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

Anonymous Author(s)
Affiliation
Address
email

Abstract

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

1 Introduction

The challenge of the Schrödinger Equation Accurately predicting properties of molecules and materials is of utmost importance for many applications, including the development of new materials or pharmaceuticals. In principle, any property of any molecule can be calculated from its wavefunction, which is obtained by solving the Schrödinger equation. In practice, computing accurate wavefunctions and corresponding energies is computationally extremely difficult for two reasons: First, the wavefunction is a high-dimensional function, depending on all coordinates of all electrons, subjecting most methods to the curse of dimensionality. Second, the required level of accuracy is extremely high. While total energies of small molecules are typically hundreds of Hartrees, the chemically relevant energy differences are on the order of 1 milli-Hartree as depicted in Fig. 1. Decades of research have produced a plethora of methods, which all require a trade-off between accuracy and computational cost: On one end of the spectrum are approximate methods such as Hartree-Fock (HF) or Density Functional Theory (DFT), which was awarded the Nobel prize in 1998. These methods can treat thousands of particles but can often only crudely approximate chemical properties. On the other end of the spectrum are "gold-standard" methods such as FCI (Full Configuration Interaction) or CCSD(T) (Coupled Clusters Singles Doubles (Perturbative Triples)) which yield energies that often closely agree with experiments, but can only treat up to 100 particles. Despite all these efforts, even for small molecules there do currently not exist highly accurate energy calculations. A 2020 benchmark of state-of-the-art methods for the benzene molecule found a spread of 4 mHa across different methods [1] – as a matter of fact, our results show that the absolute energies

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

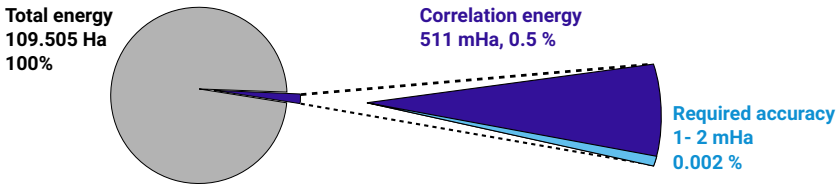

Figure 1: Conceptual visualization of the required accuracies on the example of an N_2 molecule: For chemical applications, total energies must achieve accuracies of ∼99.998%.

calculated in [1] are off by at least 600 mHa, due to the small basis set used in their calculations. An important characteristic of a method is the type of approximation being made: Hartree-Fock or FCI are "variational", meaning that their predicted energies at least upper-bound the ground-truth energy. Since a lower energy is always guaranteed to be a closer approximation of the true energy, this makes assessment of these methods straight-forward. In contrast, CCSD(T) or DFT do not have any guarantees on the accuracy of their results. The approximations resulting from such methods, while working well for many common systems, often fail for chemically challenging situations such as breaking of chemical bonds [2, 3].

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

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

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

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

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

$$\psi_\theta(\boldsymbol{r}) = \sum_{d=1}^{n_{\text{det}}} \det\left[\Lambda^d_{ki}(\boldsymbol{r})\Omega^{d\alpha_i}_k(\boldsymbol{r}_i)\right]_{i,k=1,\dots,n_{\text{el}}} \tag{2}$$

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

Related work Two major neural network architectures and their extensions have emerged throughout literature: PauliNet [9] and FermiNet [10]. PauliNet puts emphasis on maximizing physical prior knowledge, by focusing on the the envelope function. They use the output of CASSCF (Complete Active Space Self Consistent Field, a sophisticated conventional quantum-chemistry method) as Ω

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

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

2 Improved approach

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

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

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

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

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

2.1 Convolutional layers in embedding

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

$$\boldsymbol{h}_i^{l+1} = \boldsymbol{A}^l_{\text{one}}\big(\boldsymbol{f}_i^l\big) + \boldsymbol{h}_i^l \qquad \boldsymbol{g}_{ij}^{l+1} = \boldsymbol{A}^l_{\sigma_{ij}}\big(\boldsymbol{g}_{ij}^l\big) + \boldsymbol{g}_{ij}^l \qquad \boldsymbol{v}_{iI}^{l+1} = \boldsymbol{A}^l_{\text{nuc}}\big(\boldsymbol{v}_{iI}^l\big) + \boldsymbol{v}_{iI}^l \tag{4}$$

Here l denotes the embedding iteration, $\boldsymbol{A}^l$ denote fully connected neural networks, and $\boldsymbol{g}_{ij}^l$, $\boldsymbol{v}_{iI}^l$ denote electron-electron- and electron-nucleus-embeddings. We use σ_{ij} = 'same' for same-spin pairs of electrons and σ_{ij} = 'diff' for pairs of electrons with different spin. Similar to FermiNet, in each iteration we assemble the input $\boldsymbol{f}_i^l$ to the primary stream from the auxiliary streams (Fig. 2c):

$$\boldsymbol{f}_i^l = \left[\boldsymbol{h}_i^l, \quad \frac{1}{n_\uparrow}\sum_{j=1}^{n_\uparrow}\boldsymbol{h}_j^l, \quad \frac{1}{n_\downarrow}\sum_{j=1+n_\uparrow}^{n_{el}}\boldsymbol{h}_j^l, \quad \boldsymbol{s}_i^{l,\text{el}}, \quad \boldsymbol{s}_i^{l,\text{nuc}}\right]. \tag{5}$$

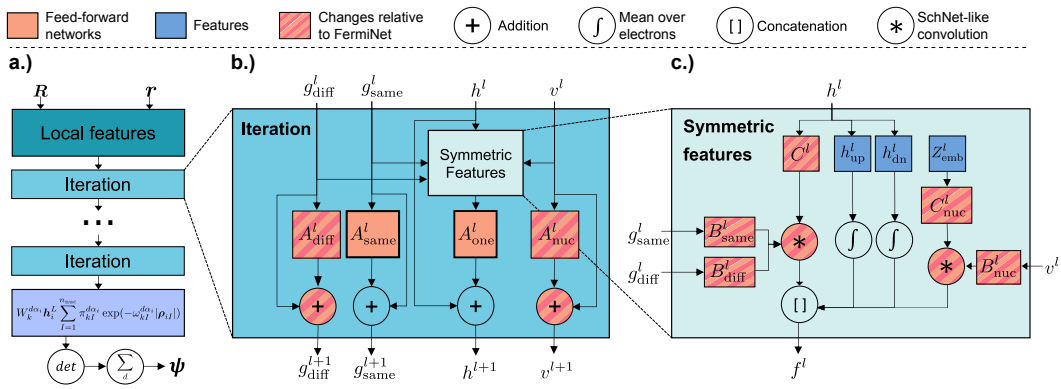

Figure 2: Our architecture: a) High-level overview b) One single embedding iteration c) Sub-block of assembling symmetric features

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

$$\boldsymbol{s}_i^{l,\mathrm{el}} = \sum_{j=1}^{n_{el}} \boldsymbol{B}^l_{\sigma_{ij}}\left(\boldsymbol{g}^l_{ij}\right) \odot \boldsymbol{C}^l_{\sigma_{ij}}\left(\boldsymbol{h}^l_j\right) \qquad \boldsymbol{s}_i^{l,\mathrm{nuc}} = \sum_{I=1}^{n_{ion}} \boldsymbol{B}^l_{\mathrm{nuc}}\left(\boldsymbol{v}^l_{iI}\right) \odot \boldsymbol{C}^l_{\mathrm{nuc}}\left(Z^{\mathrm{emb}}_I\right) \tag{6}$$

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

2.2 Local, invariant input features

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

We propose using local coordinate systems centered on every nucleus and evaluating the electron-nuclei differences in these local coordinate systems. Effectively this amounts to applying a rotation matrix U_J to the raw electron-nucleus differences: $\tilde{\boldsymbol{\rho}}_{iJ} = U_J \boldsymbol{\rho}_{iJ}$. As opposed to [12] where U is constant for all nuclei, in our approach U_J can be different for every nucleus. These electron-nucleus differences $\tilde{\boldsymbol{\rho}}_{iJ}$ are invariant under rotations, contain all the information contained in raw Cartesian coordinates and depend primarily on the local environment of an atom. To compute the 3x3 rotation matrices U_J, we first run a single Hartree-Fock calculation using a minimal basis set to obtain a density matrix D. For each atom J we choose the 3x3 block of the density matrix which

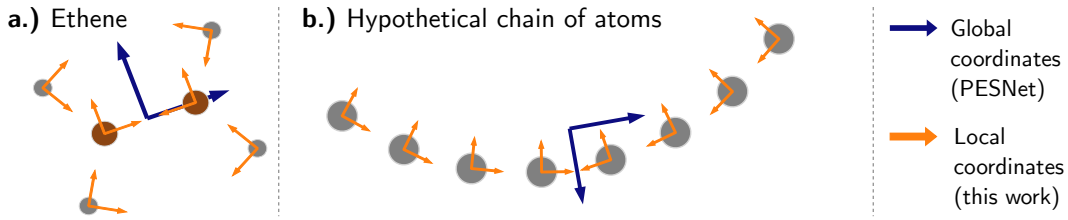

Figure 3: Visualization of resulting coordinate systems for 2 example molecules: a) Ethene b) A hypothetical bent chain of atoms.

corresponds to the 3 p-orbitals of the atom J and compute its eigenvectors $U_J = \text{eig}(D_J)$. To make our coordinate system unique we sort the eigenvectors of D_J by their corresponding eigenvalues. If the eigenvalues are degenerate, we pick the rotation of the eigenvectors that maximizes the overlap with the coordinate-axes of the previous nucleus. Fig. 3 shows the resulting coordinate systems spanned by U_J. Note that the local coordinate system generally depicts more physically meaningful directions such as "along the chain". We find that these local coordinates slightly increase the accuracy for single geometries, but more importantly we expect the wavefunction to generalize better across different molecules or geometries. This should improve the accuracy of approaches that attempt to learn wavefunctions for multiple geometries at once [11, 12].

2.3 Initialization of orbital envelope weights

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

$$\Lambda_{ki}(\boldsymbol{r}) \exp\left(-\omega_k |\boldsymbol{\rho}_{iI}|\right).$$

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

$$\widetilde{\Lambda}_k(\boldsymbol{\rho}_{iI}) \exp\left(-\frac{Z}{n_k}|\boldsymbol{\rho}_{iI}|\right),$$

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

2.4 Improved hyperparameters

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

3 Results of improved approach

We evaluated the accuracy of our approach by comparing our computed energies against the most accurate references available in the literature. Fig. 4 compares our energies against variational methods – for which lower energies are guaranteed to be more accurate – as well as non-variational high-accuracy methods. We find that across many different systems (small and large atoms, molecules at equilibrium geometry, molecules in transition states), our approach yields substantially lower – and thus more accurate – energies than previous variational results. Across all tested systems, we

outperform almost all existing variational methods, both deep-learning-based methods as well as classical ones. When comparing to high-accuracy FermiNet VMC calculations, we not only reach substantially lower energies, but also do so using 3-4x fewer training steps, with each step being 40% faster (cf. appendix C). Comparing to a concurrently published Diffusion Monte Carlo approach, which used $\sim$10x more computational resources, we achieve similar or better accuracy for molecules like N_2 and cyclobutadiene and slightly lower accuracy for benzene. Non-variational methods (e.g. CCSD(T)) yield slightly lower energies than our calculations for some molecules, but since those methods do not provide upper bounds or uncertainty guarantees they do not provide a ground-truth. For many applications not only absolute energies are important, but energy differences between

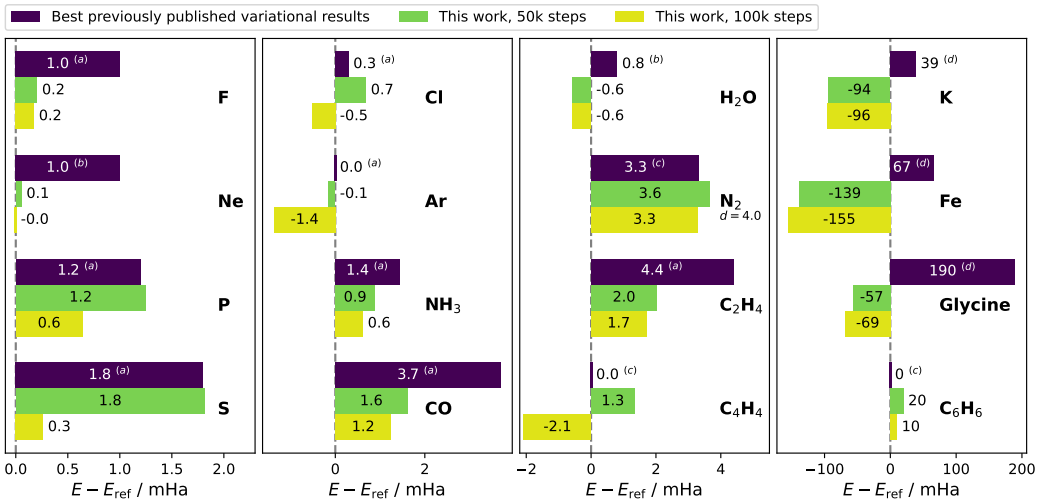

Figure 4: Energies relative to the previously known best estimate, (lower is better). Blue bars depict best published variational energies, footnotes mark the method: a: FermiNet VMC [10, 17], b: Conventional DMC [25, 26, 27], c: FermiNet DMC [21], d: MRCI-F12. A table of absolute energies and methods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

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

4 Ablation study

To investigate which specific changes lead to the observed improvements in accuracy, we start from the improved FermiNet architecture proposed in [17] and incrementally add improvements in the following order: First, we use dense $n_{\text{el}} \times n_{\text{el}}$ determinants introduced by the FermiNet authors [10, 17] in their GitHub repository and described in [18] instead of block-diagonal determinants. This generalization increases computational cost and parameter count (cf. appendix C) but has been found to better describe the wavefunction's nodal surface and thus increase expressiveness of the ansatz. Second, we change hyperparameters as described in Sec. 2.4, which increases throughput by $\sim$ 2x. Third, we augment the electron embedding using our new SchNet-like neural network architecture described in Sec. 2.1. This leads to a moderate increase in parameter count and computational cost.

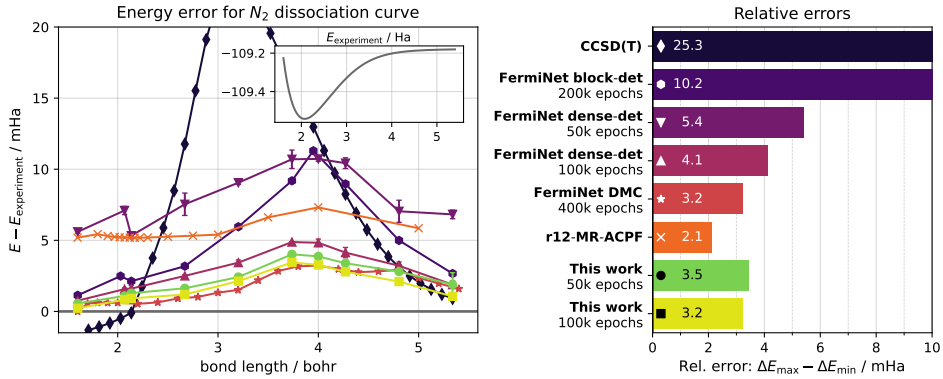

Figure 5: Comparison of energy error E - $E_{\text{experiment}}$ for the dissociation of the N_2 molecule across various methods. Errorbars corresponds to the standard deviation wrt. two different seeds. Errorbars for our work are too small to be visible ($\sim$ 0.1 mHa). Results for $E_{\text{experiment}}$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

Fourth, we switch to local, invariant input features as described in Sec. 2.2 and remove the electron-electron difference vectors $\boldsymbol{r}_{ij}$ as inputs. Lastly we switch to initializing $\omega^d_{kI} = Z/n_k$ as described in Sec. 2.3, resulting in our proposed final method. We note that the accuracy gains of these changes are not fully independent of each other and the relative attribution depends on the order in which they are applied: Earlier changes will generally generate larger energy improvements compared to later changes. At each step we compute total energies for three different molecules: ethene, N_2 at the challenging bond-length of 4.0 bohr, and the K-atom. Fig. 6 depicts the energy of our implementation of FermiNet, and the energy change caused by each subsequent improvement. Each experiment was repeated two times with different RNG seeds (appendix D), the errorbars depict the spread in energy. Overall we find that all our changes combined yield a $\sim$3-20x improvement in the energy error. For ethene, the dominant contribution (3.7 mHa) comes from improved hyperparameters, which

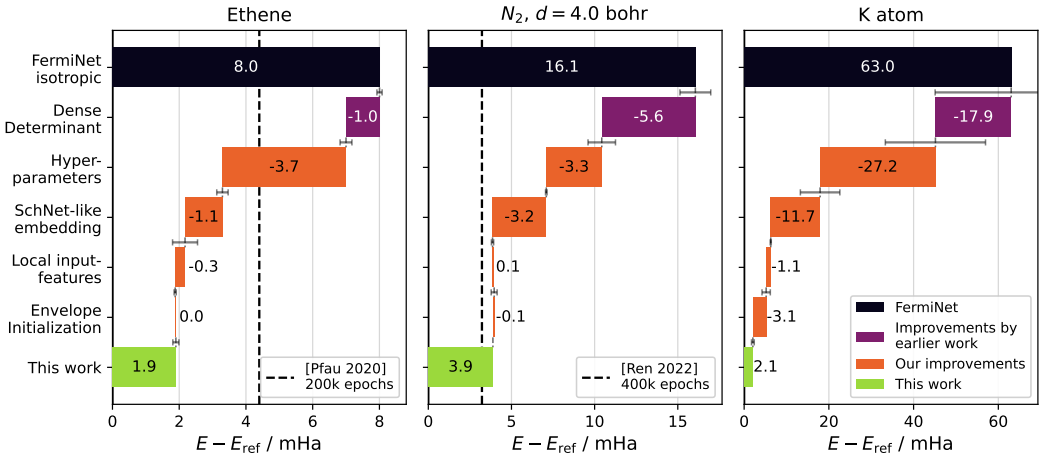

Figure 6: Breakdown of accuracy improvements for three different molecules, each trained for 50k epochs. The black dashed line depicts the best published variational result and as reference we use the best available estimates: CCSD(T) [10] for ethene, experimental data for N_2 [29], and our own VMC calculations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

lead to the results being mostly converged after 50k epochs vs. the original settings which require 200k epochs for convergence. Architectural improvements (embedding and input features) lower the energy error by additional 1.4 mHa. For N_2 it has already been observed that a single dense determinant can outperform models with multiple block-diagonal determinants [18]. We find the corresponding result that 32 dense determinants substantially lower the energy relative to an ansatz with 32 block-diagonal determinants. Comparing N_2 to ethene, we observe larger contributions

from our architectural improvements and smaller contributions from improved hyperparameters. For the K atom, the overall gains are largest, totalling 60mHa, with substantial accuracy gains from all improvements. Since K has a much larger nuclear charge (Z=19) than the constituents of ethene (Z=1,6) and N_2 (Z=7), also the physics-inspired initialization of the envelope parameters yields a substantial contribution.

5 Incorporating prior knowledge

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

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

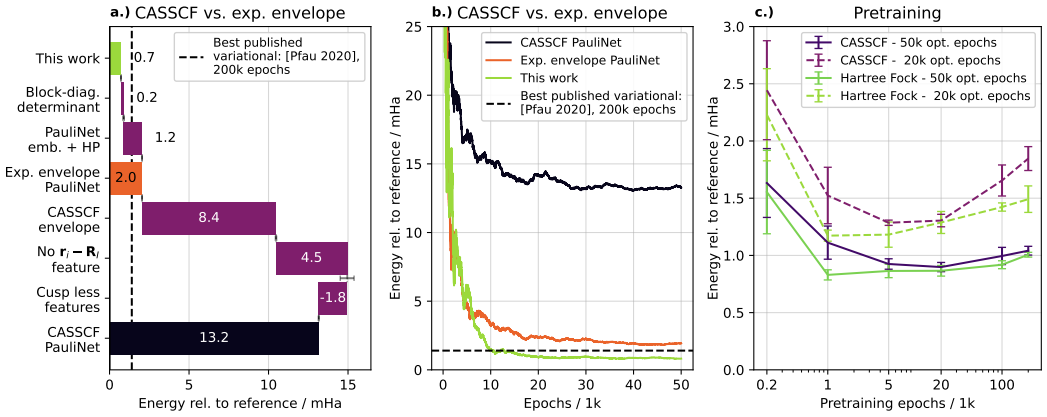

Figure 7: Effect of prior knowledge for NH_3: a) Change in accuracy after 50k optimization epochs when transitioning from our approach to a PauliNet-like architecture, by increasing prior knowledge. b) Energy error for 3 architectures over number of training epochs. c) Mean and standard deviation of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

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

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

6 Discussion and Limitations

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

Regarding the question of how much physics to include in the model, we find varying results. For exact physical constraints, such as symmetries or the cusp conditions, inclusion in the model generally appears to be helpful. However for prior knowledge from existing approximate solutions (such as CASSCF) the situation is more subtle. On the one hand, soft physical guidance such as short supervised pre-training or physics-inspired weight initialization accelerates optimization. On the other hand, we show empirically that increasing physical prior knowledge, e.g. by incorporating CASSCF or extensive supervised pre-training, does not necessarily increase accuracy, but can in fact introduce detrimental biases that are hard to overcome during wavefunction optimization.

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

References

[1] Janus J. Eriksen et al. "The Ground State Electronic Energy of Benzene". In: *The Journal of Physical Chemistry Letters* 11.20 (Oct. 2020). Publisher: American Chemical Society, pp. 8922–8929. DOI: 10.1021/acs.jpclett.0c02621. URL: https://doi.org/10.1021/acs.jpclett.0c02621 (visited on 04/21/2022).

[2] Narbe Mardirossian and Martin Head-Gordon. "Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals". In: *Molecular Physics* 115.19 (2017), pp. 2315–2372. DOI: 10.1080/00268976.2017.1333644. eprint: https://doi.org/10.1080/00268976.2017.1333644. URL: https://doi.org/10.1080/00268976.2017.1333644.

[3] Aron J. Cohen, Paula Mori-Sánchez, and Weitao Yang. "Challenges for Density Functional Theory". In: *Chemical Reviews* 112.1 (Jan. 2012). Publisher: American Chemical Society, pp. 289–320. ISSN: 0009-2665. DOI: 10.1021/cr200107z. URL: https://doi.org/10.1021/cr200107z.

[4] Jiequn Han, Linfeng Zhang, and Weinan E. "Solving many-electron Schrödinger equation using deep neural networks". In: *J. Comput. Phys.* 399 (Dec. 2019), p. 108929. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.108929. URL: http://dx.doi.org/10.1016/j.jcp.2019.108929.

[5] Giuseppe Carleo and Matthias Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325 (Feb. 2017). Publisher: American Association for the Advancement of Science, pp. 602–606. DOI: 10.1126/science.aag2302. URL: https://www.science.org/doi/10.1126/science.aag2302 (visited on 05/09/2022).

[6] Wolfgang Pauli. *Nobel lecture*. 1946.

[7] Walter Ritz. "Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik." In: 1909.135 (1909), pp. 1–61. DOI: doi:10.1515/crll.1909.135.1. URL: https://doi.org/10.1515/crll.1909.135.1.

[8] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (Apr. 1970), 97–109. ISSN: 0006-3444. DOI: 10.1093/biomet/57.1.97. eprint: https://academic.oup.com/biomet/article-pdf/57/1/97/23940249/57-1-97.pdf. URL: https://doi.org/10.1093/biomet/57.1.97.

[9] Jan Hermann, Zeno Schätzle, and Frank Noé. "Deep-neural-network solution of the electronic Schrödinger equation". In: *Nat. Chem.* 12.10 (Oct. 2020), 891–897. ISSN: 1755-4349. DOI: 10.1038/s41557-020-0544-y. URL: https://doi.org/10.1038/s41557-020-0544-y.

[10] David Pfau et al. "Ab initio solution of the many-electron Schrödinger equation with deep neural networks". In: *Phys. Rev. Res.* 2 (3 Sept. 2020), p. 033429. DOI: 10.1103/PhysRevResearch.2.033429. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429.

[11] Michael Scherbela et al. *Solving the electronic Schrödinger equation for multiple nuclear geometries with weight-sharing deep neural networks*. 2021. DOI: 10.48550/ARXIV.2105.08351. URL: https://arxiv.org/abs/2105.08351.

[12] Nicholas Gao and Stephan Günnemann. *Ab-Initio Potential Energy Surfaces by Pairing GNNs with Neural Wave Functions*. 2021. DOI: 10.48550/ARXIV.2110.05064. URL: https://arxiv.org/abs/2110.05064.

[13] Mike Entwistle et al. *Electronic excited states in deep variational Monte Carlo*. 2022. DOI: 10.48550/ARXIV.2203.09472. URL: https://arxiv.org/abs/2203.09472.

[14] Max Wilson et al. *Wave function Ansatz (but Periodic) Networks and the Homogeneous Electron Gas*. 2022. DOI: 10.48550/ARXIV.2202.04622. URL: https://arxiv.org/abs/2202.04622.

[15] G. Cassella et al. *Discovering Quantum Phase Transitions with Fermionic Neural Networks*. DOI: 10.48550/ARXIV.2202.05183. URL: https://arxiv.org/abs/2202.05183.

[16] Xiang Li, Zhe Li, and Ji Chen. "Ab initio calculation of real solids via neural network ansatz". In: *arXiv:2203.15472 [cond-mat, physics:physics]* (Mar. 2022). arXiv: 2203.15472. DOI: 10.48550/ARXIV.2203.15472. URL: http://arxiv.org/abs/2203.15472 (visited on 04/20/2022).

[17] James S. Spencer et al. *Better, Faster Fermionic Neural Networks*. 2020. DOI: 10.48550/ARXIV.2011.07125. URL: https://arxiv.org/abs/2011.07125.

[18] Jeffmin Lin, Gil Goldshlager, and Lin Lin. "Explicitly antisymmetrized neural network layers for variational Monte Carlo simulation". In: *arXiv:2112.03491 [physics]* (Dec. 2021). arXiv: 2112.03491. DOI: 10.48550/ARXIV.2112.03491. URL: http://arxiv.org/abs/2112.03491 (visited on 03/25/2022).

[19] Xiang Li et al. "Fermionic neural network with effective core potential". In: *Physical Review Research* 4.1 (Jan. 2022). Publisher: American Physical Society, p. 013021. DOI: 10.1103/PhysRevResearch.4.013021. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.4.013021 (visited on 04/26/2022).

[20] Max Wilson et al. *Simulations of state-of-the-art fermionic neural network wave functions with diffusion Monte Carlo*. 2021. DOI: 10.48550/ARXIV.2103.12570. arXiv: 2103.12570 [physics.chem-ph].

[21] Weiluo Ren, Weizhong Fu, and Ji Chen. *Towards the ground state of molecules via diffusion Monte Carlo on neural networks*. 2022. DOI: 10.48550/ARXIV.2204.13903. URL: https://arxiv.org/abs/2204.13903.

[22] Kristof Schütt et al. "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

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

[24] Aleksandar Botev and James Martens. *KFAC-JAX*. Version 0.0.1. 2022. URL: http://github.com/deepmind/kfac-jax.

[25] P. Seth, P. López Ríos, and R. J. Needs. "Quantum Monte Carlo study of the first-row atoms and ions". In: *The Journal of Chemical Physics* 134.8 (2011), p. 084105. DOI: 10.1063/1.3554625. eprint: https://doi.org/10.1063/1.3554625. URL: https://doi.org/10.1063/1.3554625.

[26] Norbert Nemec, Michael D. Towler, and R. J. Needs. "Benchmark all-electron ab initio quantum Monte Carlo calculations for small molecules". In: *The Journal of Chemical Physics* 132.3 (Jan. 2010). Publisher: American Institute of Physics, p. 034111. ISSN: 0021-9606. DOI: 10.1063/1.3288054. URL: http://aipscitation.org/doi/full/10.1063/1.3288054 (visited on 04/12/2022).

[27] Bryan K. Clark et al. "Computing the energy of a water molecule using multideterminants: A simple, efficient algorithm". In: *The Journal of Chemical Physics* 135.24 (Dec. 2011). Publisher: American Institute of Physics, p. 244105. ISSN: 0021-9606. DOI: 10.1063/1.3665391. URL: http://aipscitation.org/doi/full/10.1063/1.3665391 (visited on 05/31/2022).

[28] Robert J. Gdanitz. "Accurately solving the electronic Schrödinger equation of atoms and molecules using explicitly correlated (r12-)MR-CI: the ground state potential energy curve of N2". In: *Chemical Physics Letters* 283.5 (1998), pp. 253–261. ISSN: 0009-2614. DOI: https://doi.org/10.1016/S0009-2614(97)01392-4. URL: https://www.sciencedirect.com/science/article/pii/S0009261497013924.

[29] Robert J. Le Roy, Yiye Huang, and Calvin Jary. "An accurate analytic potential function for ground-state N2 from a direct-potential-fit analysis of spectroscopic data". In: *The Journal of Chemical Physics* 125.16 (2006), p. 164310. DOI: 10.1063/1.2354502. eprint: https://doi.org/10.1063/1.2354502. URL: https://doi.org/10.1063/1.2354502.

[30] Tosio Kato. "On the eigenfunctions of many-particle systems in quantum mechanics". In: *Commun. Pure Appl. Math.* 10.2 (1957), 151–177. DOI: 10.1002/cpa.3160100201. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160100201. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160100201.

[31] C. J. Umrigar, M. P. Nightingale, and K. J. Runge. "A diffusion Monte Carlo algorithm with very small time-step errors". In: *The Journal of Chemical Physics* 99.4 (1993), pp. 2865–2890. DOI: 10.1063/1.465195. eprint: https://doi.org/10.1063/1.465195. URL: https://doi.org/10.1063/1.465195.

[32] C. J. Umrigar. "Accelerated Metropolis method". In: *Phys. Rev. Lett.* 71 (3 1993), pp. 408–411. DOI: 10.1103/PhysRevLett.71.408. URL: https://link.aps.org/doi/10.1103/PhysRevLett.71.408.

[33] Subhas J. Chakravorty et al. "Ground-state correlation energies for atomic ions with 3 to 18 electrons". In: *Phys. Rev. A* 47 (5 1993), pp. 3649–3670. DOI: 10.1103/PhysRevA.47.3649. URL: https://link.aps.org/doi/10.1103/PhysRevA.47.3649.

[34] Bruce J. Rosenberg and Isaiah Shavitt. "Ab initio SCF and CI studies on the ground state of the water molecule. I. Comparison of CGTO and STO basis sets near the Hartree–Fock limit". In: *The Journal of Chemical Physics* 63.5 (Sept. 1975). Publisher: American Institute of Physics, pp. 2162–2174. ISSN: 0021-9606. DOI: 10.1063/1.431596. URL: http://aipscitation.org/doi/10.1063/1.431596 (visited on 05/31/2022).

[35] H.-J. Werner et al. "MOLPRO: a general-purpose quantum chemistry program package". In: *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 2 (2012), 242–253.

[36] H.-J. Werner et al. *MOLPRO, version 2012.1, a package of ab initio programs*. see https://www.molpro.net. Cardiff, UK, 2012.

[37] Thomas B. Adler, Gerald Knizia, and Hans-Joachim Werner. "A simple and efficient CCSD(T)-F12 approximation". In: *J. Chem. Phys.* 127.22 (2007), p. 221106. DOI: 10.1063/1.2817618. eprint: https://doi.org/10.1063/1.2817618. URL: https://doi.org/10.1063/1.2817618.

[38] Kirk A. Peterson, Thomas B. Adler, and Hans-Joachim Werner. "Systematically convergent basis sets for explicitly correlated wavefunctions: The atoms H, He, B–Ne, and Al–Ar". In: *J. Chem. Phys.* 128.8 (2008), p. 084102. DOI: 10.1063/1.2831537. eprint: https://doi.org/10.1063/1.2831537. URL: https://doi.org/10.1063/1.2831537.

[39] J. Grant Hill, Shivnath Mazumder, and Kirk A. Peterson. "Correlation consistent basis sets for molecular core-valence effects with explicitly correlated wave functions: The atoms B–Ne and Al–Ar". In: *J. Chem. Phys.* 132.5 (2010), p. 054108. DOI: 10.1063/1.3308483. eprint: https://doi.org/10.1063/1.3308483. URL: https://doi.org/10.1063/1.3308483.

[40] Florian Weigend, Filipp Furche, and Reinhart Ahlrichs. "Gaussian basis sets of quadruple zeta valence quality for atoms H–Kr". In: *J. Chem. Phys.* 119.24 (2003), pp. 12753–12762. DOI: 10.1063/1.1627293. URL: https://doi.org/10.1063/1.1627293.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We claim 2x higher accuracy, which we demonstrate in Fig. 4 and Fig. 6. We claim 6x speed-up which we demonstrate to be a combination of needing 3-4x fewer epochs (Sec. 4) and $\sim$40% faster epochs (Sec. 6 in main text and detailed further in Tab. 5 in the appendix).
   (b) Did you describe the limitations of your work? [Yes] in Sec. 6 in the paragraph *Limitations*
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] While we do not foresee any near-term negative impact from this basic research, we discuss potential impacts in appendix E.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Our paper conforms to the ethics review guidelines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code, configuration files and data are in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The choice of the most relevant training details is discussed in Sec. 2.4, the full set of hyperparameters is listed in Tab. 2 in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We depict error bars where possible (e.g. in Fig. 6, 7) and show in Sec. D that errors wrt. the random seed are small.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We state the used compute resources in C in the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite all used assets, in particular code and reference energies.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our source code in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Absolute energies

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

Category	System	Lowest Variational	Lowest All	This work 50k epochs	This work 100k epochs
2nd row atoms	O	-75.0666 [a]	$-\underline{75.0673}$ [f]	-75.0669	$\mathbf{-75.0670}$
	F	-99.7329 [a]	$-\underline{99.7339}$ [f]	$\mathbf{-99.7337}$	$\mathbf{-99.7337}$
	Ne	-128.9366 [c]	$-\underline{128.9376}$ [f]	-128.9375	$\mathbf{-128.9376}$
3rd row atoms	P	-341.2578 [a]	$-\underline{341.259}$ [f]	-341.2578	$\mathbf{-341.2584}$
	S	-398.1082 [a]	$-\underline{398.110}$ [f]	-398.1082	$\mathbf{-\underline{398.1097}}$
	Cl	-460.1477 [a]	-460.148 [f]	-460.1473	$\mathbf{-\underline{460.1485}}$
	Ar	-527.5405 [a]	-527.5405 [a]	-527.5406	$\mathbf{-\underline{527.5419}}$
4th row atoms	K	-599.7847 [d]	-599.8238 [g]	-599.9178	$\mathbf{-\underline{599.9195}}$
	Fe	-1263.4281 [d]	-1263.4947 [g]	-1263.633	$\mathbf{-\underline{1263.650}}$
Small molecules	N_2, $d = 2.068$	$\mathbf{-109.5416}$ [b]	$-\underline{109.5425}$ [e]	-109.5408	-109.5414
	Ethene	-78.5844 [a]	$-\underline{78.5888}$ [e]	-78.5868	$\mathbf{-78.5871}$
	H_2O	-76.4368 [c]	-76.4376 [h]	-76.4382	$\mathbf{-\underline{76.4382}}$
	CO	-113.3218 [a]	$-\underline{113.3255}$ [e]	-113.3239	$\mathbf{-113.3242}$
	NH_3	-56.5630 [a]	$-\underline{56.5644}$ [e]	-56.5635	$\mathbf{-56.5638}$
Larger molecules	Cyclobutadiene	-154.6770 [b]	-154.67700 [b]	-154.6757	$\mathbf{-\underline{154.6791}}$
	Benzene	$\mathbf{-\underline{232.2370}}$ [b]	$\mathbf{-\underline{232.2370}}$ [b]	-232.2168	-232.2267
	Glycine	-284.1741 [d]	-284.3639 [g]	-284.4201	$\mathbf{-\underline{284.4328}}$

B Computational settings

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

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

Table 2: Hyperparameter settings used in this work

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

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

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

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

Molecule	Basis set	E_{HF}	E_{CASSCF}	$E_{CCSD(T)\text{-}F12}$	$E_{MRCI\text{-}F12}$	$E_{MRCI\text{-}F12(Q)}$
H_2O	cc-pVQZ-F12	-76.0673	-76.1208	-76.3766	-76.4210	-76.4352
	cc-pVTZ-F12	-76.0669	-76.1189	-76.3752	-76.4183	-76.4324
Cyclo-butadiene	cc-pVQZ-F12	-153.7060	-153.7598	-154.4607	-154.5288	-154.6380
	cc-pVTZ-F12	-153.7054	-153.7574	-154.4597	-154.5250	-154.6340
Benzene	cc-pVQZ-F12	-230.7970	–	-231.9115	–	–
	cc-pVTZ-F12	-230.7960	-230.8580	-231.9099	-231.9584	-232.1483
Glycine	cc-pVQZ-F12	-282.9740	-282.9902	-284.1623	-284.1741	-284.3639
	cc-pVTZ-F12	-282.9730	-282.9847	-284.1587	-284.1671	-284.3570

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

Atom	E_{CASSCF}	$E_{MRCI\text{-}F12}$	$E_{MRCI\text{-}F12(Q)}$
K	-599.1645	-599.7847	-599.8238
Fe	-1262.4459	-1263.4281	-1263.4947

C Computational cost

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

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

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

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

System	FermiNet epochs	This work, 50k epochs		This work, 100k epochs	
		accuracy gain	speed-up	accuracy gain	speed-up
F	200,000	61%	6x	63%	3x
Ne	200,000	94%	6x	100%	3x
P	350,000	-4%	10x	46%	5x
S	350,000	0%	10x	85%	5x
Cl	350,000	-48%	10x	100%	5x
Ar	350,000	11%	10x	100%	5x
NH_3	200,000	38%	6x	57%	3x
CO	200,000	45%	6x	53%	3x
N_2	200,000	52%	6x	69%	3x
C_2H_4	200,000	53%	6x	61%	3x
C_4H_4	200,000	71%	6x	100%	3x

D Run-to-run variation

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

E Broader impact

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