometries of a molecule' to refer to different spatial arrangements of the same set of atoms in $\mathbb{R}^3$. We use C to denote the number of geometries, B for the number of electron configurations per geometry, N for the number of electrons, and M for the number of nuclei. For

electrons, we use $\mathbf{r} \in \mathbb{R}^{C\times B\times N\times 3}$ for the full tensor, $\boldsymbol{r} \in \mathbb{R}^{B\times N\times 3}$ for a batch of electrons of a single geometry, $\mathbf{r} \in \mathbb{R}^{N\times 3}$ for a single set of electrons, and $\boldsymbol{r} \in \mathbb{R}^3$ for a single electron. Similarly, we use $\mathbf{R} \in \mathbb{R}^{C\times M\times 3}$, $\mathbf{R} \in \mathbb{R}^{M\times 3}$, and $\boldsymbol{R} \in \mathbb{R}^3$ to denote the nuclei tensor, a nuclei geometry, and a single nucleus, respectively. We assume all coordinates in 3D space have already been transformed into the equivariant coordinate frame from Gao & Günnemann (2022) and drop the explicit transformation. We denote the local energy tensor as $\boldsymbol{E} \in \mathbb{R}^{C\times B}$ and refer to a single energy value by $E_{c,i}$. Finally, we use bracketed superscripts to index sequences, e.g., layers in a network$^{(l)}$ or training steps$^{(t)}$.

Variational Monte Carlo. To obtain the ground-state energy for a fixed molecular system, one has to solve the associated time-independent Schrödinger equation

$$\mathbf{H}|\psi\rangle = E|\psi\rangle. \tag{1}$$

where $\psi : \mathbb{R}^{N\times 3} \to \mathbb{R}$ is the wave function, $E \in \mathbb{R}$ is the energy and $\mathbf{H}$ is the Hamiltonian operator

$$\mathbf{H} = -\frac{1}{2}\sum_{i=1}^{3N}\nabla_i^2 + \underbrace{\sum_{i>j=1}^{N}\frac{1}{\|\boldsymbol{r}_i-\boldsymbol{r}_j\|} + \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{Z_m}{\|\boldsymbol{r}_i-\boldsymbol{R}_m\|} + \sum_{m>n=1}^{M}\frac{Z_m Z_n}{\|\boldsymbol{R}_m-\boldsymbol{R}_n\|}}_{V(\mathbf{r})} \tag{2}$$

with ∇^2, the Laplacian operator, and $V(\mathbf{r})$ describing the kinetic energy and potential energy, respectively. In linear algebra, Equation (1) is an eigenvalue problem where we are interested in the lowest eigenvalue E_0 which is also called the ground-state energy. In VMC, we accomplish this by iteratively updating the parameters θ of a trial wave function ψ_θ to approximate the ground state ψ_0 (McMillan, 1965). The final accuracy of VMC is mostly determined by the function class of ψ_θ. Luckily, a trial function must only fulfill two requirements, first, it must be anti-symmetric to electron permutations, i.e., $\psi_\theta(\pi(\mathbf{r})) = -\operatorname{sign}(\pi)\psi_\theta(\mathbf{r})$ for all permutation π, and its integral must be finite. This enables us to use neural networks as wave functions (Carleo & Troyer, 2017). At each step, we seek to minimize the energy

$$\mathcal{L}_{\text{VMC}} = \frac{\langle\psi_\theta|\mathbf{H}|\psi_\theta\rangle}{\langle\psi_\theta|\psi_\theta\rangle} = \frac{\int\psi_\theta(\mathbf{r})\mathbf{H}\psi_\theta(\mathbf{r})\mathrm{d}\mathbf{r}}{\int\psi_\theta^2(\mathbf{r})\mathrm{d}\mathbf{r}} = \frac{\int\psi_\theta^2(\mathbf{r})\psi_\theta^{-1}(\mathbf{r})\mathbf{H}\psi_\theta(\mathbf{r})\mathrm{d}\mathbf{r}}{\int\psi_\theta^2(\mathbf{r})\mathrm{d}\mathbf{r}}. \tag{3}$$

Following the standard procedure (Ceperley et al., 1977), we numerically approximate the integral via importance sampling by interpreting $\frac{\psi_\theta^2(\mathbf{r})}{\int\psi_\theta^2(\mathbf{r})\mathrm{d}\mathbf{r}}$ as a probability distribution of electron configurations $\mathbf{r}$. For each electron configuration, we then compute the so-called local energy

$$E_L(\mathbf{r}) = \psi_\theta^{-1}(\mathbf{r})\mathbf{H}\psi_\theta(\mathbf{r}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{3}\left[\frac{\partial^2\log|\psi_\theta(\mathbf{r})|}{\partial r_{ik}^2} + \frac{\partial\log|\psi_\theta(\mathbf{r})|}{\partial r_{ik}}^2\right] + V(\mathbf{r}). \tag{4}$$

We then use these local energies to obtain the gradients of our wave function parameters θ via

$$\nabla_\theta\mathcal{L}_{\text{VMC}} = \mathbb{E}_{\psi_\theta^2}\left[\Big(E_L(\mathbf{r}) - \mathbb{E}_{\psi_\theta^2}[E_L(\mathbf{r})]\Big)\nabla_\theta\log|\psi_\theta(\mathbf{r})|\right] \tag{5}$$

where we approximate all expectations with Monte-Carlo samples obtained via Metropolis Hastings, i.e., via a Monte Carlo Markov Chain (MCMC) starting from the positions of the last iteration.

As the trial wave function ψ_θ approaches the ground state ψ_0, local energies approach the ground-state energy E_0, and their variance zero. Furthermore, our energy estimate $\mathbb{E}_{\psi_\theta^2}[E_L(\mathbf{r})]$ is lower bounded by E_0 because the eigenfunctions of $\mathbf{H}$ are a basis of all functions with the aforementioned properties.

While noisy estimates of the energy $\mathbb{E}_{\psi_\theta^2}[E_L(\mathbf{r})]$ are sufficient for training, at inference time, one typically has to draw $O(10^6)$ samples (Gao & Günnemann, 2022; Ren et al., 2022) to accurately model the energy for a molecular system.

PESNet. Since one is mostly interested in comparing energies of different geometries in chemistry, Gao & Günnemann (2022) proposed PESNet to avoid solving many Schrödinger equations independently but rather exploit the correlation between them. They accomplished this generalization by decomposing the problem into two neural networks, a wave function model (WFModel) representing the wave function ψ_θ, and a problem-adapting GNN (MetaGNN) that updates the WFModels's parameters θ based on the geometry. Thus, each VMC step becomes a two-step process where one

first obtains the wave function parameters θ_c for each geometry $c \in \{1, \dots, C\}$ via the MetaGNN and then computes the local energies and gradients with the WFModel ψ_{θ_c} via VMC. So, instead of obtaining energy estimates for the same fixed structure, during optimization, we get local energy estimates $\boldsymbol{E} \in \mathbb{R}^{C \times B}$, where $E_{c,i} = E_L(\mathbf{r}_{c,i})$, for a batch of C geometries $\mathbf{R} \in \mathbb{R}^{C \times M \times 3}$.

Despite the unified training, one still needs to perform independent numerical integration per geometry to obtain energies for multiple geometries. As the cost per evaluation $O(N^4)$ grows with the number of electrons, evaluations are typically restricted to tens of geometries for small molecules due to computational constraints. Our PlaNet framework addresses these inference limitations enabling the evaluation of thousands of geometries, even for larger molecules, see Section 6.2.

4 PLANET – LEARNING POTENTIALS FROM NOISY INTERMEDIATE ENERGIES

To avoid the numerical integration for each geometry at inference time, we propose reusing the intermediate noisy energies $\boldsymbol{E} \in \mathbb{R}^{C \times B}$ from the optimization of PESNet(++) to train a surrogate model. This surrogate model only acts on the nuclei and outputs the energy directly, i.e., in contrast to the wave function, there is no dependence on the electrons. At inference time, we only need to query the surrogate, reducing the inference time from hours to milliseconds.

As we optimize the energy throughout the training, we must deal with the change in target labels over time. We avoid this by treating the training process as an online learning problem where we use samples only once in the order in which they were generated. This online learning biases the network towards newer lower (better) energies.

To systematically incorporate physical invariances of the energy towards the Euclidean group E(3), i.e., translation, rotation, and reflections, we use a GNN as the surrogate. Compared to traditional internal coordinate-based polynomials (Jiang & Guo, 2013; Li et al., 2013), GNNs offer straightforward generalization to larger molecules without manual feature engineering. Though, as purely distance-based GNNs are incomplete (**?**), we use the angle-encoding DimeNet^{++} (Gasteiger et al., 2020).

In each training step, we train the surrogate with the current geometries $\mathbf{R}^{(t)}$ and energies $\mathbf{E}^{(t)}$ as input and target, respectively. As loss we optimize the weighted root mean squared error (RMSE)

$$\mathcal{L}_{\text{surr}}^{(t)} = \sqrt{\frac{1}{C} \sum_{c=1}^{C} \frac{\left(\hat{E}_c^{(t)} - V_\chi(\mathbf{R}_c^{(t)})\right)^2}{\hat{\sigma}_c^{(t)}}} \tag{6}$$

where $\hat{E}_c^{(t)} = \frac{1}{B} \sum_{i=1}^{B} E_{c,i}^{(t)}$, $\hat{\sigma}_c^{(t)} = \frac{1}{B} \sqrt{\sum_{i=1}^{B} (E_{c,i}^{(t)} - \hat{E}_c^{(t)})^2}$, and $V_\chi : \mathbb{R}^{M \times 3} \to \mathbb{R}$ denotes the forward pass of DimeNet^{++} with parameters χ. Since we train in an online fashion, we perform N_{surr} many gradient steps at each iteration t to better utilize the available data.

Accounting for label noise. Since we obtain the energies from the VMC optimization described in Section 3, the labels are subject to noise. In fact, the observed noise is often orders of magnitude larger than the target energy differences we want to learn. To avoid the model oscillating between noisy labels, we encourage temporal data aggregation by applying an exponential moving average (EMA) to the weights χ as traditionally done in surrogate models (Gasteiger et al., 2019; 2021; Schütt et al., 2021). However, picking the decay factor for the EMA is non-trivial as large decay values may cause the model to oscillate while small decay values may cause slow convergence. To resolve this issue, we recognize that the mean absolute error (MAE) between the surrogate and the VMC estimates is lower bound by the mean absolute deviation (MAD) of the energy distribution. We exploit this to identify two regimes, an initial regime where the PlaNet training error is higher than the energy label error and one where it is lower. In the first regime, we choose a relatively high decay factor for the moving average to adapt quickly, while we shrink the decay in the second regime to encourage temporal data aggregation. To this end, we control the decay factor via

$$\gamma^{(t)} = \gamma_{\text{base}} + \gamma_{\text{high}} \mathbb{1}(\mathcal{L}_{\text{surr}}^{(t)} < \zeta D^{(t)}) \tag{7}$$

with $0 < \gamma_{\text{base}} + \gamma_{\text{high}} < 1$ and $\zeta > 1$ being hyperparameters and $\mathbb{1}$ being the indicator function. Since the true MAD $D^{(t)}$ at iteration t is unknown, we must estimate it. By the central limit

theorem, we assume that the energies are approximately normally distributed with standard deviation $\hat{\sigma}^{(t)} = \frac{1}{C}\sum_{c=1}^{C}\hat{\sigma}_c^{(t)}$, thus, we approximate the MAD as $D^{(t)} \approx \sqrt{\frac{2}{\pi}}\hat{\sigma}_t$.

5 PESNET++ – IMPROVING MULTI-GEOMETRY NEURAL WAVE FUNCTIONS

PESNet++ builds on the recently proposed PESNet, discussed in Section 3. In this work, we adopt several improvements to enable higher-accuracy energy surfaces at a similar cost. We use restricted neural wave functions for closed-shell systems, switch from block-diagonal orbital matrices to dense ones, add a permutation invariant component to the wave function, and propose a coordinate transformation to enable larger geometric perturbations during training. Appendix A includes a complete definition of the new PESNet++ model.

Dense orbitals. Pfau et al. (2020); Hermann et al. (2020) and Gao & Günnemann (2022) define the wave function as a linear combination of products of determinants of spin-up and spin-down orbitals

$$\psi(\mathbf{r}) = \sum_{k=1}^{K} w_k \det \phi^{k\uparrow} \det \phi^{k\downarrow} = \sum_{k=1}^{K} w_k \det \begin{bmatrix} \phi^{k\uparrow} & 0 \\ 0 & \phi^{k\downarrow} \end{bmatrix}, \phi^{k\alpha} \in \mathbb{R}^{N^\alpha \times N^\alpha} \tag{8}$$

where $\alpha \in \{\uparrow, \downarrow\}$ indicates the spin and $\boldsymbol{w} \in \mathbb{R}^k$ are learnable weights. Equivalently, one may see the product of determinants as a determinant of an $N \times N$ block-diagonal matrix. Instead of restricting the orbital matrix to be block-diagonal, we adopt the improvements discovered by (Pfau et al., 2020) in the official FermiNet repository and define a dense orbital matrix as

$$\psi(\mathbf{r}) = \sum_{k=1}^{K} w_k \det \phi^k, \phi^k = \begin{bmatrix} \phi^{k\uparrow} \\ \phi^{k\downarrow} \end{bmatrix} \in \mathbb{R}^{N \times N}, \phi^{k\alpha} \in \mathbb{R}^{N^\alpha \times N}. \tag{9}$$

This is equivalent to picking N orbital functions for each spin instead of N^α ones for each spin α. Lin et al. (2021); Ren et al. (2022) and Gerard et al. (2022) have shown this to improve variational energies.

Restricted Neural Wave Functions. The Fermi-Dirac statistic only applies to electrons of the same spin requiring us to separate between spin-up $\uparrow$ and spin-down $\downarrow$ electrons. But, as most systems of interest, e.g., stable molecules, are closed-shell, i.e., systems without unpaired electrons, all orbitals will be doubly occupied (Szabo & Ostlund, 2012). We adopt this closed-shell formulation as inductive bias by enforcing identical orbital functions for both spins. One may see this as a neural analog to restricted Hartree-Fock (RHF), where one uses the same orbital functions for both spins, whereas previous work is related to unrestricted Hartree-Fock (UHF) with independent orbitals for both spins (Szabo & Ostlund, 2012). While UHF traditionally results in lower variational energies, we found this inductive bias to be better suited for neural wave functions, see Section 6, potentially due to its easier optimization by parameter sharing.

Speaking in symmetries, this restriction introduces invariance to the exchange of all spin-up and spin-down electrons. As both spin states are physically identical, all observables and, thus, the amplitude of the wave function stays the same under the exchange of spin states, i.e., $|\psi(\mathbf{r}^\uparrow, \mathbf{r}^\downarrow)| = |\psi(\mathbf{r}^\downarrow, \mathbf{r}^\uparrow)|$.

We implement this restriction by reformulating the interaction layers to only depend on the spin equality between two particles rather than their absolute spins. To incorporate this, we reformulate the update rules of our interaction layer to

$$\boldsymbol{h}_i^{l+1} = \sigma\left(\boldsymbol{W}^l_{\text{single}}\left[\boldsymbol{h}_i^l, \sum_{j\in\mathbb{A}^{\alpha_i}} \boldsymbol{g}_{ij}^l, \sum_{j\in\mathbb{A}^{\tilde{\alpha}_i}} \boldsymbol{g}_{ij}^l\right] + \boldsymbol{b}^l_{\text{single}} + \boldsymbol{W}^l_{\text{global}}\left[\sum_{j\in\mathbb{A}^{\alpha_i}} \boldsymbol{h}_j^l, \sum_{j\in\mathbb{A}^{\tilde{\alpha}_i}} \boldsymbol{h}_j^l\right]\right), \tag{10}$$

$$\boldsymbol{g}_{ij}^{l+1} = \sigma(\boldsymbol{W}^l_{\alpha_i=\alpha_j}\boldsymbol{g}_{ij}^t + \boldsymbol{b}^l_{\alpha_i=\alpha_j}) \tag{11}$$

where $\boldsymbol{h}_i$ and $\boldsymbol{g}_{ij}$ are single and pairwise electron features, α_i is the spin of the i-the electron, $\tilde{\alpha}$ denotes the opposing spin of α, and $\mathbb{A}^\alpha$ is the index set of all α spin electrons. Subscripts $\alpha = \beta$ indicate different weights depending on whether the spins α and β are identical. While Wilson et al. (2021) and Gerard et al. (2022) studied similar update functions, we are the first to explore the restricted closed-shell setting where one restricts the orbital functions to be identical between spins.

Finally, it remains to reformulate the orbital construction as

$$\phi^k = \begin{bmatrix} \phi^{k\uparrow\uparrow} & \phi^{k\uparrow\downarrow} \\ \phi^{k\downarrow\uparrow} & \phi^{k\downarrow\downarrow} \end{bmatrix}, \tag{12}$$

$$\phi_{ij}^{k\alpha\beta} = \left((\boldsymbol{w}_i^{k,\alpha=\beta})^T \boldsymbol{h}_j^{(L_{\mathrm{WF}})} + b_i^{k,\alpha=\beta} \right) \sum_{m=1}^{M} \pi_{im}^{k\alpha=\beta} \exp(-\sigma_{im}^{k\alpha=\beta} \|r_j - R_m\|). \tag{13}$$

We initialize all $\boldsymbol{w}_i^{k,\neq}$ with zeros to ensure identical initial behavior to block-diagonal orbitals.

Jastrow factor. To ease optimization, we follow classical quantum mechanical methods and introduce a permutation invariant part to the wave function, the so-called *Jastrow* factor (Jastrow, 1955; Brown et al., 2007; Hermann et al., 2020). We implement the Jastrow factor $J : \mathbb{R}^{N\times D} \to \mathbb{R}$ as a permutation invariant function on the final electron embeddings $\boldsymbol{h}^{(L_{\mathrm{WF}})}$:

$$\psi(\mathbf{r}) = \exp\left(J\left(\boldsymbol{h}^{(L_{\mathrm{WF}})} \right) \right) \sum_{k=1}^{K} w_k \det \phi^k, \qquad J(\boldsymbol{h}) = \sum_{i=1}^{N} \mathrm{MLP}(\boldsymbol{h}_i). \tag{14}$$

Avoiding dead neurons. In recent years, the flow of information in neural networks has been heavily studied (Brock et al., 2022; 2021; Hendrycks & Gimpel, 2020). While such works have shown great success in classical deep learning tasks such as computer vision, they have not yet been applied to neural wave functions. We implement several such improvements: careful parameter initialization, rescaling after activation functions (Brock et al., 2022), and the SiLU activation function (Hendrycks & Gimpel, 2020). As all features are learned based on Euclidean distances, we still use $\tanh$ after the first layer to limit the magnitude of feature embeddings. We observed significant improvements in performance and trace it back to a significantly reduced number of dead neurons, i.e., the number of neurons whose value is independent of the input. We illustrate these observations in Appendix B.

Coordinate transform. In machine learning, one typically draws a batch of samples independently from some data distribution. Unfortunately, we cannot do the same for molecular geometries. Recall from Section 3, as we use an MCMC to sample the electron positions, we have to keep track of the last electron positions for each geometry to sample new ones. Thus, instead of i.i.d. sampling, Gao & Günnemann (2022) obtain the next batch of geometries by applying small perturbations to the previous one. While performing sufficiently well for low-dimensional energy surfaces, such small changes do not scale to higher-dimensional problems where the search space grows exponentially. To close the gap to the i.i.d. setting, we increase the magnitude of perturbations. However, this comes with the danger of decorrelating the electron positions from the nuclei. We minimize such decorrelation effects by transforming each electron in accord with its closest nucleus. This displacement avoids electron configurations from reaching dead regions of the probability distribution by guaranteeing that the distance between an electron and its closest nucleus may only decrease. Formally, if $\mathbf{R}' \in \mathbb{R}^{M\times 3}$ are the new nuclei positions we obtain the new electron positions $\mathbf{r}'$ via $\lambda : \mathbb{R}^3 \times \mathbb{R}^{M\times 3} \times \mathbb{R}^{M\times 3} \to \mathbb{R}^3$

$$\begin{aligned} \lambda(\boldsymbol{r}|\mathbf{R}', \mathbf{R}) &= \boldsymbol{r} + (\boldsymbol{R}_i' - \boldsymbol{R}_i), \\ i &= \arg\min_m \|\boldsymbol{R}_m - \boldsymbol{r}\|_2. \end{aligned} \tag{15}$$

LIMITATIONS

Firstly, while PlaNet enables high-resolution multi-dimensional energy surfaces, scaling to higher-dimensional energy surfaces remains a challenge for all computational methods. Due to the curse of dimensionality, high dimensional energy surfaces will only be sparsely sampled, e.g., in Appendix G.5 we find PlaNet exceeding the threshold of chemical accuracy once modeling the full six-dimensional energy surface of H_2-HF. To lessen this issue, one could look into systematically reusing previous labels, extending the training time for higher dimensional energy surfaces, or adopting equivariant surrogate architectures as these have shown to be more sample efficient (Batzner et al., 2022).

Secondly, DimeNet^{++} cannot distinguish some molecules. As this work seeks to present the general PlaNet framework, the choice of surrogate architecture is not an essential part of this work. We encourage the adoption of universal models in later works (Thomas et al., 2018; Gasteiger et al., 2021).

Table 1: MAE↓ in $\mathrm{m}E_\mathrm{h}$ of VMC energy surfaces compared to experimental results on N_2 (Le Roy et al., 2006). PESNet++ reduces the error by $74\,\%$.

	MAE ↓
PESNet	5.36
+ SiLU	4.90
+ zero bias	4.23
+ Jastrow	3.29
+ Dense orbitals	2.60
+ Restricted	1.39

Table 2: MAE↓ in $\mathrm{m}E_\mathrm{h}$ between VMC and PlaNet over several energy surfaces. SE indicates the standard error of VMC energies in $\mathrm{m}E_\mathrm{h}$. Brackets indicate the standard deviation at the last digit(s) across 5 trainings.

	MAE ↓	SE
H_4	0.0089(5)	0.004
Li_2	0.051(11)	0.019
H_{10}	0.152(25)	0.018
N_2	0.26(5)	0.186
H_2-HF	0.324(14)	0.14
C_2H_5OH	0.298(26)	0.33

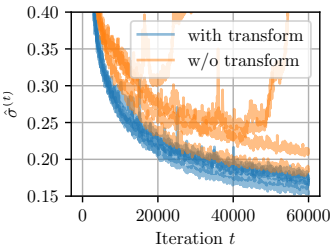


Figure 2: Energy standard deviation $\hat{\sigma}^{(t)}\downarrow$ during optimization with and without coordinate transform. For each setting 5 runs are plotted.

Lastly, the restricted neural wave function formulation is only suitable for molecules where the ground state is a singlet state. For molecules with multiple unpaired electrons, for instance the O_2 ground-state, or atomic systems, restricted neural wave functions cannot properly describe the ground state and, thus, lead to higher energies.

6 Experiments

We split our experiments into two parts. Firstly, we present exhaustive ablation studies on the contributions from Sections 4 and 5. Secondly, we analyze PlaNet's scaling by applying it to several potential energy surfaces, including the challenging nitrogen dimer, a two-dimensional energy surface, and a high-resolution energy surface of the larger ethanol molecule.

We denote the sampling-free inference energy estimates as PlaNet and the VMC energies by PESNet++. Note that VMC energies are guaranteed to be upper bounds to the true energy, while no such a bound exists for the PlaNet energies. When evaluating PlaNet energies, PESNet++ should be seen target since PlaNet should surrogate the VMC integration. Appendix D details the setup, and Appendix C the optimization. Timings are given in Nvidia A100 GPU hours.

6.1 Ablation studies

PESNet++. Firstly, we evaluate the proposed architectural changes of PESNet++ (Section 5) on the nitrogen molecule in Table 1. We picked the nitrogen molecule as it is known for strong electron correlation effects that result in relatively high errors for most computational methods (Gdanitz, 1998; Pfau et al., 2020; Gao & Günnemann, 2022). Evidently, all improvements gradually lower the MAE to the experimental results leading to a total error reduction of $74\,\%$. Interestingly, restricting the wave function to doubly occupied orbitals results in the largest improvements, both in percentage ($46\,\%$) and absolute value ($1.21\,\mathrm{m}E_\mathrm{h}$). See Appendix H for a training dynamics comparison.

Coordinate transform. To identify whether the in Section 5 proposed coordinate transform benefits training on multi-dimensional energy surfaces, we train multiple PESNet++ with and without the coordinate transform, on the full six-dimensional energy surface of H_2-HF and observe the convergence of the standard deviation of the energy $\hat{\sigma}^{(t)}$. As the wave functions converge we expect $\hat{\sigma}^{(t)}$ to approach 0. Figure 2 shows the convergence of several PESNet++ with and without the proposed coordinate transform. We find our transformation aiding in consistency and quality of convergence. Noticeably, without our coordinate transforms 3 out of 5 runs diverge during training. Compared to the converged runs, final standard deviations are on average $17\,\%$ lower, reducing the number of samples to obtain the same statistical error by $31\,\%$.

PlaNet energies. Lastly, we analyze PlaNet's reconstruction of VMC energies across a variety of energy surfaces, see Appendix J for the list of used geometries. The low errors in Table 2 suggest a very close fit between the PlaNet and VMC energies as all deltas are significantly below the threshold for chemical accuracy of $1.6\,\mathrm{m}E_\mathrm{h}$. Further, for larger systems, the fit is indistinguishable from

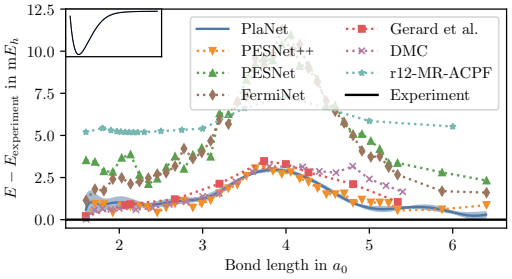


Figure 3: Potential energy curve of N_2 (Pfau et al., 2020; Gao & Günnemann, 2022; Le Roy et al., 2006; Gerard et al., 2022; Ren et al., 2022).

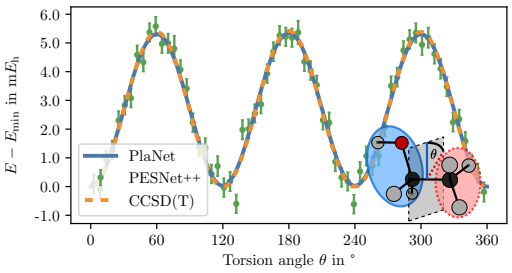


Figure 4: Potential energy curve of the CH_3 torsion angle of Trans-ethanol (Nandi et al., 2022). Error bars indicate the standard error.

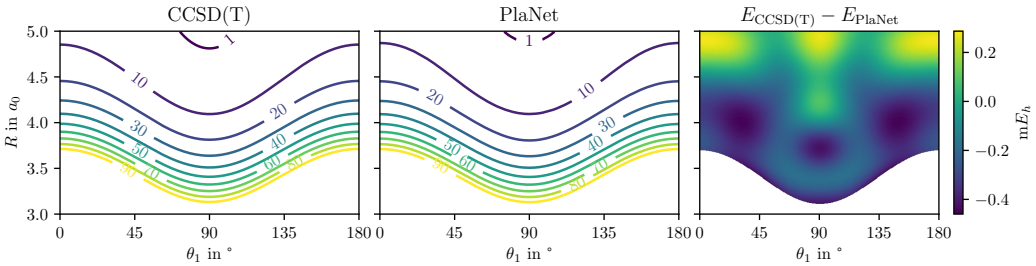


Figure 5: Two-dimensional slice of the potential energy surface of H_2-HF. The remaining four dimensions are fixed at the equilibrium structure. Energies are zeroed at the equilibrium structure and in $\mathrm{m}E_\mathrm{h}$. PlaNet agrees well ($\approx 0.28(9)\,\mathrm{m}E_\mathrm{h}$) with the CCSD(T)-based fit (Yang et al., 2018).

the 'ground-truth' VMC energies as the VMC error per geometry increases. Appendix I presents additional ablation studies on PlaNet hyperparameters and Appendix E an analysis of relative errors.

6.2 Potential Energy Surfaces

Evaluating *ab-initio* methods remains a complicated issue as true energy values are rarely known. Furthermore, the threshold for chemical accuracy at $1\,\mathrm{kcal\,mol^{-1}} \approx 1.6\,\mathrm{m}E_\mathrm{h}$ is often relatively small compared to the absolute magnitude of energies. However, as energy differences in chemistry are much more important than their absolute value, we follow the standard procedure (Pfau et al., 2020; Hermann et al., 2020; Gerard et al., 2022; Gao & Günnemann, 2022) and compare relative energy surfaces. As references, we use CCSD(T) or experimental results. Additional experiments on the lithium dimer (G.1), the hydrogen chain (G.2), cyclobutadiene (G.3), different ethanol states (G.4), and higher-dimensional energy surfaces of H_2-HF (G.5) are available in Appendix G.

Heavily correlated systems. The nitrogen dimer is known to be a challenging system due to significant electron correlation effects (Le Roy et al., 2006), causing even the 'gold-standard' CCSD(T) method to fail (Lyakh et al., 2012). Nonetheless, FermiNet presented very accurate results comparable to the factorially scaling r12-MR-ACPF method (Pfau et al., 2020; Gdanitz, 1998). The comparison in Figure 3 shows that PESNet++ estimates the lowest variational energies. As for the relative error $\Delta E_\mathrm{max} - \Delta E_\mathrm{min}$, PESNet++ outperforms Gerard et al. (2022)'s FermiNet extension and expensive DMC methods with a relative error of $3\,\mathrm{m}E_\mathrm{h}$ compared to $3.2\,\mathrm{m}E_\mathrm{h}$ for DMC (Ren et al., 2022) and Gerard et al. (2022), despite training only a single model for a fraction of the time. Finally, PlaNet's equilibrium distance of $2.0747\,\mathrm{a}_0$ is very close to the experimental results at $2.0743\,\mathrm{a}_0$ (Le Roy et al., 2006). For details on finding the equilibrium geometries see Appendix K.

Scaling to larger systems. To explore the generalization towards larger systems, we investigate the potential energy surface of Trans-ethanol along the CH3 torsion angle. In Figure 4, we compare PESNet++ and PlaNet estimates with accurate CCSD(T) results from Nandi et al. (2022). PlaNet agrees with the 99% confidence interval of all PESNet++ energies. Further, PlaNet almost perfectly

recovers the CCSD(T) results with a maximum disagreement of $0.13\,\mathrm{m}E_\mathrm{h}$ while only requiring $\approx 0.11\,\mathrm{ms}$ per inference after training compared to the $\approx 2.7\,\mathrm{h}$ per inference required for VMC integration. Additionally, the PlaNet energies are more robust and less noisy than the VMC energies obtained via numerical integration. While this discrepancy can be closed by increasing the number of samples during Monte-Carlo integration, reducing the VMC error bars by an order magnitude requires $\approx 270\,\mathrm{h}$ per geometry. This consistency is an encouraging sign that PlaNet-based interpolation has significant advantages in multi-geometry VMC over traditional post hoc interpolation methods (Nandi et al., 2022; Yang et al., 2018; Jiang & Guo, 2013). In Appendix G.4, we explore the same CH3 torsion angle energy surfaces for different ethanol states.

Two-dimensional energy surface. Finally, we are interested in PlaNet's scaling to multi-dimensional energy surfaces. We chose a two-dimensional slice of the H_2-HF interaction energy surface with CCSD(T) reference calculations (Yang et al., 2018) as described in Appendix F. Figure 5 depicts a comparison between the two energy surfaces. PlaNet accurately reproduces this high-resolution two-dimensional energy surface with an MAE of $\approx 0.28(9)\,\mathrm{m}E_\mathrm{h}$, leading to a T-shaped equilibrium structure at $R = 5.34\,\mathrm{a}_0$ and $\theta_1 = 90°$ consistent with previous results (Bernholdt et al., 1986; Yang et al., 2018; Guillon et al., 2008). Note that such a high-resolution energy surface would not have been obtainable with previous neural wave function methods, see Appendix 6.3. In Appendix G.5, we analyze the fitting of PlaNet to higher-dimensional slices of the energy surface.

6.3 Training and inference times

In the previous experiments, we have shown that PlaNet approximates the underlying Monte Carlo integration well with little to no additional errors. In this section, we put the training and inference times of PlaNet's surrogate in perspective to the standard VMC procedure. In Table 3, we list the required times for training PESNet++ (VMC training), performing Monte Carlo integration (VMC inference), training the surrogate, and performing surrogate inference. While the training times refer to training the model to a whole energy surface, the inference times are given per geometry. Note that we exclude pretraining and initial MCMC thermalization in training since these contribute only a small fraction of the total time. Evidently, training the surrogate is equivalent to between <1 and 3 VMC inferences worth of time, while enabling 6 to 9 orders of magnitude faster inference. Further, surrogate training accounts for $<1\,\%$ of the total training time. These time advantages are growing with system size, e.g., for ethanol the surrogate training accounts for $<0.2\,\%$ of the total training time.

Table 3: VMC and surrogate training and inference times per system. Inferences times are given per geometry. The training of the surrogate accounts for $<1\,\%$ of total training times while accelerating inference by 6 to 9 orders of magnitude.

	VMC		Surrogate	
	Training	Inference	Training	Inference
H_4	32 h	9.4 min	20 min	4.7 µs
Li_2	40 h	13.6 min	18.5 min	0.75 µs
H_{10}	81 h	26.8 min	47 min	188 µs
H_2-HF	107 h	38.4 min	20 min	4.7 µs
N_2	123 h	52.7 min	18.5 min	0.75 µs
C_2H_5OH	478 h	2.7 h	43 min	118 µs

7 Conclusion

We presented the PlaNet framework for sampling-free inference of potential energy surface networks by training an additional surrogate on intermediate noisy variables from the VMC optimization. We addressed the changing energies during training by viewing the problem as an online learning task. Moreover, we identified two training regimes depending on PlaNet's accuracy and energy distribution. These regimes enabled us to perform efficient temporal data aggregation, reducing the impact of noisy labels. In addition, we explored doubly-occupied orbitals for neural wave functions by introducing PESNet++. In our evaluation, we observed that: (1) PESNet++ results in new state-of-the-art VMC results on several molecules, and (2) PlaNet suffers no loss in accuracy while decreasing inference times from hours to milliseconds. Thanks to these improvements in inference and accuracy, accurate high-resolution multi-dimensional energy surfaces with neural wave functions are obtainable. These are promising indicators for the future application of neural wave function methods for tasks that typically require hundreds of thousands or millions of energy evaluations, like analyzing multi-dimensional energy surfaces or molecular dynamics simulations.

ACKNOWLEDGEMENTS

We thank Daniel Zügner, Filippo Guerranti, and Jan Schuchardt for their invaluable feedback on the manuscript, and Marten Lienen for our discussion on visualizing torsion angles.

Funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

REPRODUCIBILITY

Our source is publicly available [1] licensed under the Hippocratic license (Ehmke, 2019). Further, Appendix A details the PESNet++ architecture, Appendix C the optimization algorithm, and Appendix D the rest of the experimental setup including all hyperparameters.

REFERENCES

Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, May 2013. ISSN 1098-0121, 1550-235X. doi: 10.1103/PhysRevB.87.184115.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5.

Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, February 2011. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.3553717.

David E. Bernholdt, Shi-yi Liu, and Clifford E. Dykstra. A theoretical study of the structure, bonding, and vibrational frequency shifts of the H_2–HF complex. *The Journal of Chemical Physics*, 85(9):5120–5127, November 1986. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.451705.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018.

Andrew Brock, Soham De, and Samuel L. Smith. Characterizing signal propagation to close the performance gap in unnormalized ResNets. In *International Conference on Learning Representations*, February 2022.

Andy Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-Performance Large-Scale Image Recognition Without Normalization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 1059–1071. PMLR, July 2021.

M. D. Brown, J. R. Trail, P. López Ríos, and R. J. Needs. Energies of the first row atoms from quantum Monte Carlo. *The Journal of Chemical Physics*, 126(22):224110, June 2007. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2743972.

Giuseppe Carleo and Matthias Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science*, 355(6325):602–606, February 2017. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aag2302.

G. Cassella, H. Sutterud, S. Azadi, N. D. Drummond, D. Pfau, J. S. Spencer, and W. M. C. Foulkes. Discovering Quantum Phase Transitions with Fermionic Neural Networks. *arXiv:2202.05183 [cond-mat, physics:physics]*, February 2022.

D. Ceperley, G. V. Chester, and M. H. Kalos. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, October 1977. doi: 10.1103/PhysRevB.16.3081.

[1] https://www.cs.cit.tum.de/daml/pesnet/

Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature Communications*, 11(1):2368, December 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15724-9.

Anders S. Christensen, Lars A. Bratholm, Felix A. Faber, and O. Anatole von Lilienfeld. FCHL revisited: Faster and more accurate quantum machine learning. *The Journal of chemical physics*, 152(4):044107, 2020.

Coraline Ada Ehmke. The Hippocratic License 2.1: An Ethical License for Open Source. https://firstdonoharm.dev, 2019.

R. P. Feynman and Michael Cohen. Energy Spectrum of the Excitations in Liquid Helium. *Physical Review*, 102(5):1189–1204, June 1956. ISSN 0031-899X. doi: 10.1103/PhysRev.102.1189.

Nicholas Gao and Stephan Günnemann. Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions. In *International Conference on Learning Representations*, April 2022.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, September 2019.

Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. In *NeurIPS-W*, December 2020.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. GemNet: Universal Directional Graph Neural Networks for Molecules. In *Advances in Neural Information Processing Systems*, May 2021.

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. *Chemical Physics Letters*, 283(5):253–261, February 1998. ISSN 0009-2614. doi: 10.1016/S0009-2614(97)01392-4.

Leon Gerard, Michael Scherbela, Philipp Marquetand, and Philipp Grohs. Gold-standard solutions to the Schrödinger equation using deep learning: How much physics do we need? *Advances in Neural Information Processing Systems*, May 2022.

G. Guillon, T. Stoecklin, A. Voronin, and Ph. Halvick. Rotational relaxation of HF by collision with ortho- and para-H2 molecules. *The Journal of Chemical Physics*, 129(10):104308, September 2008. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2975194.

Thomas A. Halgren. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of Computational Chemistry*, 17(5-6):490–519, 1996. ISSN 1096-987X. doi: 10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.

Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, December 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.108929.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv:1606.08415 [cs]*, July 2020.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y.

Johannes Hoja, Leonardo Medrano Sandonas, Brian G. Ernst, Alvaro Vazquez-Mayagoitia, Robert A. DiStasio Jr., and Alexandre Tkatchenko. QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific Data*, 8:43, February 2021. ISSN 2052-4463. doi: 10.1038/s41597-021-00812-2.

Robert Jastrow. Many-body problem with strong forces. *Physical Review*, 98(5):1479, 1955.

Jan H. Jensen. *Molecular Modeling Basics*. CRC Press, 2010.

Bin Jiang and Hua Guo. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. *The Journal of Chemical Physics*, 139(5):054112, August 2013. ISSN 0021-9606. doi: 10.1063/1.4817187.

Jan Kessler, Francesco Calcavecchia, and Thomas D. Kühne. Artificial Neural Networks as Trial Wave Functions for Quantum Monte Carlo. *Advanced Theory and Simulations*, 4(4):2000269, 2021. ISSN 2513-0390. doi: 10.1002/adts.202000269.

James Kirkpatrick, Brendan McMorrow, David H. P. Turban, Alexander L. Gaunt, James S. Spencer, Alexander G. D. G. Matthews, Annette Obika, Louis Thiry, Meire Fortunato, David Pfau, Lara Román Castellanos, Stig Petersen, Alexander W. R. Nelson, Pushmeet Kohli, Paula Mori-Sánchez, Demis Hassabis, and Aron J. Cohen. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374(6573):1385–1389, December 2021. doi: 10.1126/science.abj6511.

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. *The Journal of Chemical Physics*, 125(16):164310, October 2006. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2354502.

Jun Li, Bin Jiang, and Hua Guo. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. II. Four-atom systems. *The Journal of Chemical Physics*, 139(20): 204103, November 2013. ISSN 0021-9606. doi: 10.1063/1.4832697.

Xiang Li, Cunwei Fan, Weiluo Ren, and Ji Chen. Fermionic neural network with effective core potential. *Physical Review Research*, 4(1):013021, January 2022a. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.4.013021.

Xiang Li, Zhe Li, and Ji Chen. Ab initio calculation of real solids via neural network ansatz. *arXiv:2203.15472 [cond-mat, physics:physics]*, March 2022b.

Jeffmin Lin, Gil Goldshlager, and Lin Lin. Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation. *arXiv:2112.03491 [physics]*, December 2021.

Dmitry I. Lyakh, Monika Musiał, Victor F. Lotrich, and Rodney J. Bartlett. Multireference Nature of Chemistry: The Coupled-Cluster View. *Chemical Reviews*, 112(1):182–243, January 2012. ISSN 0009-2665, 1520-6890. doi: 10.1021/cr2001417.

Angelo M. Maniero and Paulo H. Acioli. Full configuration interaction pseudopotential determination of the ground-state potential energy curves of Li2 and LiH. *International Journal of Quantum Chemistry*, 103(5):711–717, 2005. ISSN 1097-461X. doi: 10.1002/qua.20593.

W. L. McMillan. Ground State of Liquid He4. *Physical Review*, 138(2A):A442–A451, April 1965. doi: 10.1103/PhysRev.138.A442.

Mario Motta, David M. Ceperley, Garnet Kin-Lic Chan, John A. Gomez, Emanuel Gull, Sheng Guo, Carlos A. Jiménez-Hoyos, Tran Nguyen Lan, Jia Li, Fengjie Ma, Andrew J. Millis, Nikolay V. Prokof'ev, Ushnish Ray, Gustavo E. Scuseria, Sandro Sorella, Edwin M. Stoudenmire, Qiming Sun, Igor S. Tupitsyn, Steven R. White, Dominika Zgid, Shiwei Zhang, and Simons Collaboration on the Many-Electron Problem. 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. *Physical Review X*, 7(3):031059, September 2017. ISSN 2160-3308. doi: 10.1103/PhysRevX.7.031059.

Apurba Nandi, Riccardo Conte, Chen Qu, Paul L. Houston, Qi Yu, and Joel M. Bowman. Quantum Calculations on a New CCSD(T) Machine-Learned Potential Energy Surface Reveal the Leaky Nature of Gas-Phase *Trans* and *Gauche* Ethanol Conformers. *Journal of Chemical Theory and Computation*, pp. acs.jctc.2c00760, August 2022. ISSN 1549-9618, 1549-9626. doi: 10.1021/acs.jctc.2c00760.

Eric Neuscamman, C. J. Umrigar, and Garnet Kin-Lic Chan. Optimizing large parameter sets in variational quantum Monte Carlo. *Physical Review B*, 85(4):045103, January 2012. ISSN 1098-0121, 1550-235X. doi: 10.1103/PhysRevB.85.045103.

David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, September 2020. doi: 10.1103/PhysRevResearch.2.033429.

Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller III. OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital Features. *The Journal of Chemical Physics*, 153(12):124111, September 2020. ISSN 0021-9606, 1089-7690. doi: 10.1063/5.0021955.

Zhuoran Qiao, Anders S. Christensen, Frederick R. Manby, Matthew Welborn, Anima Anandkumar, and Thomas F. Miller III. UNiTE: Unitary N-body Tensor Equivariant Network with Applications to Quantum Chemistry. *arXiv:2105.14655 [physics]*, May 2021.

Weiluo Ren, Weizhong Fu, and Ji Chen. Towards the ground state of molecules via diffusion Monte Carlo on neural networks. *arXiv:2204.13903 [physics]*, April 2022.

Michael Scherbela, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs. Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks. *Nature Computational Science*, 2(5):331–341, May 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00228-x.

K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24): 241722, June 2018. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.5019779.

Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9377–9388. PMLR, July 2021.

J. C. Slater. The Theory of Complex Spectra. *Physical Review*, 34(10):1293–1322, November 1929. doi: 10.1103/PhysRev.34.1293.

John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding Density Functionals with Machine Learning. *Physical Review Letters*, 108(25):253002, June 2012. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.108.253002.

James S. Spencer, David Pfau, Aleksandar Botev, and W. M. C. Foulkes. Better, Faster Fermionic Neural Networks. *3rd NeurIPS Workshop on Machine Learning and Physical Science*, November 2020.

Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, 2012.

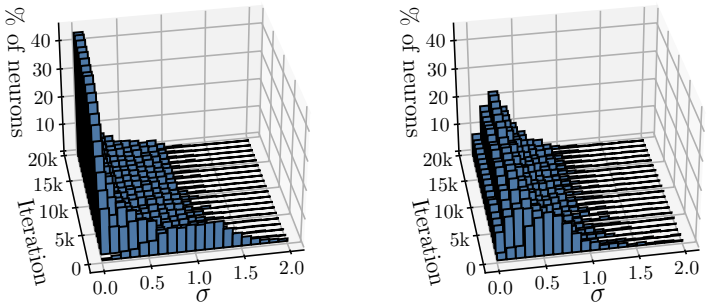
Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv:1802.08219 [cs]*, May 2018.

Simon Wengert, Gábor Csányi, Karsten Reuter, and Johannes T. Margraf. Data-efficient machine learning for molecular crystal structure prediction. *Chemical Science*, pp. 10.1039.D0SC05765G, 2021. ISSN 2041-6520, 2041-6539. doi: 10.1039/D0SC05765G.

Max Wilson, Nicholas Gao, Filip Wudarski, Eleanor Rieffel, and Norm M. Tubman. Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo. *arXiv:2103.12570 [physics, physics:quant-ph]*, March 2021.

Max Wilson, Saverio Moroni, Markus Holzmann, Nicholas Gao, Filip Wudarski, Tejs Vegge, and Arghya Bhowmik. Wave function Ansatz (but Periodic) Networks and the Homogeneous Electron Gas. *arXiv:2202.04622 [cond-mat, physics:physics, physics:quant-ph]*, February 2022.

Dongzheng Yang, Jing Huang, Junxiang Zuo, Xixi Hu, and Daiqian Xie. A full-dimensional potential energy surface and quantum dynamics of inelastic collision process for H2–HF. *The Journal of Chemical Physics*, 148(18):184301, May 2018. ISSN 0021-9606. doi: 10.1063/1.5030384.



(a) tanh + normal bias (b) SiLU + zero bias

Figure 6: Distribution of standard deviation of neurons during training.

A PESNET++ ARCHITECTURE

PESNet++ consists of three key ingredients, the MetaGNN, the equivariant coordinate system, and the wave function model (WFModel). The WFModel is used within the VMC framework to generate gradients and find the ground-state wave function. The MetaGNN's goal is to adapt the WFModel to the geometry at hand and the equivariant coordinate system enforces the physical symmetries of the energy. We directly adopt the MetaGNN and the equivariant coordinate system from Gao & Günnemann (2022) and would like to redirect the reader to the original work for more information. All architectural improvements in PESNet++ affect the WFModel. The WFModel acting on an electron configuration $\mathbf{r}$ first constructs single-electron $\boldsymbol{h}_i^{(1)}$ and pair-wise $\boldsymbol{g}_{ij}^{(1)}$ features in a permutation equivariant way. We adopt the same construction as in (Gao & Günnemann, 2022):

$$\boldsymbol{h}_1^{(1)} = \sum_{m=1}^{M} \text{MLP}\left(\boldsymbol{W}\left[\left(\boldsymbol{r}_i - \boldsymbol{R}_i\right) E, \|\boldsymbol{r}_i - \boldsymbol{R}_i\|\right] + z_m\right), \tag{16}$$

$$\boldsymbol{g}_{ij}^{(1)} = \left(\left(\boldsymbol{r}_i - \boldsymbol{r}_j\right) E, \|\boldsymbol{r}_i - \boldsymbol{r}_j\|\right) \tag{17}$$

where $E \in \mathbb{R}^{3\times 3}$ is our equivariant coordinate system and z_m are nuclei embeddings outputted by the MetaGNN. These features are then iteratively updated with our new update rules from Equation (10) and Equation (11). After L_{WF} many layers, we use the final electron embeddings $h_i^{(L_{\text{WF}})}$ to construct the orbital matrices ϕ with Equation (13). Finally, we compute the final amplitude with the Jastrow factor with Equation 14.

B DEAD NEURON ANALYSIS

To illustrate the impact of $\tanh$ as an activation function and the normally distributed biases, Figure 6a plots the standard deviation of PESNet's neurons throughout training. One may see that the number of dead neurons increases during training to $>40\,\%$, reducing the effective network size. Our improvements reduce the number of dead neurons to $<10\,\%$ as seen in Figure 6b. Note that we still keep one $\tanh$ activation function in the embedding block to limit the magnitude of our embeddings if distances increase.

C OPTIMIZATION

Each PlaNet optimization step consists of two parts, the first is a VMC optimization step with the additional coordinate transform, and the second is the optimization of the surrogate.

In each VMC step, we first perturb the geometry from the previous iteration followed by the coordinate transformation from Equation (15). Next, we sample for each geometry c new electron positions $\boldsymbol{r}_c$ via Metropolis-Hastings and calculate the local energy $E_{c,i}$ for each electron configuration $\mathbf{r}_{c,i}$ (McMillan, 1965). Given these energies, we compute the gradients and construct our updates via natural gradient descent with the conjugate gradient (CG)-method (Neuscamman et al., 2012). For

Algorithm 1 $t+1$th optimization step

Input: $\mathbf{R}^{(t)}, \mathbf{r}^{(t)}, \Theta^{(t)}, \chi^{(t)}$
Output: $\mathbf{R}, \mathbf{r}, \Theta, \chi$
$\mathbf{R} \sim \rho(\mathbf{R}|\mathbf{R}^{(t)})$
$\mathbf{r}' = \lambda(\mathbf{r}^{(t)}|\mathbf{R}, \mathbf{R}^{(t)})$ ▷ Equation (15)
for $c \in \{1, \dots, C\}$ **do**
  $\boldsymbol{r}_c \sim \psi^2_{\theta_c^{(t)}}(\boldsymbol{r}'_c)$ ▷ MCMC
  $E_{c,i} = \psi_{\theta_c^{(t)}}(\mathbf{r}_{c,i})^{-1}\mathbf{H}\psi_{\theta_c^{(t)}}(\mathbf{r}_{c,i})$ ▷ (Pfau et al., 2020; Hermann et al., 2020)
  $\delta_{c,i} = E_{c,i} - (\frac{1}{B}\sum_{i=1}^{B} E_{c,i})$
end for
$\nabla_{\Theta^{(t)}}\mathcal{L} = \mathbb{E}_{c,i}\left[\delta_{c,i}\nabla_{\Theta^{(t)}}\log|\psi_{\Theta^{(t)}}(\mathbf{r}_{c,i})|\right]$
$\Theta = \Theta^{(t)} - \eta \boldsymbol{F}^{-1}\nabla_{\Theta^{(t)}}\mathcal{L}$ ▷ CG-method (Gao & Günnemann, 2022)

for $i \in \{1, \dots, N_{\text{surr}}\}$ **do**
  $\chi' \leftarrow$ AdamW step on $\mathcal{L}_{\text{surr}}$ ▷ Equation (6)
end for
Compute γ ▷ Equation (7)
$\chi = \gamma\chi^{(t)} + (1-\gamma)\chi'$

further details on the VMC step, we would like to refer the reader to Pfau et al. (2020) and Gao & Günnemann (2022). Next, we fit our surrogate model over N_{surr} many steps to the local energies and, finally, temporally average the surrogate parameters via the moving average described in Section 4. The complete optimization algorithm is given in Algorithm 1. Note that we dropped the dependence on the $t+1$ step and explicitly wrote down the loop over all geometries. Though, in practice one implements those as an additional batch dimension and performs these operations in parallel. To reduce the dependence on the current batch, we smooth $\mathcal{L}^{(t)}_{\text{surr}}$ and $D^{(t)}$ when evaluating Equation (7) with EMAs.

D Experimental setup

All default hyperparameters are reported in Table 4. To avoid exhaustive hyperparameter searches, we use the default hyperparameters for PESNet (Gao & Günnemann, 2022) and DimeNet^{++} (Gasteiger et al., 2020). Exceptions are the nitrogen dimer for which we increased the number of determinants to 32, the H_2-HF energy surfaces for which we increased the determinants to 32, and the number of geometry random walkers to 64. In contrast to PESNet (Gao & Günnemann, 2022), we do not deploy any early stopping criteria for the CG-method as we found the convergence to benefit slightly if we always do the full 100 steps. Note that this has barely any impact on the optimization time as the early stopping criteria were rarely met.

We implemented PlaNet and PESNet++ on top of the official JAX (Bradbury et al., 2018) implementation of PESNet (Gao & Günnemann, 2022) which is distributed under the Hippocratic license (Ehmke, 2019). We ran all experiments on a machine with 16 AMD EPYC 7742 cores and a single Nvidia A100 GPU.

E Relative PlaNet errors

While we analyzed the quality of the potential fit in the main body, the MAE is not a perfect metric, as one is often interested in relative energies rather than their absolute value. To account for this, we present extended results in Table 5 where we also list the relative MAE and the MAD at the last iteration. We define the relative MAE as

$$\text{rel. MAE} = \frac{1}{N}\sum_{i=1}^{N}|(E'_i - \hat{E}') - (E_i - \hat{E}_i)| \tag{18}$$

Table 4: Default hyperparameters.

	Parameter	Value
Optimization	Local energy clipping	5.0
Optimization	Batch size	4096
Optimization	Iterations	60000
Optimization	#Geometry random walker	16
Optimization	Learning rate η	$\frac{0.1}{(1+t/1000)}$
Natural Gradient	Damping	$10^{-4}\sigma^{(t)}$
Natural Gradient	CG steps	100
WFModel	Nuclei embedding dim	64
WFModel	Single-stream width	256
WFModel	Double-stream width	32
WFModel	#Update layers	4
WFModel	#Determinants	16
WFModel	#Jastrow layers	3
MetaGNN	#Message passings	2
MetaGNN	Embedding dim	64
MetaGNN	Message dim	32
MetaGNN	N_{SBF}	7
MetaGNN	N_{RBF}	6
MetaGNN	MLP depth	2
MCMC	Init proposal step size	0.02
MCMC	Steps between updates	40
Pretraining	Iterations	2000
Pretraining	Learning rate	0.003
Pretraining	Method	RHF
Pretraining	Basis set	STO-6G
Evaluation	#Samples	10^6
Evaluation	MCMC Steps	400
DimeNet^{++}	Cutoff r_c	$10.0a_0$
DimeNet^{++}	Basis embedding size	8
DimeNet^{++}	Interaction embedding size	64
DimeNet^{++}	Out embedding size	256
DimeNet^{++}	#Layer after skip	2
DimeNet^{++}	#Layer before skip	1
DimeNet^{++}	#Blocks	4
DimeNet^{++}	#Layer out	3
DimeNet^{++}	N_{RBF}	6
DimeNet^{++}	N_{SBF}	7
PlaNet Optimization	γ_{base}	0.99
PlaNet Optimization	γ_{high}	0.0099
PlaNet Optimization	Learning rate	$\frac{10^{-4}}{(1+t/10000)}$
PlaNet Optimization	N_{surr}	5
PlaNet Optimization	ζ	1.05
PlaNet Optimization	Optimizer	AdamW
PlaNet Optimization	$\mathcal{L}^{(t)}_{\text{surr}}$ and $D^{(t)}$ EMA decay	0.999

Table 5: Extended version of Table 2. MAE↓ in $\mathrm{m}E_\mathrm{h}$ between VMC and PlaNet absolute and relative energies. 'SE' indicates the standard error of the mean for VMC target energies. The 'MAD' row refers to the MAD of the energy distribution at the end of training. Numbers in brackets indicate the standard deviation at the last digit(s) across 5 trainings.

	H_4^+	H_4	Li_2	H_{10}	N_2	H_2-HF	C_2H_5OH
MAD	0.43	0.3	1.5	1.4	9.0	18.8	15.0
SE	0.004	0.004	0.019	0.018	0.186	0.14	0.33
abs. MAE↓	0.0110(4)	0.0089(5)	0.051(11)	0.152(25)	0.26(5)	0.324(14)	0.298(26)
rel. MAE↓	0.0101(4)	0.00442(22)	0.049(11)	0.13(4)	0.25(32)	0.219(32)	0.274(5)

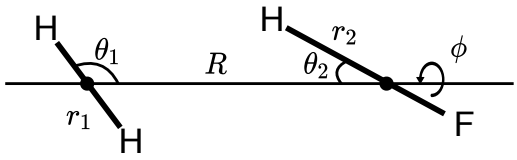


Figure 7: Internal coordinate description of the H_2-HF system.

where E_i and E'_i refer to the VMC and the PlaNet estimate of the energy of the ith configuration, and $\hat{E}$ and $\hat{E}'$ to their respective means.

An interesting result is the comparatively low relative error for systems with heavy nuclei, i.e., non-hydrogen nuclei. We suspect due to the exponential moving average, PlaNet is biased towards higher energies if the VMC energies converge slowly.

F H_2-HF SYSTEM

Figure 7 describes the internal coordinates of the H_2-HF system. We follow (Yang et al., 2018) in the definition of the equilibrium structure and set it to $r_1 = 1.4051$, $r_2 = 1.73496$, $\phi = 0$, $\theta_1 = 90$, $\theta_2 = 180$ and $R = 5.4$. For the two-dimensional slices, we fix all but θ_1 and R at the equilibrium structure.

G ADDITIONAL ENERGY SURFACES

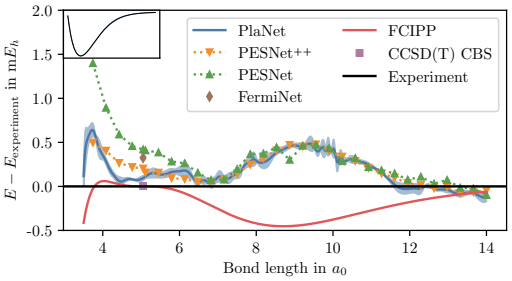


Figure 8: Potential energy curve of Li_2 (Pfau et al., 2020; Maniero & Acioli, 2005). Lightly shaded regions indicate standard deviation across 5 surrogate fits. The inset shows the absolute energy surface. Compared to PESNet, PESNet++ reduces the maximum error by $0.9\,\mathrm{m}E_\mathrm{h}$.

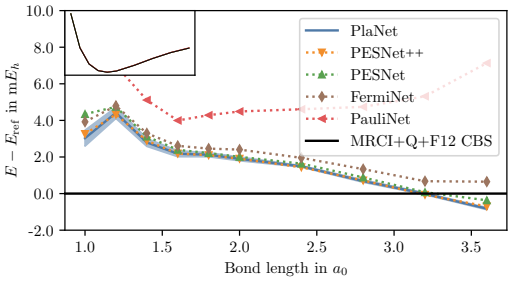


Figure 9: Potential energy curve of the hydrogen chain (Pfau et al., 2020; Hermann et al., 2020; Gao & Günnemann, 2022; Motta et al., 2017). Lightly shaded regions represent the standard deviation across 5 surrogate fits. PESNet++ improves upon PESNet by $0.27\,\mathrm{m}E_\mathrm{h}$.

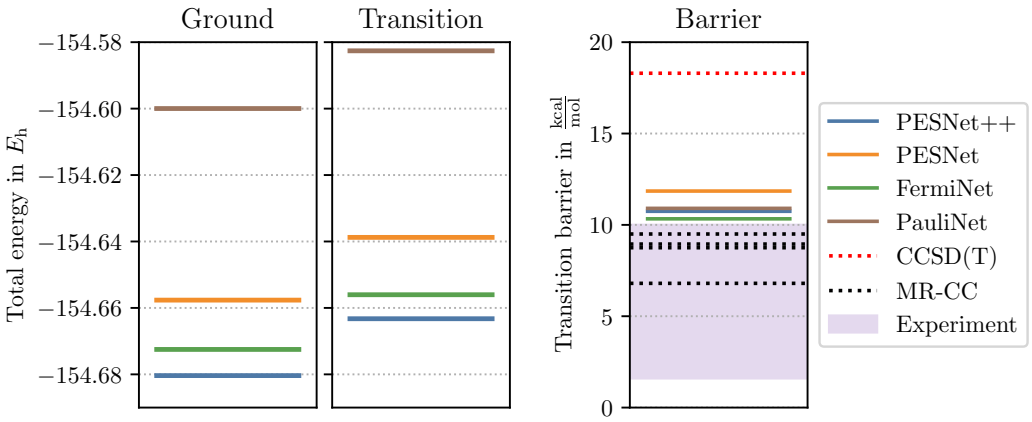


Figure 10: Comparison of PESNet++, PESNet, FermiNet, and PauliNet on estimating the transition barrier of cyclobutadiene. The first two plots show the total energy of the ground and transition state while the right plot shows estimates of the transition barrier. All neural wave function methods estimate similar transition barriers at the upper end of the experimental range. In terms of total energy, PESNet++ outperforms PauliNet by $80.5\,\mathrm{m}E_\mathrm{h}$, PESNet by $23.6\,\mathrm{m}E_\mathrm{h}$ and FermiNet by $7.6\,\mathrm{m}E_\mathrm{h}$.

G.1 Lithium dimer

A typical task in chemistry is identifying equilibrium structures. To quantify PlaNet's suitability for such tasks we apply it to the lithium dimer. One can see in Figure 8 that PlaNet agrees well with experimental results and PESNet++. Further, PlaNet is virtually indistinguishable from the VMC estimates. When searching for the equilibrium structure with PlaNet, we find the minimum at $5.041\,\mathrm{a}_0$, i.e., only $0.01\,\mathrm{a}_0$ apart from the experimental results (Maniero & Acioli, 2005). PESNet++ reduces the maximum error from $1.4\,\mathrm{m}E_\mathrm{h}$ to $0.5\,\mathrm{m}E_\mathrm{h}$ (64 %).

G.2 Hydrogen chain

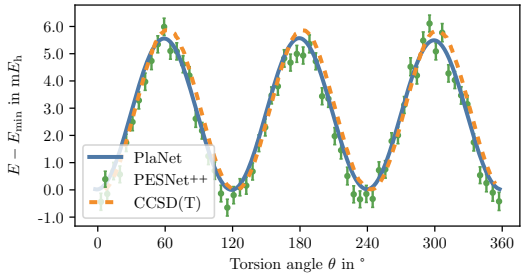
The hydrogen chain is a classic benchmark geometry with a rich history (Motta et al., 2017). Thus, we are interested in how our PlaNet and PESNet++ compare to other neural wave functions (Pfau et al., 2020; Hermann et al., 2020; Gao & Günnemann, 2022). The potential energy curve is plotted in Figure 9. We find our PESNet++ to produce the, to date, lowest energies with an improvement of $\approx 0.3\,\mathrm{m}E_\mathrm{h}$ over PESNet and $\approx 0.6\,\mathrm{m}E_\mathrm{h}$ over FermiNet. Further, the PlaNet energies only differ by $0.12\,\mathrm{m}E_\mathrm{h}$ from the VMC estimates.

G.3 Transition barrier of cyclobutadiene

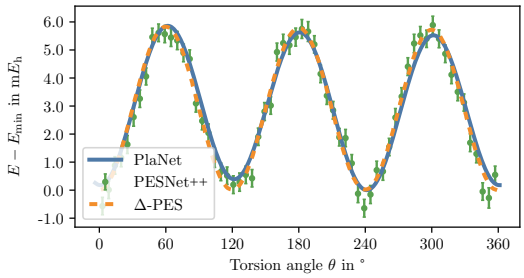
To compare the architectural improvements of PESNet++ to existing neural wave functions, we report total energies and transition barrier estimates in Figure 10. As in FermiNet (Spencer et al., 2020) and PESNet (Gao & Günnemann, 2022), we increased the hidden dimension of the single electron features to 512 and use 32 instead of 16 determinants. We resolved the convergence difficulties reported by Gao & Günnemann (2022) by training on the continuous energy surface obtained by linearly interpolating between the ground and transition state instead of only training on the ground and transition state. From the results in Figure 10, it is apparent that PESNet++ improves variational energies significantly by setting new state-of-the-art variational energies that are on average $7.6\,\mathrm{m}E_\mathrm{h}$ better than FermiNet's. Though, it should be noted that compared to PESNet more compute resources have been used due to the additional cost of the dense orbitals and restricted neural wave function, see Appendix H.

G.4 Different ethanol states

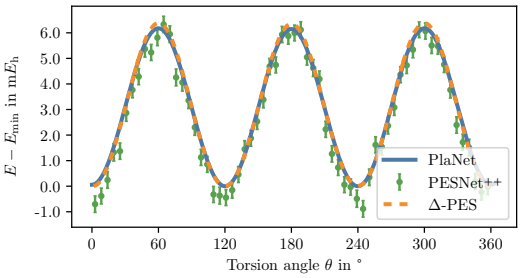
In the main body, we covered the CH_3 torsion angle of Trans-ethanol which is in perfect agreement with the CCSD(T) reference (Nandi et al., 2022) calculations. Here, we additionally explore the



(a) Gauche



(b) TS1



(c) TS2

Figure 11: Potential energy curves of the CH_3 torsion angle of ethanol for different OH torsion angle arrangements.

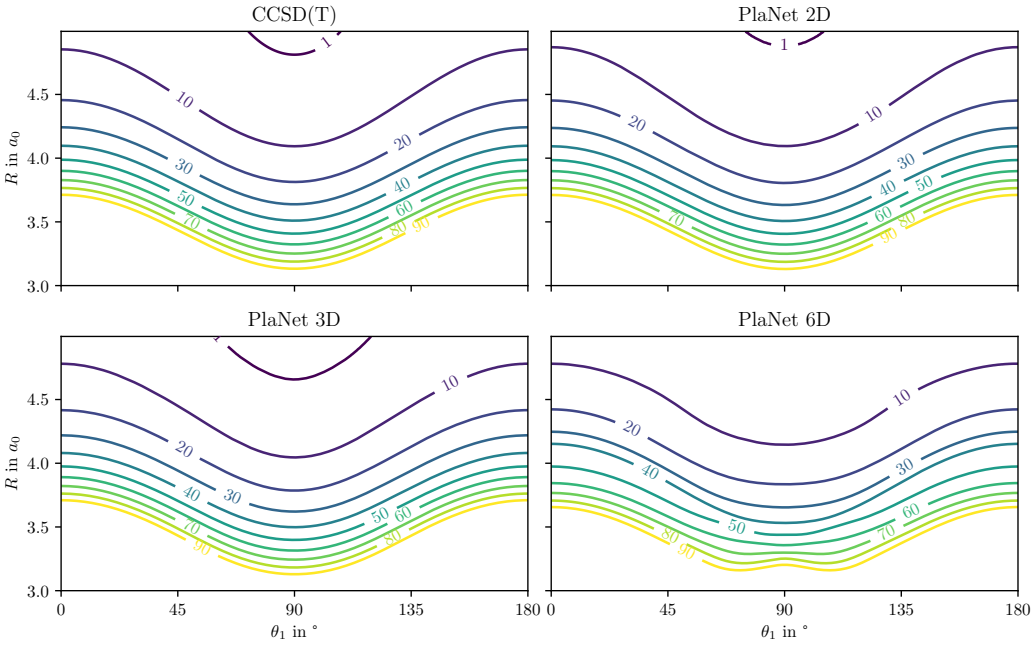


Figure 12: Comparison between different PlaNets trained on a two-dimensional slide (top right), three-dimensional slice (bottom left), and the full six-dimensional energy surface of H_2-HF compared to the reference CCSD(T) energy surface (Yang et al., 2018).

Gauche conformer as well as the two transition states TS1 and TS2. All energy surfaces are plotted in Figure 11. Note that while the reference calculations for Gauche in Figure 11a are CCSD(T) based, for both transition states we use reference data from an Δ-ML approach (Nandi et al., 2022). While all surfaces generally agree on the approximate shape of the energy surface, we notably identify differences between our PlaNet estimates and the reference data for the Gauche and TS1 states. For the Gauche conformer, we locate a slightly different minimum and for TS1 we find the 120° periodicity to be broken in the given geometries. Note that the broken symmetry is due to working on partially relaxed structures (Nandi et al., 2022).

In terms of relative error, we observe a maximum discrepancy of $0.63\,\mathrm{m}E_\mathrm{h}$, $0.55\,\mathrm{m}E_\mathrm{h}$ and $0.21\,\mathrm{m}E_\mathrm{h}$ for Gauche, TS1, and TS2, respectively.

G.5 Higher dimensional energy surfaces

While the main body already contains results on the two-dimensional energy surface of H_2-HF, we are often interested in the full-dimensional energy surface. Though, obtaining sufficient samples from higher dimensional surfaces is a challenge due to the curse of dimensionality. To see how well our PlaNet framework generalizes to more dimensions, we train additional models on a three-dimensional slice and the full six-dimensional energy surface of H_2-HF. For the three-dimensional training, we additionally include the θ_2 angle. Albeit training on additional dimensions, we stick to the comparison of the same two-dimensional slice as in the main body to ensure comparability.

Figure 12 shows a comparison between the CCSD(T) baseline, and the three differently trained PlaNet models. While the general shape is similar between all models, it is noticeable that the additional dimensions cause the energy surface to deform at angles of $\theta_1 \approx 90°$ increasing the MAE from $0.28(9)\,\mathrm{m}E_\mathrm{h}$ to $1.02(34)\,\mathrm{m}E_\mathrm{h}$ and $2.39(25)\,\mathrm{m}E_\mathrm{h}$, respectively. We hypothesize that this is due to the curse of dimensionality and the consequent scarcity of data. Note that while the CCSD(T) baseline required first the selection and evaluation of 35.000 individual data points and an additional function fit, our PlaNet framework is able to approximately capture the energy surface with a single training. Although we do not know the exact time requirements for the CCSD(T) calculations, we expect them to be in the range of a few minutes per geometry on a modern CPU, i.e., if we take an

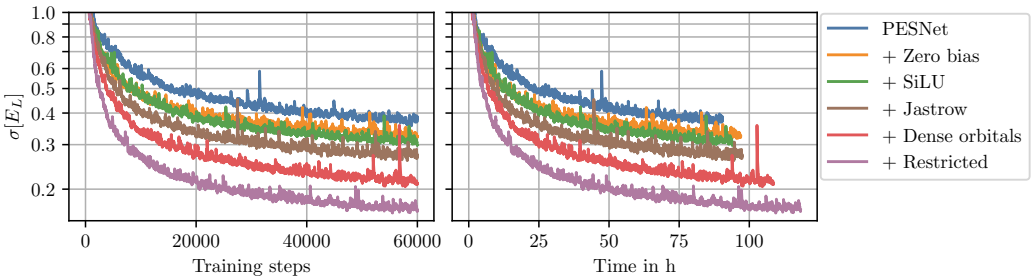


Figure 13: Training plots on the nitrogen dimer with different enhancements to PESNet. On the left, the standard deviation of the local energy (lower is better) is plotted on the y-axis, and the training steps on the x-axis. The right plot shares the same y-axis but uses the training time ax x-axis. It is apparent that all enhancements of PESNet++ strictly enhance the efficiency of PESNet as they provide lower errors in less time and steps.

Table 6: Ablation study on the number of surrogate training steps N_{surr}.

N_{surr}	H_4^+	H_4	Li_2	H_{10}	N_2	H_2-HF
1	0.0124(4)	0.0084(8)	0.0819(13)	0.14(4)	0.57(21)	0.338(31)
5	0.0117(4)	0.0072(7)	0.0786(9)	0.14(7)	0.33(7)	0.333(35)
10	0.0115(4)	0.00551(20)	0.0758(12)	0.19(8)	0.32(6)	0.33(4)

optimistic guess of 1 minute per evaluation, the total compute time for the energy surface is at least 24 days. In contrast, our PlaNet model has been trained in only 4.5 days with a single Nvidia A100 GPU.

H Convergence

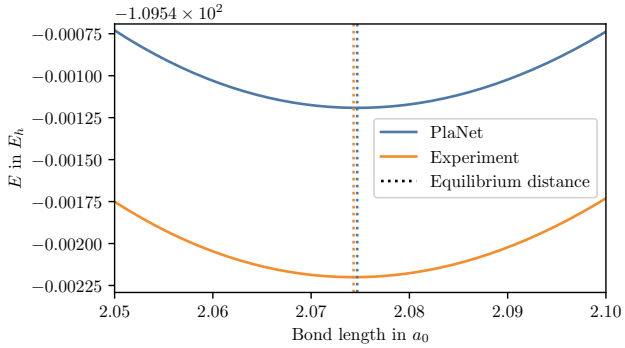
To analyze the efficiency of our proposed enhancements to PESNet, we investigate the convergence rate based on terms of the number of steps as well as time. The resulting training curves for the nitrogen molecule are plotted in Figure 13. it can be seen that all enhancements strictly improve convergence and result in significantly lower errors. While especially the dense orbitals add a non-trivial amount of additional computational requirement, they also result in significantly better energies. The convergence rate per time and per step is strictly improving for all of our proposed improvements.

I Ablation studies

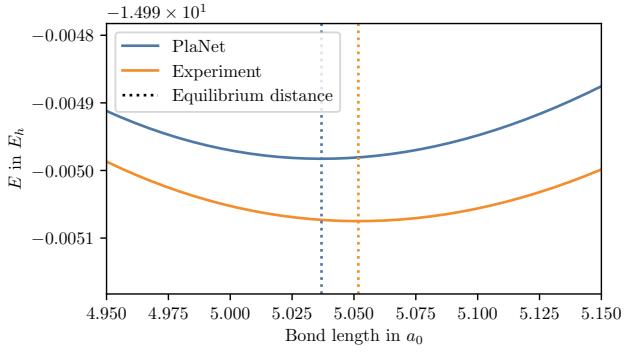
To evaluate the value of our training heuristics, we perform ablation studies on the number of surrogate training steps N_{surr} and the exponential moving average decay rate γ. For the first experiment, we alter the N_{surr} hyperparameter between 1, 5, and 10 and run each experiment 5 times. The results are listed in Table 6. It can be seen that performing more than one step typically improves the results across all systems. This effect is more pronounced for larger systems.

Table 7: Ablation study with fixed values for the decay γ of the exponential moving average.

γ	H_4^+	H_4	Li_2	H_{10}	N_2	H_2-HF
0.99	0.0117(4)	0.0076(6)	0.0786(9)	0.112(21)	0.57(15)	0.39(4)
0.9999	0.019(4)	0.167(30)	0.087(4)	0.55(7)	0.46(14)	0.26(5)
adaptive	0.0117(4)	0.0072(7)	0.0786(9)	0.14(7)	0.33(7)	0.333(35)



(a) Nitrogen dimer



(b) Lithium dimer

Figure 14: High-resolution energy surface scans around the equilibrium structure.

In the second experiment, we fix the exponential moving average parameter γ instead of adapting it. The results in Table 7 show that an initial high decay factor results in significantly higher final errors due to the large influence of the initial bad labels. While a low γ works reasonably across all systems, we are able to obtain better results on larger systems with our adaptive scheduling.

In general, we find our PlaNet framework to be robust against different hyperparameter settings.

J Systems

In the following, we list all geometries used throughout this paper:

- Hydrogen rectangle H_4: geometries taken from Pfau et al. (2020).
- Lithium dimer Li_2: 32 evenly distributed distances between $3.5\,a_0$ and $14\,a_0$.
- Hydrogen chain H_{10}: geometries taken from Motta et al. (2017).
- Nitrogen dimer N_2: geometries taken from Pfau et al. (2020).
- H_2-HF: 64 regular grid points of the N-dimensional energy surface. The boundaries are chosen as: $r_1, r_2 \in [1.2\,a_0, 1.8\,a_0], R \in [3.0\,a_0, 8.0\,a_0], \theta_1, \theta_2, \phi \in [0°, 180°]$.
- Ethanol C_2H_5OH: 64 evenly distributed torsion angles between $0°$ and $360°$.

K Equilibrium structures

To find equilibrium structures, we performed high-resolution energy surface scans to find the global minimum. As many structures result in identical energies when performing inference with float32 precision, we switch to float64 precision. Note that all models were still trained with float32 and just converted for finding the equilibrium structure. We visualize these high dimensional energy surfaces around the equilibrium geometries for the nitrogen and lithium dimer in Figure 14.