

QMCTorch: Molecular Wave Function with Neural Components for Energy and Force Calculations

Nicolas Renaud^{1*}

¹ Netherlands eScience Center, Matrix THREE Science park 402, 1098 XH Amsterdam, The Netherlands

* nicolas.gm.renaud@gmail.com

Abstract

In this paper, we present results obtained using QMCTorch, a modular framework for real-space Quantum Monte Carlo (QMC) simulations of small molecular systems. Built on the popular deep learning library PyTorch, QMCTorch is GPU-native and enables the integration of machine learning-inspired components into the wave function ansatz, such as neural network backflow transformations and Jastrow factors, while leveraging efficient optimization algorithms. QMCTorch interfaces with two widely used quantum chemistry packages - PySCF and ADF - which provide initial values for atomic orbital exponents and molecular orbital coefficients. In this study, we present wavefunction optimizations for four molecules: H_2 , LiH , Li_2 , and CO , using various wavefunction ansätze. We also compute their dissociation energy curves and the corresponding interatomic forces along these curves. Our results show good agreement with baseline calculations and recover a significant portion of the correlation energy. QMCTorch provides a modular and extendable platform for rapidly prototyping new wavefunction ansätze, evaluating their performance, and analyzing optimization outcomes.

Copyright attribution to authors.

This work is a submission to SciPost Chemistry.

License information to appear upon publication.

Publication information to appear upon publication.

Received Date

Accepted Date

Published Date

1

Contents

1	Introduction	2
2	Methods	4
2.1	Wave Function Ansatz	4
2.2	Gradients of the Wave Function	5
2.3	Interatomic Forces	5
2.4	Computational Performance	6
3	Results	7
3.1	Wavefunction Optimization	7
3.2	Wavefunction and Energy Landscape of H_2	8
3.3	Energy Curve and Interatomic Forces	9
4	Conclusion	10

14	A Computational Performance	12
15	B Parameters for the Wavefunction Optimization	13
16	C Neural Electron-Electron Jastrow Factor	14
17	D Additional Wavefunction Ansatz	15
18	D.1 Electron-Electron Padé Jastrow Factor	15
19	D.2 Many-body Jastrow Factor	16
20	D.3 Neural Backflow Transformation	17
21	E Wavefunction and Energy Profile of LiH and Li_2	17
22	F Computation of the Local Energies of Backflow-Transformed Wavefunctions	18
23	F.1 Derivative of a Single Determinant	19
24	F.2 Derivative of Excited-State Determinants	20
25	G Simplification of Eq. F.5 in Absence of Backflow Transformation	21
26	H Derivation of Eqs. F.6 and F.8	22
27	H.1 Inverse of A_X	23
28	H.2 Demonstration of eq. F.6	23
29	H.3 Demonstration of eq. F.8	24
30	References	26

1 Introduction

The electronic structure of molecular systems is typically determined by solving the time-independent Schrödinger equation: $\mathcal{H}\Psi(\mathbf{r}) = E\Psi(\mathbf{r})$ where $\Psi(\mathbf{r})$ represents the many-body wave function, \mathbf{r} denotes the electronic coordinates and \mathcal{H} is the molecular Hamiltonian:

$$\mathcal{H} = -\frac{1}{2} \sum_i \nabla_i^2 + \sum_{i>j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_i \sum_\alpha \frac{Z_\alpha}{|\mathbf{r}_i - \mathbf{R}_\alpha|} + \sum_{\alpha>\beta} \frac{Z_\alpha Z_\beta}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|} \quad (1)$$

where indexes i, j run over the electrons and α, β over the nuclei. The first term in eq. (1), $\nabla_i^2 = \frac{d^2}{dx_i^2} + \frac{d^2}{dy_i^2} + \frac{d^2}{dz_i^2}$, represents the kinetic energy of the i -th electron. The second term accounts for electron-electron repulsion while the third describes electron-nucleus attraction, with Z_α the atomic number of the α atom. The last term corresponds to the repulsive interaction between nuclei. According to the variational principle, the ground-state solution of the Schrödinger equation can be approximated by minimizing the variational energy:

$$E_v = \frac{\int \Psi^*(\mathbf{r}) \mathcal{H} \Psi(\mathbf{r}) d\mathbf{r}}{\int |\Psi(\mathbf{r})|^2 d\mathbf{r}} \geq E_0 \quad (2)$$

that reaches the ground state energy E_0 when Ψ exactly matches the ground state wave function of the system. Solving equation (2) becomes increasingly challenging when the number of electrons in the system increases. Quantum Monte-Carlo (QMC) simulations offer a powerful alternative to mean-field and post Hartree-Fock (HF) approaches, providing high-accuracy

approximations of the variational energy. In QMC simulations, the high-dimensional integral in eq. (2) is evaluated through Monte Carlo integration [1]:

$$E_v = \int \rho(\mathbf{r}) E_L(\mathbf{r}) d\mathbf{r} \approx \frac{1}{M} \sum_{k=1}^M E_L(\mathbf{r}_k) \quad (3)$$

where \mathbf{r}_k are electronic positions, sampled from the probability distribution $\rho(\mathbf{r}) = |\Psi(\mathbf{r})|^2 / \int |\Psi(\mathbf{r})|^2 d\mathbf{r}$. The evaluation of the local energies at the sampling points: $E_L(\mathbf{r}) = \mathcal{H}\Psi(\mathbf{r})/\Psi(\mathbf{r})$ are independent from each other, enabling massive parallel execution. This characteristic makes QMC methods particularly well-suited for exascale computing environments [2].

Several high-performance software packages have been developed to solve the Schrödinger equation of molecular systems using a QMC approach [3–5]. These tools have been successfully applied to a broad range of computational chemistry problems, including excited state calculations [6–8], geometry optimization [9] and molecular dynamics simulations [10]. For a comprehensive overview of QMC methodologies and their applications, we refer the reader to Ref. [11]. In recent years, significant progress has been made in integrating machine learning techniques into QMC simulations [12–20]. In these approaches, the wave function is represented by an artificial neural network, with variational parameters optimized via automatic differentiation. By leveraging modern deep learning frameworks such as PyTorch [21], TensorFlow [22], or JAX [23], these methods can efficiently utilize hardware accelerators, significantly enhancing computational performance.

Two notable machine learning-inspired approaches, Ferminet [24] and Paulinet [25], have demonstrated remarkable accuracy in computing ground-state energies of molecular systems. Both methods are capable of recovering 99% or more of the correlation energy across a diverse set of molecules. Ferminet employs a wave function ansatz that enforces only the antisymmetry requirement of the wavefunction, delegating the task of learning atomic and molecular orbital structure entirely to the neural network. In contrast, Paulinet incorporates molecular orbitals obtained from conventional quantum chemistry calculations and augments them with neural network components to capture electronic correlation effects more effectively. These approaches have since been extended to excited-state calculations [26–28], underscoring the growing impact of machine learning techniques in advancing quantum chemistry.

We introduce QMCTorch [29], a real-space Quantum Monte Carlo (QMC) framework for small molecular systems, built on the PyTorch deep learning library [21]. In QMCTorch, the wave function is represented using a Slater-Jastrow ansatz, where atomic orbitals, molecular orbitals, Jastrow factors, and optionally backflow transformations are implemented as differentiable layers. The parameters of these components are optimized to minimize the variational energy using gradient-based methods. QMCTorch interfaces with PySCF [30] and ADF [31] to initialize atomic orbital exponents and molecular orbital coefficients. Efficient kinetic energy estimators, originally developed by Filippi et al. [32, 33], are extended here to support backflow-transformed wave functions, replacing standard automatic differentiation techniques. Additionally, QMCTorch enables the evaluation of interatomic forces by treating atomic coordinates as variational parameters within the wave function ansatz. This capability opens the door to applications such as geometry optimization and molecular dynamics simulations.

2 Methods

2.1 Wave Function Ansatz

Like many QMC codes, QMCTorch employs a Slater-Jastrow wave function ansatz defined as:

$$\Psi_{\theta}(\mathbf{r}) = e^{\mathcal{J}(\mathbf{r})} \cdot \sum_n d_n D_n^{\uparrow}(\tilde{\mathbf{r}}^{\uparrow}) D_n^{\downarrow}(\tilde{\mathbf{r}}^{\downarrow}) \quad (4)$$

that satisfies the anti-symmetry principle $\Psi(r_0, \dots, r_i, \dots, r_j, \dots, r_N) = -\Psi(r_0, \dots, r_j, \dots, r_i, \dots, r_N)$. In eq. (4), $\mathcal{J}(\mathbf{r})$ denotes the Jastrow factor and $D_n^{\uparrow}(\mathbf{r}^{\uparrow})$ are the Slater determinants corresponding to spin-up and spin-down electrons respectively. The symbol θ represents the set of variational parameters included in the wavefunction. Fig. 1 illustrates the structure of the wave function ansatz. Starting from the electronic coordinates, a backflow transformation is used to capture electronic correlations [11]. This transformation is defined as: $\tilde{\mathbf{r}}_i = \mathbf{r}_i + \sum_{j \neq i} K_{\text{BF}}(r_{ij})(\mathbf{r}_i - \mathbf{r}_j)$, where K_{BF} is the backflow kernel and r_{ij} denotes the inter-electronic distance. A variety of backflow kernels have been proposed over the years, ranging from many-body formulations [34], to orbital-dependent kernels [35] and more recently to flexible neural kernels [36–39]. QMCTorch includes several built-in backflow kernels, such as inverse-distance kernels and fully connected feedforward neural network kernels. Additionally, users can define custom backflow transformations by implementing their own function for $K_{\text{BF}}(r_{ij})$. All necessary derivatives for computing the local energies can be automatically obtained via automatic differentiation, allowing users to rapidly prototype and test new backflow kernels without the need to manually derive and implement complex analytical expressions.

The backflow-transformed electronic positions are used to evaluate the atomic orbitals (AOs) of the molecular system. QMCTorch includes a fully differentiable atomic-orbital layer that computes the AO values at the given electronic coordinates. This layer supports both Gaussian and Slater Type Orbitals (STOs) whose initial exponents - and, when applicable, contraction coefficients - are provided by PySCF [30] or ADF [31], respectively. In the following, we exclusively use STOs as they naturally respect electron-nucleus cusp conditions. STOs are given by: $\varphi_{\alpha}(\tilde{\mathbf{r}}_i) = \mathcal{N} x^{k_x} y^{k_y} z^{k_z} \tilde{r}^{k_n} e^{-\zeta_{\alpha}|\tilde{\mathbf{r}}_i|}$, where x, y, z are the Cartesian components of the vector between the electron at $\tilde{\mathbf{r}}_i$ and the center of the basis function, k_m are integers determined by the quantum numbers of the atomic orbitals and \mathcal{N} is a normalization constant. The exponents ζ_{α} control the spatial extent of the orbitals and are treated as a variational parameters within the wavefunction ansatz. As illustrated in Fig. 1, the AO layer also takes the atomic coordinates as input. This design enables the computation of the local energy derivatives with respect to atomic positions via automatic differentiation, an essential feature for evaluating interatomic forces.

The molecular orbitals (MOs) are constructed as a linear combination of AOs following the expression: $\phi_{\alpha}(\tilde{\mathbf{r}}_i) = \sum_{\beta} (c_{\beta, \alpha} \times \mu_{\beta, \alpha}) \varphi_{\beta}(\tilde{\mathbf{r}}_i)$, where $c_{\beta, \alpha}$ are the molecular orbital coefficients obtained from a Hartree–Fock calculation and remain fixed during optimization. The $\mu_{\beta, \alpha}$ are variational scaling parameters introduced in the MO layer, all initialized to 1. This formulation enables the optimization of molecular orbitals while preserving their original symmetries, in contrast to directly optimizing the Hartree–Fock coefficients $c_{\beta, \alpha}$, which can mix orbitals of different symmetries.

Slater determinants are computed from MO matrices corresponding to spin-up and spin-down electrons, based on the electronic configurations specified by the user. Once these determinants are calculated, a fully connected layer perform the weighted sum defined in eq. (4).

The coefficients d_n in this expansion are treated as variational parameters, all initialized to 0 except for the ground-state determinant, which is assigned an initial weight of 1. In contrast to other approaches [25], QMCTorch does not require a preliminary calculation to determine the most relevant determinants to include in the expansion.

The Jastrow factor is computed using the original electronic coordinates based on the following expression [40]:

$$J(\mathbf{r}) = \sum_{i < j} K_{ee}(r_{ij}) + \sum_{\alpha, i} K_{en}(r_{\alpha i}) + \sum_{\alpha, i > j} K_{een}(r_{\alpha i}, r_{\alpha j}, r_{ij}) \quad (5)$$

where K_{ee} , K_{en} and K_{een} are the electron-electron, electron-nucleus and electron-electron-nucleus Jastrow kernels. Not all the terms in the Jastrow factor need to be included; for instance, the wavefunction ansatz can be restricted to contain only an electron–electron kernel. QMCTorch provides several built-in kernels, and users can easily define and test their own custom kernels. As with the backflow transformation, all necessary derivatives for computing the local energies can be obtained via automatic differentiation. This enables rapid experimentation with new forms of the Jastrow factor without requiring manual derivation of lengthy analytical expressions. The wave function, as defined in Eq. (4), is obtained as the product of the Jastrow factor and the sum over the Slater determinants.

2.2 Gradients of the Wave Function

The variational parameters of the wave function ansatz are optimized by minimizing the loss function: $\mathcal{L}(\theta) = 1/M \sum_m E_L(\mathbf{r}_m)$. The gradients of the loss function with respect to the variational parameters are computed using a low-variance estimator originally described in Ref. [41].:

$$\nabla_{\theta} \mathcal{L} = \frac{1}{M} \sum_m 2 \frac{\nabla_{\theta} \Psi_{\theta}(\mathbf{r}_m)}{\Psi_{\theta}(\mathbf{r}_m)} (E_L(\mathbf{r}_m) - E_v) \quad (6)$$

This expression only requires computing the derivatives of the wave function with respect to the variational parameters ($\nabla_{\theta} \Psi_{\theta}(\mathbf{r}_m)$) which can be efficiently obtained via automatic differentiation. This estimator avoids the need to differentiate the local energy itself. The evaluation of the local energies requires computing the diagonal elements of the Hessian of the wave function with respect to the electronic coordinates ($\nabla_i^2 \Psi(r)$). We employ an efficient numerical scheme which, in the absence of a backflow transformation, expresses the diagonal elements of the Hessian for both ground and excited state determinants as traces of matrix products [32]. We have extended this approach to accommodate backflow-transformed wave functions, significantly accelerating the computation of local energies (see Appendices F, G, and H).

2.3 Interatomic Forces

Interatomic forces can be calculated using the wave function ansatz depicted in Fig. 1 as the atomic positions are input parameters of the model. The interatomic forces are evaluated using the estimator [10]:

$$\mathcal{F}_{\alpha} = -\frac{1}{M} \sum_{m=1}^M \nabla_{\alpha} E_L(\mathbf{r}_m) + (E_L(\mathbf{r}_m) - E_v) \nabla_{\alpha} \ln \rho(\mathbf{r}_m) \quad (7)$$

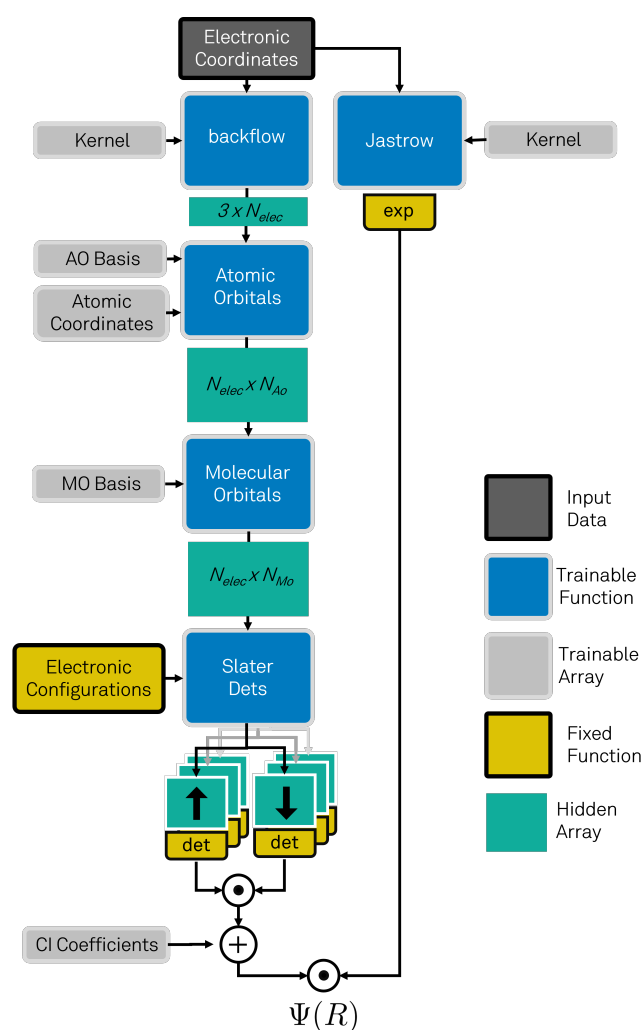


Figure 1: Representation of the wave function ansatz in QMCTorch. The architecture consists of multiple layers that transform electronic and atomic coordinates into a scalar wave function value. All layers are fully differentiable, enabling efficient optimization of the variational parameters via gradient-based methods. Additionally, certain layers support user-defined kernels, allowing for fine-tuning and customization of the ansatz.

that only requires the gradients calculation of the local energies ($\nabla_{\alpha} E_L$) and logarithmic density ($\nabla_{\alpha} \ln \rho(\mathbf{r}_m)$) with respect to the atomic coordinates. It is important to note, however, that this estimator exhibits infinite variance when applied to approximate wave functions, necessitating a large number of samples for reliable convergence. To address this limitation, regularization techniques have been proposed to improve the estimator's stability and efficiency, and their implementation is planned for future work [42, 43].

2.4 Computational Performance

Leveraging PyTorch's capabilities, QMCTorch can compute the wave function, energy, loss function gradients, and interatomic forces on both CPUs and GPUs. As demonstrated in Appendix A, GPU acceleration becomes effective only when the number of sampling points is sufficiently large. For instance, with 1000 sampling points for a CO molecule, the GPU run-

time is one to two orders of magnitude lower than that on a CPU (see Fig. 5). The runtime scales favorably with the number of determinants included in the wave function, owing to the efficient numerical schemes for evaluating ground and excited-state determinants. Notably, the computation of $\nabla_{\theta}\mathcal{L}$ and \mathcal{F}_{α} incurs only a marginal overhead compared to the evaluation of local energies alone, especially when the wave function includes a large number of determinants.

3 Results

We demonstrate the capabilities of QMCTorch on four small molecules: H_2 , LiH , Li_2 and CO . We use here a simple inverse backflow transformation given by the kernel: $K_{BF}(r_{ij}) = \omega/r_{ij}$ with ω the single variational parameter of this transformation. We use an electron-electron neural Jastrow factor whose kernel, $K_{ee}(r_{ij})$, is given by two fully connected feedforward neural networks - one for parallel-spin and one for antiparallel-spin electron pairs. Each network contains two hidden layers of 32 and 64 nodes respectively and uses a sigmoid as activation function. Additional details regarding the optimization of the wave functions are provided in Appendix B.

3.1 Wavefunction Optimization

Fig. 2 shows the correlation energy retrieved during the optimization of the four molecules using different wave function ansätze. SD-NJ: a single determinant with a neural Jastrow but no backflow; MD-NJ: multiple determinants with a neural Jastrow but no backflow; MD-NJ-iBF: multiple determinants with a neural Jastrow and an inverse backflow. For each ansatz, simulations were performed both with and without optimization of the atomic orbital exponents. The retrieved correlation energy is defined by: $\Delta E = 1 - (E_v - E_{HF})/(E_{\text{ref}} - E_{HF})$ with E_{HF} (E_{ref}) the Hartree-Fock(Reference) energies reported in Table 1 and E_v the variational energy obtained with QMCTorch.

As shown in 2, the results obtained with the three different ansätze give satisfying results on all four molecules. In most cases, the total energy initially lies above the Hartree-Fock energy but quickly converges toward the reference value during optimization. Depending on the molecule and ansatz, the optimization recovers between 50% and 99% of the correlation energy. Optimizing the atomic orbital exponents consistently improves the variational energy. Notably, a comparison of the first two columns in Fig. 2 reveals that the variational energy obtained with the SD-NJ ansatz and optimized AOs is comparable to that of the MD-NJ ansatz with frozen AOs. This suggests that computational frameworks currently lacking AO optimization could benefit from extending the variational space to include differentiable AOs, thereby enhancing the accuracy of the resulting wave function.

The neural Jastrow factors used in SD-NJ and MD-NJ ansätze are, in principle, capable of approximating any arbitrary functions. However, as illustrated in Fig. 6, their optimized form closely resemble a Padé-Jastrow factor, a widely used functional form in QMC simulations. As a result, replacing the neural Jastrow factor by a much simpler Padé-Jastrow factor yields qualitatively similar performance (see Fig. 7). It is interesting to note that the electron-electron cusp conditions of the neural Jastrow factors, i.e. $\frac{dJ(x)}{dx}\big|_{x=0}$, fluctuate around their theoretical limits during the optimization. While the neural Jastrow is more flexible than a Padé-Jastrow factor, and could in principle lead to lower variational energy values, it suffers from its inability to exactly respect these cusp conditions. Incorporating these conditions directly into the neu-

	E_{HF} (a.u.)	E_{ref} (a.u.)	$\Delta E_{\text{QMCTorch}}$	$\Delta E_{\text{PauliNet}}$	$\Delta E_{\text{Ferminet}}$
H_2	-1.133	-1.17447 [51]	99.12%	99.99%	-
LiH	-7.97	-8.07055 [52]	93.57%	99.30%	-
Li_2	-14.867	-14.9954 [53]	89.21%	-	99.47%
CO	-112.7871	-113.3225 [24]	63.47%	-	99.32%

Table 1: Ground state energy and fraction of the correlation energy recovered by QMCTorch, Paulinet [25] and Ferminet [24] for four different molecules.

ral architecture through hard constraints [44], may help overcome this limitation in the future.

229

230 The third column of Fig. 2 presents results obtained by augmenting the MD-NJ ansatz with
 231 an inverse backflow transformation. Consistent with previous findings [25], the inclusion of a
 232 backflow transformation introduces significant noise into the optimization process and gener-
 233 ally fails to improve performance. A similar trend is observed in Fig. 7, where a more flexible
 234 neural backflow transformation is used. Interestingly, the neural backflow in Fig. 7 tends to
 235 converge toward a $1/x$ functional form during the optimization (see Fig. 8). This suggests
 236 that while neural backflows offer flexibility, simpler forms may be sufficient to obtain simi-
 237 lar performance. A different approach consists of incorporating the backflow transformation
 238 through a differentiable envelope function, as implemented in Ref. [25] using the SchNet ar-
 239 chitecture [45]. This approach has demonstrated greater stability and is planned for future
 240 integration into QMCTorch.

241

242 The Jastrow factors employed in the preceding analysis include only electron-electron in-
 243 teraction terms. It is well established that incorporating many-body terms can significantly
 244 enhance the accuracy of variational energy estimates [40]. Figure 7 illustrates that, in our
 245 case, the addition of electron-nucleus and electron-electron-nucleus Jastrow kernels yields
 246 only a marginal improvement in the computed energy values. These results, along with those
 247 presented earlier, suggest that the limitations of our approach may not stem from the ex-
 248 pressiveness of the ansatz itself, but rather from the optimization strategy or the sampling
 249 techniques used to evaluate the local energies and the gradients of the loss function.

250

251 As shown in Table 1, the maximum correlation energy retrieved with QMCTorch remains
 252 lower than that obtained with Paulinet and Ferminet. However it is important to remember
 253 that the ansätze used here involved significantly fewer variational parameters - on the order
 254 of $\sim 10^2$ to 10^3 for QMCTorch - compared to $\sim 7 \cdot 10^4$ for Paulinet and $\sim 7 \cdot 10^5$ for Ferminet.
 255 Additionally, our optimization was explicitly limited to 500 steps, whereas Paulinet and Fer-
 256 minet used up to 7×10^3 and 10^5 steps, respectively. We also employed a simple ADAM [46]
 257 optimizer in contrast to the more advanced AdamW [47] and KFAC [48] optimizers used in
 258 Paulinet and Ferminet. Furthermore, our results may be constrained by the quality of the
 259 sampling procedure. We employed a standard Metropolis algorithm while more sophisticated
 260 variants are often adopted [49] and many studies refine the sampling points using diffusion
 261 Monte Carlo techniques [50].

262 3.2 Wavefunction and Energy Landscape of H_2

263 Fig. 3 shows the wavefunction and energy landscape of a H_2 molecule as its two electrons
 264 are independently moved along the molecular axis. As shown on this figure, the wavefunction
 265 naturally satisfies the electron-nucleus cusp conditions due to the use of STOs, resulting in

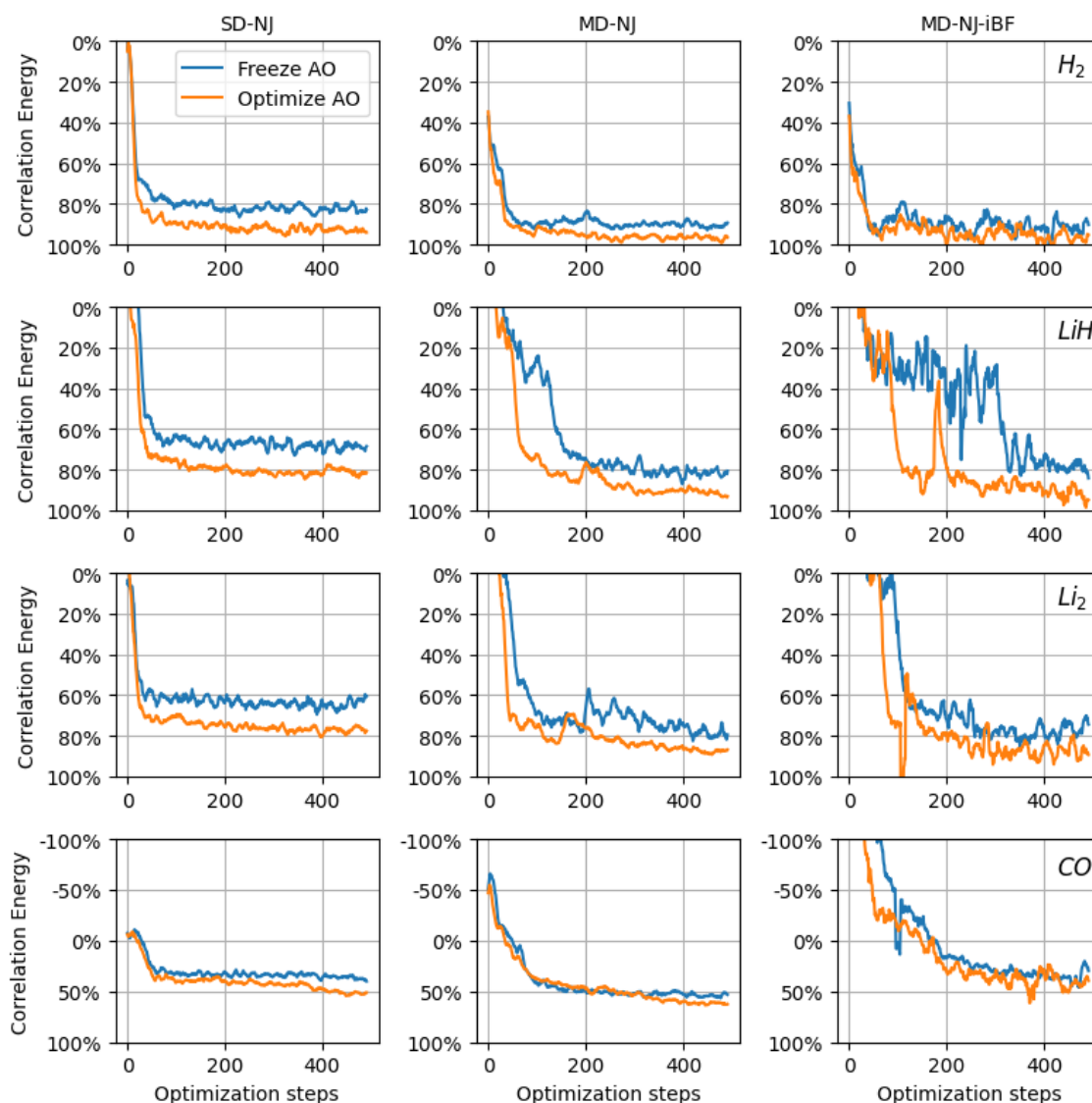


Figure 2: The y-axis indicates the fraction of the correlation energy recovered during the optimization process. The orange curves correspond to cases where all variational parameters were optimized, while the blue curves represent optimizations in which the exponents of the atomic orbitals were held fixed at their initial values. For clarity, all curves have been smoothed using a sliding window of size 10.

finite local energy values at the electron-nucleus coalescence points. One-dimensional profiles of the wavefunction and energy of LiH and Li_2 are provided in Fig. 9. These plots confirmed that the wavefunction respects the electron-electron and electron-nucleus cusp conditions, as well as the Fermi exclusion principle. The wavefunction profile obtained for LiH closely resembles that produced by Paulinet. However, the simpler nature of the ansatz used in our approach yields a smoother energy profile compared to Paulinet, albeit at the cost of reduced expressiveness.

3.3 Energy Curve and Interatomic Forces

To further assess the applicability of our approach, we have computed the dissociation energy curves of the four molecules studied here. The calculations were carried out using a multi-

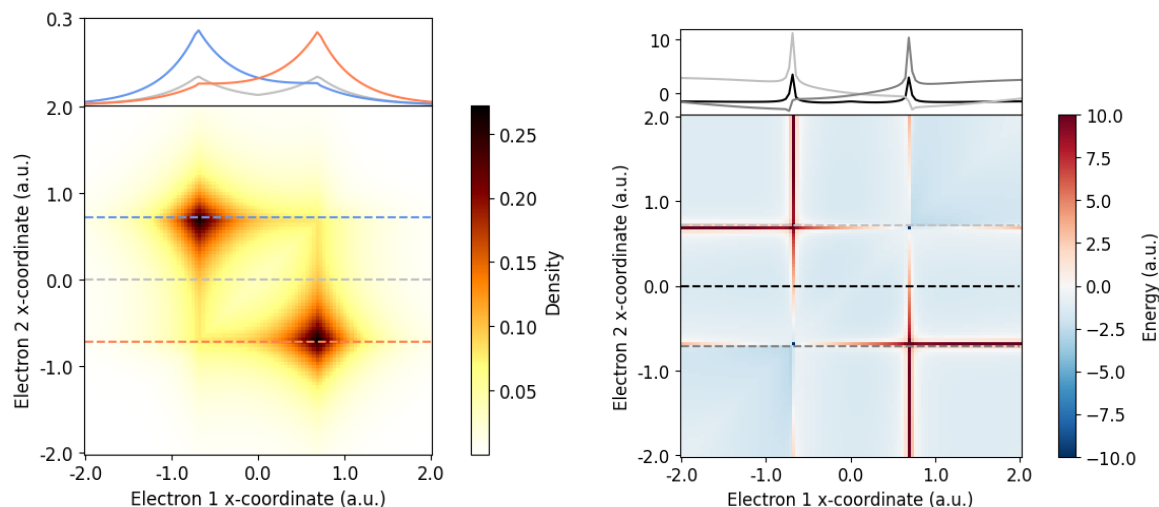


Figure 3: Values of $|\Psi(r_1, r_2)|^2$ (left) and $E(r_1, r_2)$ (right) of an optimized H_2 molecule as a function of the position of its two electrons along the axis of the molecule.

determinant wavefunction without any backflow transformation. Figure 4 presents the outcome of these simulations and compares them with simulations performed at the Hartree–Fock and coupled-cluster singles and doubles (CCSD) levels, using the PySCF package with a DZP basis set. As shown on this figure, the energy curves obtained with QMCTorch are in very good agreement with the CCSD results, even yielding slightly lower energies for H_2 , LiH and Li_2 . However, as previously observed in Figure 2, QMCTorch recovers only about 60% of the correlation energy for CO. Consequently, in this case, our results do not outperform those obtained with CCSD.

284

Figure 4 also displays the interatomic forces computed along the dissociation curves, as evaluated using Eq. (7). To compute these forces, the optimized wavefunction was resampled using 10^6 sampling points. These samples were divided in 100 batches and explicitly symmetrized around the molecular axis. The force component along the molecular axis (taken here as the z axis), defined as: $\Delta z = F_z^\alpha - F_z^\beta$, with F_z^α the force along the z -axis for atom α , was computed for each batch. The mean value and standard deviation of these values are shown in Fig. 4. As shown on this figure, the forces computed for H_2 and LiH are in very good agreements with those obtained at the CCSD level of theory. However, the accuracy of the forces predicted by QMCTorch decreases for Li_2 and CO , particularly at larger interatomic separations, where the standard deviation increases significantly.

294

4 Conclusion

295

In this paper, we presented results obtained with QMCTorch, our software for performing real-space quantum Monte Carlo (QMC) simulations of molecular systems using PyTorch. QMCTorch is built around a modular implementation of a backflow-transformed Slater–Jastrow wavefunction. Unlike many similar codes, QMCTorch supports the use of Slater-type orbitals and enables the optimization of atomic orbital exponents through a fully differentiable atomic-orbital layer. The software includes efficient numerical schemes for computing determinants and kinetic energy, allowing it to handle large numbers of determinants with relatively low

302

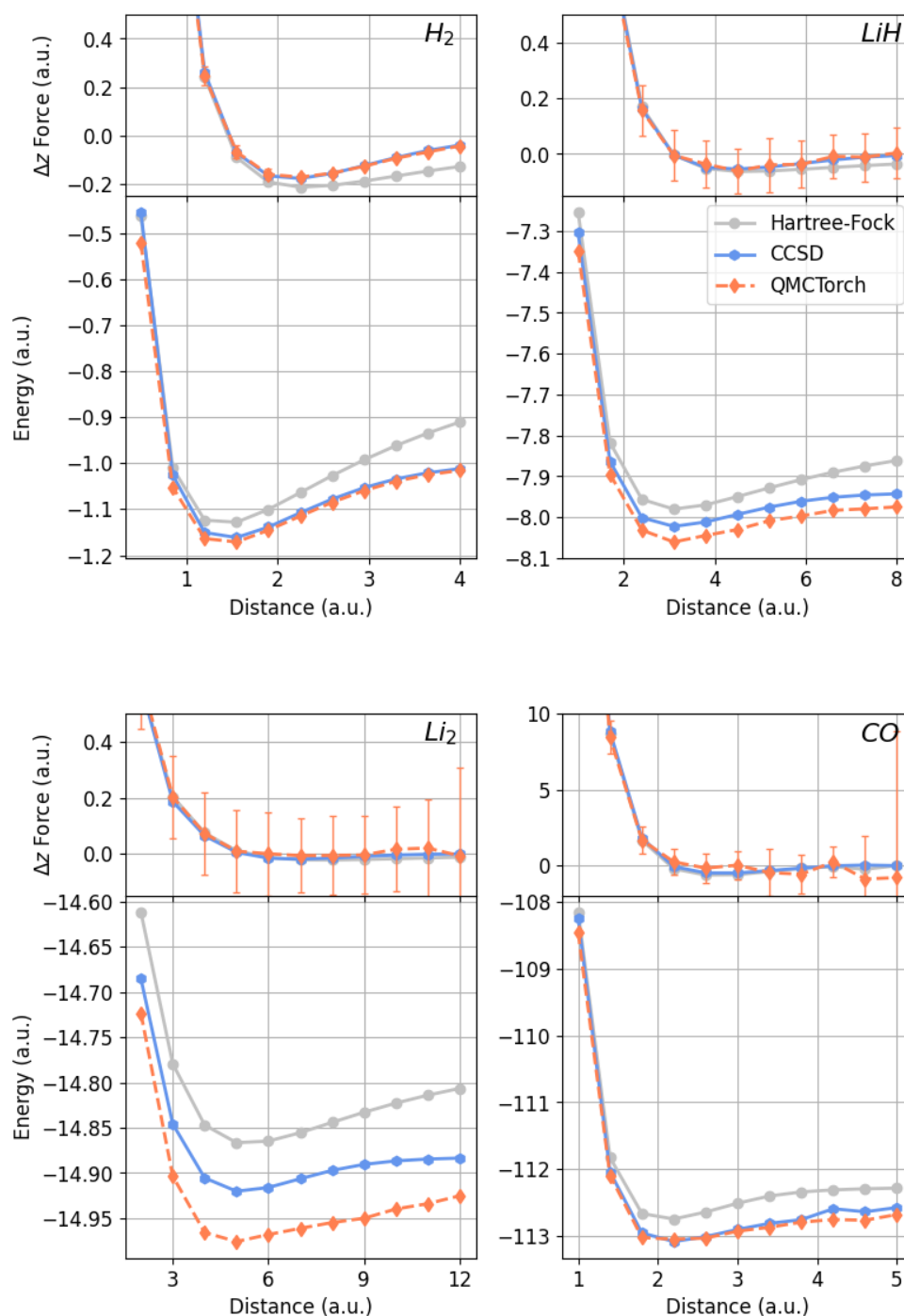


Figure 4: Dissociation energy curve and interatomic force (Δz) of four small molecules as a function of their interatomic distance computed with Hartree-Fock, CCSD and QMCTorch. Error bars on the interatomic forces represent the standard deviation across sampling batches.

303 computational cost.

304

305 The results obtained for four small molecules demonstrate that our approach is capable of
306 recovering a significant fraction of the correlation energy, despite the simplicity of the wave-
307 function ansatz. We have shown that optimizing the exponents of the atomic orbitals plays a

crucial role in improving wavefunction quality. Additionally, our findings indicate that while neural Jastrow factors enhance the flexibility of the ansatz, they do not significantly outperform the simpler, widely used Jastrow forms. Incorporating hard physical constraints into the architecture of neural Jastrow factors may further improve their effectiveness in future work. The inclusion of backflow transformations, as currently implemented in QMCTorch, led to noisy optimization and, consequently, a degradation in performance. Our results demonstrate that QMCTorch is able to produce accurate dissociation energy curves for all four molecules studied, underscoring the robustness of the approach. We also computed interatomic forces along these dissociation curves and found them to be in good agreement with CCSD reference values.

Currently, QMCTorch is limited to small molecular systems due to the absence of parallelization across multiple computing nodes. Extending the software to large-scale computing infrastructures through data parallelism is a key focus of future development. This enhancement will enable the simulation of larger systems, where QMC methods are often more efficient than alternative approaches [2]. Another major direction for future work is the integration of diffusion Monte Carlo (DMC) sampling techniques [54], which will allow for a more comprehensive assessment of the method’s capabilities when combined with neural wavefunctions [55, 56]. We also plan to extend QMCTorch to support the optimization of excited-state wavefunctions [7, 8, 57], a feature already available in other neural wavefunction frameworks [26–28].

Data Availability

QMCTorch is freely available under Apache 2.0 License at <https://github.com/QMCTorch/QMCTorch>. The input files used to run the simulations as well the output files and plotting scripts used for the different figures can be found at: <https://zenodo.org/records/15583384>

Acknowledgments

We would like to express our gratitude to Prof. Claudia Filippi for guidance throughout the development of the software and to Felipe Zapata for collaboration in the early stage of the project.

Funding information The development of the code was funded through the Joint NWO Computational Science and Energy Research & eScience Research Programme via the project *A Light in the Dark: quantum Monte Carlo meets solar energy conversion* grant ID: CSER.JCER.022.

A Computational Performance

Fig. 5 presents the GPU and CPU runtime for the calculation of the wave function (Ψ), local energies (E_L), gradients of the loss function ($\nabla_{\theta}\mathcal{L}$) and interatomic forces (\mathcal{F}_{α}) for a CO molecule. The calculations were performed using varying number of determinants and either 10 or 1000 sampling points. As shown in this figure, no GPU acceleration is observed when using 10 points; in fact CPU calculation can be faster in this case. However, with 1000 sampling points, the GPU demonstrates a clear advantage, achieving a speedup of approximately one

order of magnitude for the wave function computation and nearly two orders of magnitude for the energy, gradient, and force evaluations.

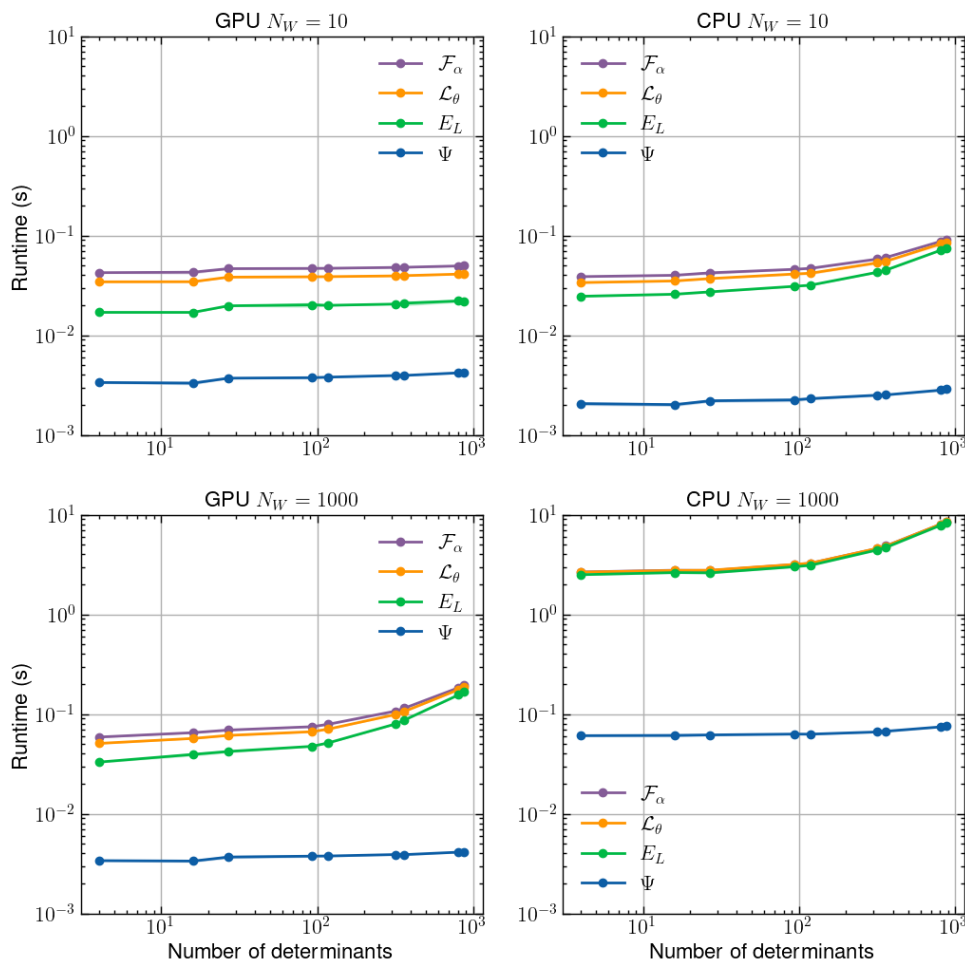


Figure 5: Runtime performance for computing the wave function (Ψ), local energies (E_L), gradients of the loss function ($\nabla_{\theta} \mathcal{L}$) and interatomic forces (\mathcal{F}_{α}) of a CO molecule on an Nvidia A100 GPU and an AMD Rome 7H12 CPU. The calculations employed an electron-electron Padé-Jastrow factor and an inverse backflow transformation. Simulations were conducted across various electronic configurations, ranging from SD(2,2) to SD(10,10), corresponding to 4 to 876 determinants. Results are shown for two sampling regimes: 10 points (top) and 1000 points (bottom).

B Parameters for the Wavefunction Optimization

We provide here additional details regarding the wavefunction optimization procedure employed for the four molecules studied in this article. Table 2 summarizes all parameters used during the optimization process.

Wavefunction The initial HF calculations were done using ADF [31] with a DZP basis set. Various electronic configurations were used in defining the wavefunction ansatz, ranging from SD(2,8) for H_2 to SD(8,12) for CO. Depending on the molecule and the chosen ansatz, the resulting wave functions comprised between 182 and 6906 variational parameters.

Sampling The sampling of $\rho(\mathbf{r})$ was done using a Metropolis-Hasting algorithm [58]. A total of 10^4 independent walkers were employed, each performing 10^4 Monte-Carlo (MC) steps. The walkers were moved using a normal proposal distribution with a standard deviation of 0.05 Bohr. All the electrons were moved simultaneously at each MC step. Only the final positions of the walkers were retained as sampling points for evaluating the local energies and gradients of the loss function.

Optimization The ADAM optimizer [46] was used to optimize the variational parameters using a learning rate of 10^{-2} and 500 optimization steps were performed. After each optimization step, all the sampling points were updated by resampling the probability distribution, using 500 MC steps.

Parameter	Value(s)
SCF Package	ADF
Basis	DZP
Electronic configuration	
H_2	SD(2,8) - (64 dets)
LiH	SD(4,12) - (531 dets)
Li_2	SD(4,12) - (531 dets)
CO	SD(8,12) - (1425 dets)
Number walkers	10^4
Number of initial MC steps	10^4
Sampling step size	0.05
Resample Frequency	1
Number of resampling MC steps	500
Optimizer	ADAM
Learning rate	10^{-2}
Number of optimization steps	500
Number of variational parameters	
H_2	low: 182 - high: 4662
LiH	low: 857 - high: 5343
Li_2	low: 1189 - high: 5669
CO	low: 2425 - high: 6905

Table 2: Parameters used for the wavefunction optimizations shown in Fig. 2 and 7. The lower and upper bounds on the number of variational parameters correspond to the e-e Pade-Jastrow (Fig. 7) and MD-NJ (Fig. 2) ansatzes respectively.

C Neural Electron-Electron Jastrow Factor

The left panel in Fig. 6 shows the optimized form of the neural Jastrow factor obtained with a MD-NJ wavefunction ansatz depicted in Fig. 2. For clarity, these functions were normalized to their value at $x = 0$. While neural Jastrow factors are, in principle, capable of approximating arbitrary functions, they exhibit a form similar to that of the Padé–Jastrow factor (see Equation D.1).

The right panel in Fig. 6 illustrates the evolution of the electron-electron cusp of the

Jastrow factor, defined as: $\frac{dJ(x)}{dx}|_{x=0}$, throughout the optimization process. It is interesting to note that in most cases, the neural Jastrow factor converges toward the correct electron-electron cusp conditions given by [59]:

$$\left. \frac{dJ(x)}{dx} \right|_{x=0} = \frac{1}{4} \quad \text{for parallel spins and} \quad \left. \frac{dJ(x)}{dx} \right|_{x=0} = \frac{1}{2} \quad \text{for opposite spins} \quad (\text{C.1})$$

However, these constraints are not explicitly enforced in the design of the neural Jastrow factor. As a result, there is no guarantee that they will be approached—let alone satisfied—by the optimized Jastrow function.

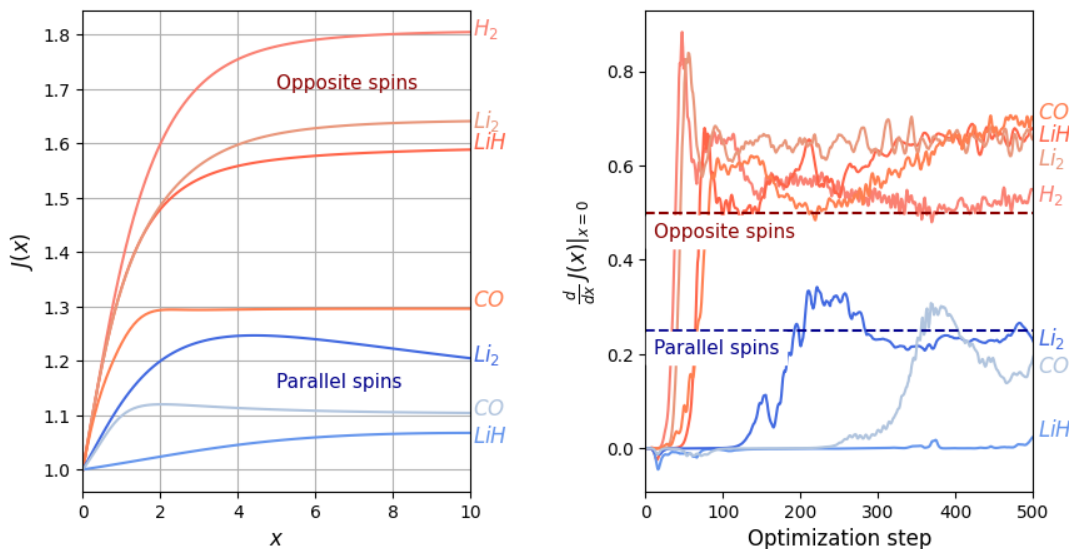


Figure 6: Left - Optimized form of the neural Jastrow factors. Right - Evolution of the electron-electron cusp conditions for the neural Jastrow factor throughout the optimization process. The dashed lines represent the theoretical values of 0.25(0.5) for parallel(opposite)-spins electron pairs.

D Additional Wavefunction Ansatz

Fig. 7 presents the correlation energy obtained with three different wavefunction ansatzes for the four molecules studied here.

D.1 Electron-Electron Padé Jastrow Factor

In the first wave function considered here, we employed a simple electron-electron Padé-Jastrow factor defined as:

$$K_{ee}(r_{ij}) = \frac{\omega_0 r_{ij}}{1 + \omega r_{ij}} \quad (\text{D.1})$$

where $\omega_0 = 0.25(0.5)$ for parallel(antiparallel)-spin electron pairs and ω a variational parameter. The fraction of the correlation energy recovered using this Jastrow factor is represented in the first column of Fig. 7. As shown in this figure, the results are not significantly different from those obtained with the neural Jastrow factor and presented in Fig. 2.

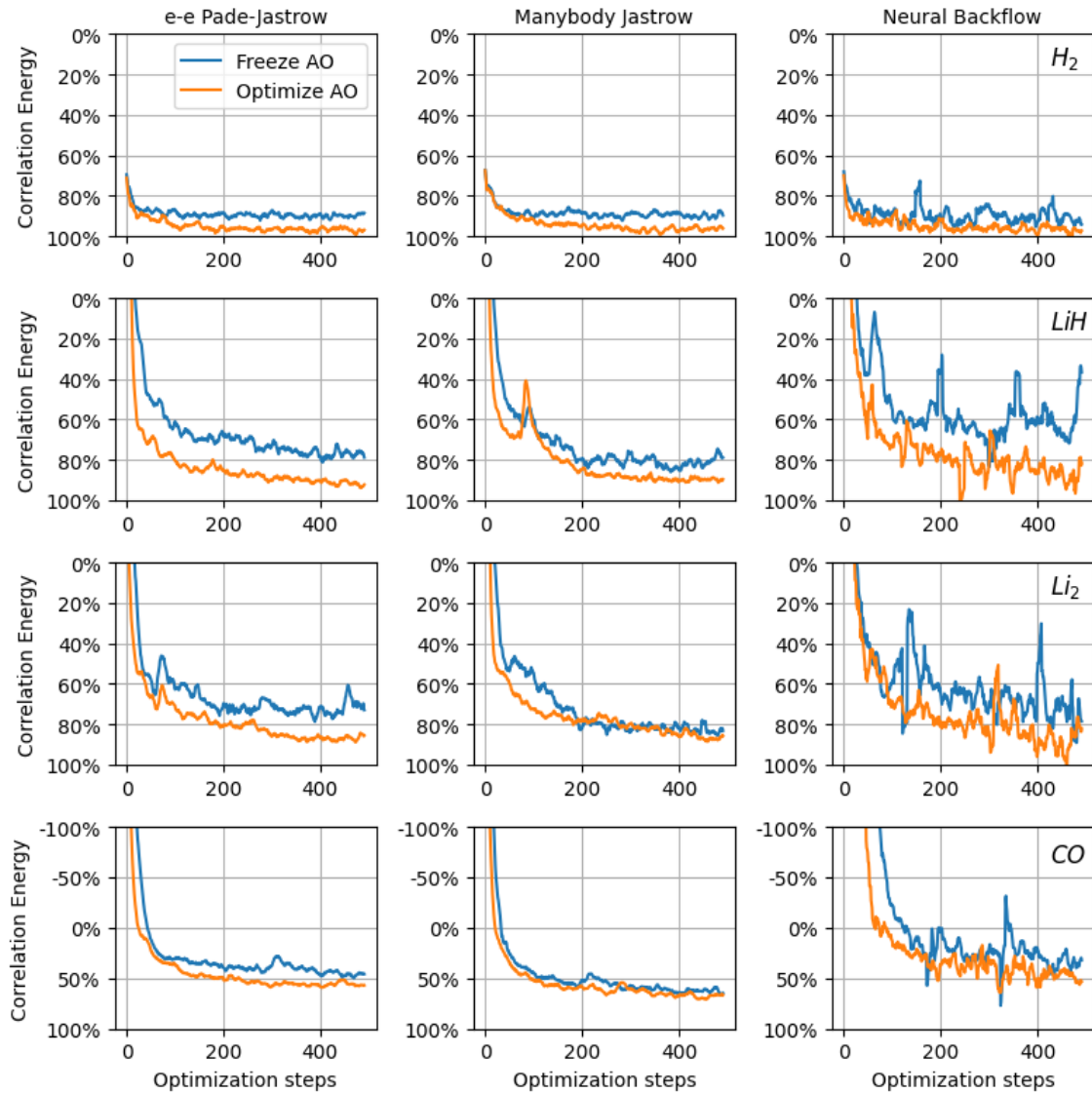


Figure 7: Optimization of the wave function of small molecules: H_2 , LiH , Li_2 and CO using different ansatzes (see text for details)

D.2 Many-body Jastrow Factor

A many-body Jastrow factor was incorporated in the second wavefunction ansatz considered here. In addition to the electron-electron Padé-Jastrow kernel, we introduced an electron-nucleus kernel given by [60]:

$$K_{en}(r_{ai}) = \frac{\omega_0 r_{ai} + \omega_1 r_{ai}^2}{1 + \omega_2 r_{ai}} \quad (D.2)$$

with ω_0 , ω_1 and ω_2 variational parameters and r_{ai} the distance between atom α and electron i . We also introduce an electron-electron-nucleus kernel given by:

$$K_{een}(r_{ai}, r_{aj}, r_{ij}) = \sum_{\mu=1}^N c_{\mu} \left(\frac{a_{1\mu} r_{ai}}{1 + b_{1\mu} r_{ai}} \right) \left(\frac{a_{2\mu} r_{aj}}{1 + b_{2\mu} r_{aj}} \right) \left(\frac{a_{3\mu} r_{ij}}{1 + b_{3\mu} r_{ij}} \right) \quad (D.3)$$

where the coefficients $a_{n\mu}$, $b_{n\mu}$, c_{μ} are variational parameters. We used here $N = 5$ terms in the expansion. The fraction of the correlation energy obtained with this Jastrow factor

is represented in the second column of Fig. 7. As seen in this figure, the introduction of many-body Jastrow factor does not significantly changes the results obtained with an electron-electron Jastrow factor as shown in Fig. 2.

D.3 Neural Backflow Transformation

The third wavefunction examined here, we used a simple Padé-Jastrow electron-electron factor but complemented the ansatz with a neural backflow transformation based on radial basis function network [61]. The kernel of this backflow transformation can be expressed as:

$$K_{BF}(r_{ij}) = \sum_{n=1}^{N_{RBF}} \omega_n \mathcal{N}_{\mu_n, \sigma_n}(r_{ij}) \quad (D.4)$$

with \mathcal{N} the normal function of mean μ_n and std σ_n . ω_n , μ_n and σ_n are here a variational parameters. We used here $N_{RBF} = 10$ basis function in the expansion. The fraction of the correlation energy recovered using this Jastrow factor is shown in the third column of Figure 7. As illustrated, incorporating a neural backflow yields results that are qualitatively similar to those presented in Figure 2. However, the optimization process remains challenging, preventing the correlation energy from fully stabilizing.

The evolution of the backflow kernel $K_{BF}(r_{ij})$ during the optimization is shown in Fig. 8. As seen there, the values of the kernel function converges toward a form that is proportional to $1/r_{ij}$. This explain the limited impact that this neural backflow has on the final results compared to the ones obtained with an inverse backflow kernel.

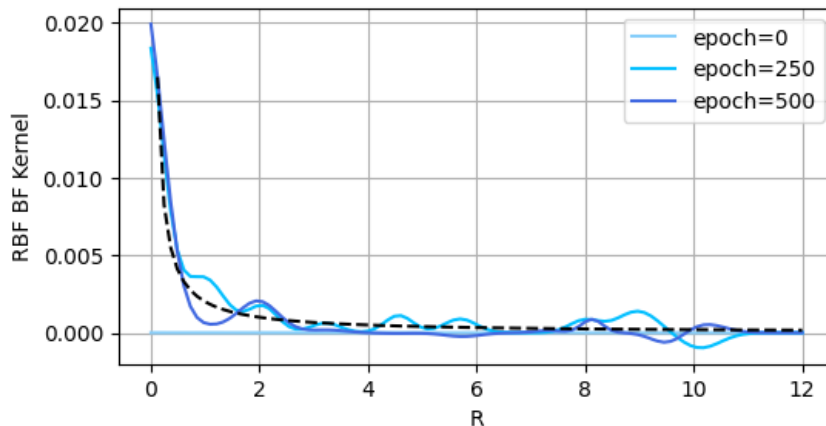


Figure 8: Evolution of the RBF neural backflow kernel during the optimization of the wavefunction.

E Wavefunction and Energy Profile of LiH and Li_2

Fig. 9 shows the wavefunction and energy profile of LiH and Li_2 , before and after the optimization of the wavefunction. The x-axis corresponds to the location of a spin-up electron that moves along the molecular axis (taken here as the z-axis). The positions of the nuclei and static electrons are marked on the figure. For LiH , the final spin-down electron is located at $(x, y, z) = (0.5, 0.5, 0.0)$, to enable direct comparison with the results reported in [25]. As shown on this figure, the wavefunction vanishes when two electrons with the same spin

coincide, in accordance with the Pauli exclusion principle. When two electrons of opposite spin occupy the same position, the wavefunction presents the correct electron-electron cusp behavior. Additionally, the electron-nucleus cusp conditions are well satisfied, due to the use of STOs in the calculation.

The wavefunction profile of LiH closely resembles that obtained with Paulinet [25]. Interestingly, the wave function undergoes only minor changes during the optimization process, underscoring the subtle yet essential adjustments required to reach the variational minimum. The energy profile obtained with QMCTorch is noticeably smoother and more regular than that reported in Ref. [25]. This can be attributed to the simpler wavefunction ansatz employed in this work. However, as previously discussed, this simplicity also limits the ansatz's flexibility, which in turn constrains its ability to fully capture the true variational minimum

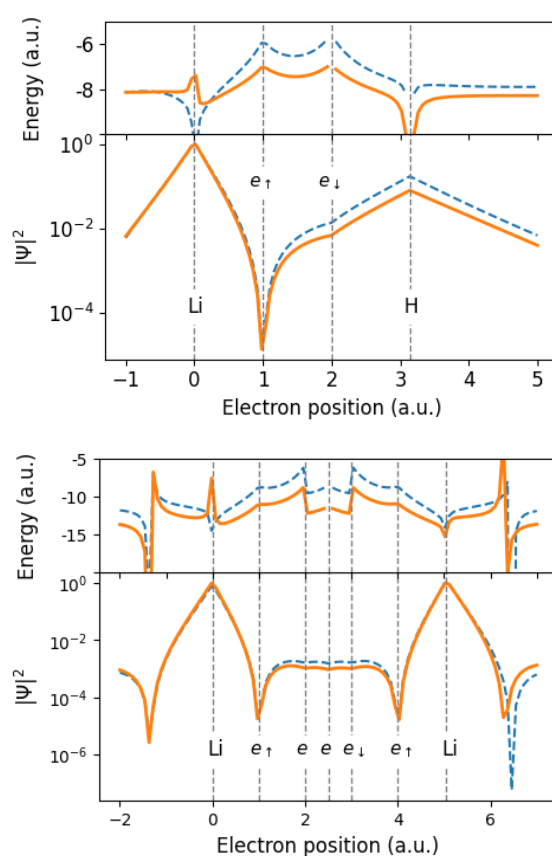


Figure 9: Wavefunction and energy profile of LiH (top) and Li_2 (bottom) with respect to the position of a spin-up electron moved along the molecular axis. The dashed line represents the original wavefunction while the orange plain line represents the optimized wavefunction. The positions of the atomic centers and static electrons are marked in the figure for clarity.

F Computation of the Local Energies of Backflow-Transformed Wavefunctions

While all the derivatives of the wave function and the loss function could be obtained directly via automatic differentiation, we exploit here several numerical schemes to decrease the com-

putational and memory requirement of the calculations.

F.1 Derivative of a Single Determinant

The local energy $E_L(\mathbf{r})$ requires the calculation of the kinetic energy $\frac{\nabla^2 \Psi(\mathbf{r})}{\Psi(\mathbf{r})}$. Back-propagating twice through the wave function via automatic differentiation can evaluate these terms but this approach is computationally costly. To alleviate this issue we have implemented the results obtained by Filippi et. al. [32] and extended them to the case of backflow transformed wave function.

Setting $\Psi(\mathbf{r}) = \mathcal{J}(\mathbf{r})\Sigma(\mathbf{r})$, where $\Sigma(\mathbf{r}) = \sum_n d_n D_n^\uparrow D_n^\downarrow$, the kinetic energy can be written as:

$$\frac{\nabla^2 \Psi(\mathbf{r})}{\Psi(\mathbf{r})} = \frac{\nabla^2 \mathcal{J}(\mathbf{r})}{\mathcal{J}(\mathbf{r})} + 2 \frac{\nabla \mathcal{J}(\mathbf{r})}{\mathcal{J}(\mathbf{r})} \frac{\nabla \Sigma(\mathbf{r})}{\Sigma(\mathbf{r})} + \frac{\nabla^2 \Sigma(\mathbf{r})}{\Sigma(\mathbf{r})} \quad (\text{F.1})$$

The derivatives of the Jastrow factor can be obtained given its functional form, either by providing analytical expression for the derivatives of the different kernels or by relying on automatic differentiation. Since the Jastrow kernels are $\mathbb{R} \rightarrow \mathbb{R}$ maps automatic evaluation of the second derivative via double backpropagation does not lead to the calculation of unnecessary mixed second derivative terms. The gradient of the determinant part can be expressed as :

$$\nabla \Sigma = \sum_n d_n \left(\frac{\nabla D_n^\uparrow}{D_n^\uparrow} + \frac{\nabla D_n^\downarrow}{D_n^\downarrow} \right) D_n^\uparrow D_n^\downarrow \quad (\text{F.2})$$

where the gradient of the individual determinant terms can be obtained via the Jacobi formula [62]:

$$\frac{\nabla D}{D} \equiv \frac{\nabla \det A}{\det A} = \text{Tr}(A^{-1} \nabla A) \quad (\text{F.3})$$

The diagonal hessian of the determinant part Σ is obtained via:

$$\nabla^2 \Sigma = \sum_n d_n \left(\frac{\nabla^2 D_n^\uparrow}{D_n^\uparrow} + 2 \frac{\nabla D_n^\uparrow}{D_n^\uparrow} \frac{\nabla D_n^\downarrow}{D_n^\downarrow} + \frac{\nabla^2 D_n^\downarrow}{D_n^\downarrow} \right) D_n^\uparrow D_n^\downarrow \quad (\text{F.4})$$

Here the diagonal Hessian of the determinant terms can be obtained via the formula [62]:

$$\frac{\nabla^2 D}{D} \equiv \frac{\nabla^2 \det(A)}{\det(A)} = \text{Tr}(A^{-1} \nabla^2 A) + (\text{Tr}(A^{-1} \nabla A))^2 - \text{Tr}(A^{-1} \nabla A A^{-1} \nabla A) \quad (\text{F.5})$$

Note that in the case of single-electron orbital, i.e. in absence of backflow transformation, the last two terms of eq. (F.5) cancel each other leading to : $\frac{\nabla^2 \det(A)}{\det(A)} = \text{Tr}(A^{-1} \nabla^2 A)$ as demonstrated in [32] (see appendix G).

The use of the eq. (F.5) (or G.7 in absence of backflow transformation), leads to a significant speed up of the calculation of the kinetic energy. As seen in Fig. 10, a speedup of almost two orders of magnitude can be obtained for CO. Even larger speedups are expected for larger molecules as the automatic differentiation approach scales poorly with the number of electrons in the molecule.

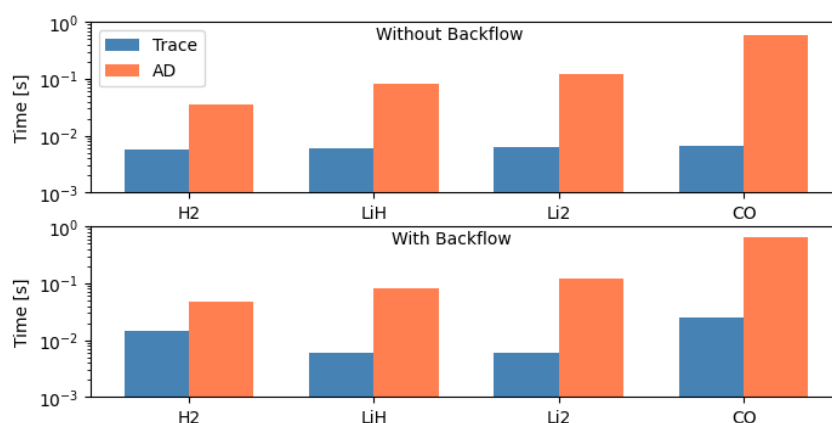


Figure 10: Runtime for the calculation of the kinetic energy using automatic differentiation (AD) or the matrix trace formulas (Trace). The runtime was computed on an NVIDIA A100 for four different molecules, using only the ground state in the wavefunction expansion, 1000 sampling points and with/without backflow transformation using eqs. (F.5) and (G.7) respectively

468 F2 Derivative of Excited-State Determinants

469 Calculating the first and second determinant derivatives given by eqs. (F.2) and (F.4) requires
 470 the inversion of a large number of A matrices corresponding to the different determinants. As
 471 the number of determinants increases, this becomes a computational bottleneck. The determi-
 472 nants included in the wave function (4) correspond to well defined electronic configurations.
 473 The determinants contained in the wave function only differ from the ground state determi-
 474 nant by the exchange of a finite number of columns (1 for single-electron excitations, 2 for
 475 double-electron excitations etc ...). This relationship allows to avoid the calculation of many
 476 matrix inverses.

477
 478 Let's consider a reference determinant $D_0 = \det A_0$, an excited determinant $D_X = \det A_X$,
 479 a 1 body operator \mathcal{O} (e.g. ∇ or ∇^2) with $B = \mathcal{O}A$. The matrix of molecular orbitals can be
 480 written as $\Phi = [A_0|A_v]$ where A_v is the rectangular matrix of the virtual molecular orbitals.
 481 The matrix A_X is consequently constructed by replacing some columns of A_0 by columns from
 482 A_v . One can then show that [32]:

$$\text{Tr}(A_X^{-1}B_X) = \text{Tr}(A_0^{-1}B_0) + \text{Tr}((PA_0^{-1}A_XP)^{-1} \cdot PMP) \quad (\text{F.6})$$

483 with:

$$M = A_0^{-1}B_X - A_0^{-1}B_0A_0^{-1}A_X \quad (\text{F.7})$$

484 Similarly one can show that (see Appendix H):

$$\begin{aligned} \text{Tr}\left((A_X^{-1}B_X)^2\right) &= \text{Tr}\left((A_0^{-1}B_0)^2\right) \\ &+ \text{Tr}\left(\left((PA_0^{-1}A_XP)^{-1} \cdot PMP\right)^2\right) \\ &+ 2\text{Tr}\left((PA_0^{-1}A_XP)^{-1} \cdot PYP\right) \end{aligned} \quad (\text{F.8})$$

485 with :

$$Y = A_0^{-1}B_0M \quad (\text{F.9})$$

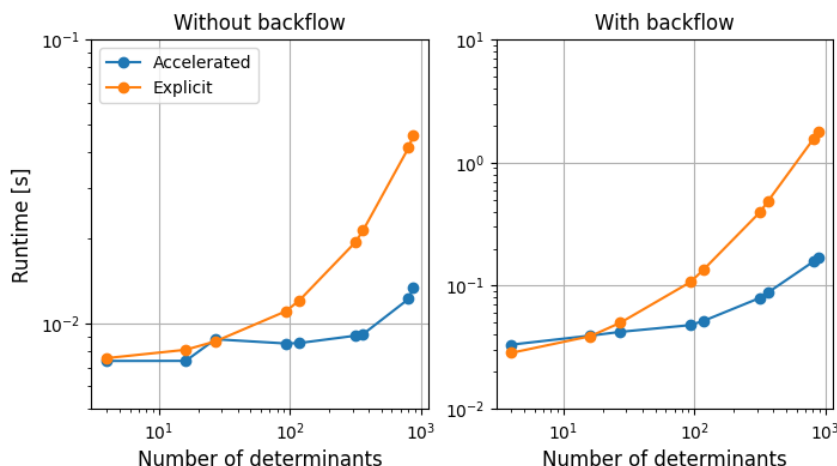


Figure 11: Runtime for the calculation of the kinetic energy of a CO molecule using 1000 walkers and different electronic configurations. The calculations were done on an NVIDIA A100 GPU and with and without backflow transformation. Two approaches are compared: Explicit - using eq (F.5) (or (G.7) without backflow) on all the determinants in the wave function; Accelerated - using eqs. (F.6) and (F.8) to compute the contribution of the excited state determinants of the wavefunction.

and P the projector on the space of columns which are different in A_0 and A_x [32]. Note that in eq. F.8, the convention: $(AB)^2 = ABAB$ was used to shorten the expressions. As seen on the equations (F.6) and (F.8), A_0 is the only large matrix whose inverse needs to be computed. The other inverses, namely $(PA_0^{-1}A_xP)^{-1}$, are in our case only of dimension 1 or 2 as we only consider single and double excitations. This greatly accelerate the calculations of the different terms present in eq. (F.2) and (F.4).

As shown in Fig. 11, using eqs. (F.6) and (F.8) to compute the kinetic energy leads to a one order of magnitude speedup compared to computing all the determinants explicitly. Note however that if only a small number of determinants are included in the wavefunction expansion, the explicit calculation might be faster or at least competitive with the accelerated method.

G Simplification of Eq. F.5 in Absence of Backflow Transformation

As explained in the main text, eq. F.5 reduces to the expression derived by Filippi et al. [32] in the case of single-electron orbitals, i.e. $A_{i,j} = \phi_{k_i}(r_j)$. In such a case the derivative of A w.r.t a given electronic position is given by:

$$\frac{\partial A}{\partial x_j} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ \frac{\partial \phi_{k_1}(r_j)}{\partial x_j} & \dots & \frac{\partial \phi_{k_n}(r_j)}{\partial x_j} \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix} \quad (\text{G.1})$$

We therefore have:

$$\text{Tr} \left(A^{-1} \frac{\partial A}{\partial x_j} \right) = \text{Tr}(uv^T) = v^T \cdot u \equiv c \quad (\text{G.2})$$

where u and v are two column vectors representing respectively the j -th column of A^{-1} and v the j -th row of $\frac{\partial A}{\partial x_j}$. We therefore have:

$$\left(\text{Tr} \left(A^{-1} \frac{\partial A}{\partial x_j} \right) \right)^2 = c^2 \quad (\text{G.3})$$

Similarly we can show that:

$$\text{Tr} \left(A^{-1} \frac{\partial A}{\partial x_j} A^{-1} \frac{\partial A}{\partial x_j} \right) = \text{Tr}(u(v^T u)v^T) = c \cdot \text{Tr}(uv^T) = c^2 \quad (\text{G.4})$$

Combining the last two equations we have:

$$\left(\text{Tr} \left(A^{-1} \frac{\partial A}{\partial x_j} \right) \right)^2 = \text{Tr} \left(A^{-1} \frac{\partial A}{\partial x_j} A^{-1} \frac{\partial A}{\partial x_j} \right) \quad (\text{G.5})$$

We can easily extend that results to show that:

$$(\text{Tr}(A^{-1} \nabla A))^2 = \text{Tr}(A^{-1} \nabla A A^{-1} \nabla A) \quad (\text{G.6})$$

As a consequence, for single-electron molecular orbital the two last terms of eq. F5 cancel each other and one obtain the result of Filippi et al [32]:

$$\frac{\nabla^2 D}{D} = \text{Tr}(A^{-1} \nabla^2 A) \quad (\text{G.7})$$

H Derivation of Eqs. F.6 and F.8

Equation F.6 has already been demonstrated by Filippi et al. [32]. We demonstrate it again following a slightly different method to facilitate the demonstration of eq. F.8.

Let's define A_0 and A_X as the molecular orbital matrix of the ground state determinant and of a given excited state determinant respectively. The two matrices only differ by a few columns of A_0 that have been replaced by column vectors of virtual orbitals. We also define the projector P on the subspace of these exchanged columns. P is a diagonal matrix with ones at the indices of the exchanged columns and zeros elsewhere. We also define the complementary projector Q such as $P + Q = \mathbb{I}$. Finally we define the matrices B_0 and B_X as the application of a 1-body operator (either ∇ or ∇^2) on A_0 and A_X .

As shown in [32] $A_X(B_X)$ can be expressed as :

$$A_X = A_0 + (A_X - A_0)P \quad (\text{H.1})$$

and

$$B_X = B_0 + (B_X - B_0)P \quad (\text{H.2})$$

524 H.1 Inverse of A_X

525 Equations F.6 and F.8 involve the inverse of the excited state matrix A_X . Using eq. H.1 we can
526 write this inverse as:

$$A_X^{-1} = (A_0 + (A_X - A_0)P)^{-1} \quad (\text{H.3})$$

527 We can use the Kailath variant of the Woodbury matrix inverse formulas [62]:

$$(U + VW)^{-1} = U^{-1} - U^{-1}V(I + WU^{-1}V)^{-1}WU^{-1} \quad (\text{H.4})$$

528 with $U = A_0$, $V = (A_X - A_0)$ and $W = P$ to express the inverse of A_X :

$$A_X^{-1} = A_0^{-1} - A_0^{-1}(A_X - A_0)(\mathbb{I} + PA_0^{-1}(A_X - A_0))^{-1}PA_0^{-1} \quad (\text{H.5})$$

$$= A_0^{-1} - (A_0^{-1}A_X - \mathbb{I})(\mathbb{I} + PA_0^{-1}A_X - P)^{-1}PA_0^{-1} \quad (\text{H.6})$$

$$= A_0^{-1} - (A_0^{-1}A_X - \mathbb{I})(Q + PA_0^{-1}A_X)^{-1}PA_0^{-1} \quad (\text{H.7})$$

$$= A_0^{-1} - (A_0^{-1}A_X - \mathbb{I})(Q + PA_0^{-1}A_XP)^{-1}PA_0^{-1} \quad (\text{H.8})$$

529 The equation H.8 is obtained by noting that following H.1 we have: $A_0^{-1}A_X = A_0^{-1}A_XP$ and
530 therefore: $PA_0^{-1}A_X = PA_0^{-1}A_XP$.

531

532 The term $(Q + PA_0^{-1}A_XP)$ in eq. H.8 is a block matrix on the subspaces spanned by the P
533 and Q and its inverse can therefore be written as [62]

$$(Q + PA_0^{-1}A_XP)^{-1} = Q + (PA_0^{-1}A_XP)^{-1} = Q + P(A_0^{-1}A_X)^{-1}P \quad (\text{H.9})$$

534 Note that the last equality is only valid since $QA_0^{-1}A_XQ = Q$. Since $QP = 0$ and $P^2 = P$ by
535 definition, eq. H.8 can then be simplified to:

$$A_X^{-1} = A_0^{-1} - (A_0^{-1}A_X - \mathbb{I})(P(A_0^{-1}A_X)^{-1}P)A_0^{-1} \quad (\text{H.10})$$

536 In the following we use the following definitions:

$$T = (P(A_0^{-1}A_X)^{-1}P) \quad (\text{H.11})$$

537 and

$$Z = (A_0^{-1}A_X - \mathbb{I})TA_0^{-1} \quad (\text{H.12})$$

538 H.2 Demonstration of eq. F.6

539 We are trying here to find a convenient expression for:

$$\text{Tr}(A_X^{-1}B_X) \quad (\text{H.13})$$

540 that does not require to compute the inverse of all the possible A_X matrices contained in
541 the CI expansion considered in the definition of the wave function. Using eq. H.10, equation
542 H.13 can therefore be written as :

$$\begin{aligned} \text{Tr}(A_X^{-1}B_X) &= \text{Tr}(A_0^{-1}B_X) - \text{Tr}((A_0^{-1}A_X - \mathbb{I})TA_0^{-1}B_X) \\ &= \text{Tr}(A_0^{-1}B_X) - \text{Tr}(A_0^{-1}A_XTA_0^{-1}B_X) + \text{Tr}(TA_0^{-1}B_X) \end{aligned}$$

Using eq. H.2 and the cyclic property of the trace, we can further obtain:

$$\begin{aligned}\mathrm{Tr}(A_X^{-1}B_X) &= \mathrm{Tr}(A_0^{-1}B_0) + \mathrm{Tr}(A_0^{-1}(B_X - B_0)P) \\ &- \mathrm{Tr}(TA_0^{-1}B_0A_0^{-1}A_X) \\ &- \mathrm{Tr}(TA_0^{-1}(B_X - B_0)PA_0^{-1}A_X) \\ &+ \mathrm{Tr}(TA_0^{-1}B_X)\end{aligned}$$

The fourth term of this equation can be simplified since $PA_0^{-1}A_X = PA_0^{-1}A_XP$ and as demonstrated in [32]

$$(P(A_0^{-1}A_X)P)(P(A_0^{-1}A_X)^{-1}P) = P \quad (\text{H.14})$$

Consequently the forth term can be simplified to

$$\begin{aligned}\mathrm{Tr}((P(A_0^{-1}A_X)^{-1}P)A_0^{-1}(B_X - B_0)PA_0^{-1}A_X) &= \mathrm{Tr}((P(A_0^{-1}A_X)^{-1}P)A_0^{-1}(B_X - B_0)PA_0^{-1}A_XP) \\ &= \mathrm{Tr}(A_0^{-1}(B_X - B_0)PA_0^{-1}A_XP(P(A_0^{-1}A_X)^{-1}P)) \\ &= \mathrm{Tr}(A_0^{-1}(B_X - B_0)P)\end{aligned}$$

and can be simplified with the second term of the equation. Factorizing the terms we arrive at the following equation:

$$\begin{aligned}\mathrm{Tr}(A_X^{-1}B_X) &= \mathrm{Tr}(A_0^{-1}B_0) \\ &+ \mathrm{Tr}((P(A_0^{-1}A_X)^{-1}P)(A_0^{-1}B_X - A_0^{-1}B_0A_0^{-1}A_X))\end{aligned}$$

H.3 Demonstration of eq. F.8

We are trying here to derive a convenient expression for:

$$\mathrm{Tr}(A_X^{-1}B_XA_X^{-1}B_X) \equiv \mathrm{Tr}((A_X^{-1}B_X)^2) \quad (\text{H.15})$$

Using eq. H.10, i.e. $A_X^{-1} = A_0^{-1} - Z$, we can unroll the square to obtain:

$$\mathrm{Tr}((A_X^{-1}B_X)^2) = \underbrace{\mathrm{Tr}((A_0^{-1}B_X)^2)}_{\alpha} + \underbrace{\mathrm{Tr}((ZB_X)^2)}_{\gamma} - 2 \underbrace{\mathrm{Tr}(A_0^{-1}B_XZB_X)}_{\beta} \quad (\text{H.16})$$

We start by injecting eq. H.2 in the first term of the equation above to obtain:

$$\mathrm{Tr}((A_0^{-1}B_X)^2) = \underbrace{\mathrm{Tr}((A_0^{-1}B_0)^2)}_{\alpha_1} + \underbrace{\mathrm{Tr}((A_0^{-1}B_\Delta)^2)}_{\alpha_2} + 2 \underbrace{\mathrm{Tr}(A_0^{-1}B_\Delta A_0^{-1}B_0)}_{\alpha_3} \quad (\text{H.17})$$

with $B_\Delta = (B_X - B_0)P = B_X - B_0$. Using the definition of Z , the second term of eq. H.16 leads to:

$$\mathrm{Tr}((ZB_X)^2) = \underbrace{\mathrm{Tr}((A_0^{-1}A_XTA_0^{-1}B_X)^2)}_{\gamma_1} + \underbrace{\mathrm{Tr}((TA_0^{-1}B_X)^2)}_{\gamma_2} - 2 \underbrace{\mathrm{Tr}((TA_0^{-1}B_X)^2A_0^{-1}A_X)}_{\gamma_3} \quad (\text{H.18})$$

555 Using the definition of B_X , the cyclic property of the trace and the identity: $PA_0^{-1}A_XT = P$
 556 this can be simplified to:

$$\gamma_1 = \underbrace{\text{Tr}((A_0^{-1}A_XTA_0^{-1}B_0)^2)}_{\gamma_{11}} + \underbrace{\text{Tr}((A_0^{-1}B_\Delta)^2)}_{\gamma_{12}} + 2\underbrace{\text{Tr}(A_0^{-1}B_0A_0^{-1}A_XTA_0^{-1}B_\Delta)}_{\gamma_{13}} \quad (\text{H.19})$$

$$\gamma_2 = \underbrace{\text{Tr}((TA_0^{-1}B_0)^2)}_{\gamma_{21}} + \underbrace{\text{Tr}((TA_0^{-1}B_\Delta)^2)}_{\gamma_{22}} + 2\underbrace{\text{Tr}(TA_0^{-1}B_0TA_0^{-1}B_\Delta)}_{\gamma_{23}} \quad (\text{H.20})$$

$$\begin{aligned} \gamma_3 = & \underbrace{\text{Tr}((TA_0^{-1}B_0)^2A_0^{-1}A_X)}_{\gamma_{31}} + \underbrace{\text{Tr}((A_0^{-1}B_\Delta)^2T)}_{\gamma_{32}} \\ & + \underbrace{\text{Tr}(A_0^{-1}B_0TA_0^{-1}B_\Delta)}_{\gamma_{33}} + \underbrace{\text{Tr}(TA_0^{-1}B_\Delta TA_0^{-1}B_0A_0^{-1}A_X)}_{\gamma_{34}} \end{aligned}$$

557 Finally the last term can be expanded to:

$$\text{Tr}(A_0^{-1}B_XZB_X) = \underbrace{\text{Tr}(A_0^{-1}B_XA_0^{-1}A_XTA_0^{-1}B_X)}_{\beta_1} - \underbrace{\text{Tr}(A_0^{-1}B_XTA_0^{-1}B_X)}_{\beta_2} \quad (\text{H.21})$$

558 which can be expanded to:

$$\begin{aligned} \beta_1 = & \underbrace{\text{Tr}((A_0^{-1}B_0)^2A_0^{-1}A_XT)}_{\beta_{11}} + \underbrace{\text{Tr}((A_0^{-1}B_\Delta)^2)}_{\beta_{12}} \\ & + \underbrace{\text{Tr}(A_0^{-1}B_0A_0^{-1}A_XTA_0^{-1}B_\Delta)}_{\beta_{13}} + \underbrace{\text{Tr}(A_0^{-1}B_\Delta A_0^{-1}B_0)}_{\beta_{14}} \end{aligned}$$

559 and

$$\beta_2 = \underbrace{\text{Tr}((A_0^{-1}B_0)^2T)}_{\beta_{21}} + \underbrace{\text{Tr}((A_0^{-1}B_\Delta)^2T)}_{\beta_{22}} + \underbrace{\text{Tr}(A_0^{-1}B_\Delta A_0^{-1}B_0T)}_{\beta_{23}} + \underbrace{\text{Tr}(A_0^{-1}B_0A_0^{-1}B_\Delta T)}_{\beta_{24}} \quad (\text{H.22})$$

560 Some of the above terms cancel each other:

$$\alpha_2 + \gamma_{12} - 2\beta_{12} = 0 \quad (\text{H.23})$$

$$\alpha_3 - \beta_{13} = 0 \quad (\text{H.24})$$

$$\gamma_{32} - \beta_{22} = 0 \quad (\text{H.25})$$

$$\gamma_{33} - \beta_{23} = 0 \quad (\text{H.26})$$

561 and some of the terms can be re-written:

$$\gamma_{11} + \gamma_2 - 2(\gamma_{31} + \gamma_{34}) = \text{Tr}((TM)^2) \quad (\text{H.27})$$

$$\beta_{21} + \beta_{24} = \text{Tr}(A_0^{-1}B_0A_0^{-1}B_XT) \quad (\text{H.28})$$

562 which after some re-factorization leads to:

$$\text{Tr}(A_X^{-1}B_XA_X^{-1}B_X) = \text{Tr}((A_0^{-1}B_0)^2) + \text{Tr}((MT)^2) + 2\text{Tr}(A_0^{-1}B_0MT)$$

563 Note that here the 2 means $(AB)^2 = ABAB$

References

- [1] W. M. C. Foulkes, L. Mitas, R. J. Needs and G. Rajagopal, *Quantum monte carlo simulations of solids*, Rev. Mod. Phys. **73**, 33 (2001), doi:[10.1103/RevModPhys.73.33](https://doi.org/10.1103/RevModPhys.73.33).
- [2] R. Shinde, C. Filippi, A. Scemama and W. Jalby, *Shifting sands of hardware and software in exascale quantum mechanical simulations*, Nature Reviews Physics (2025), doi:[10.1038/s42254-025-00823-7](https://doi.org/10.1038/s42254-025-00823-7).
- [3] C. Filippi, *Champ* (2019).
- [4] J. Kim, A. D. Baczewski, T. D. Beaudet, A. Benali, M. C. Bennett, M. A. Berrill, N. S. Blunt, E. J. L. Borda, M. Casula, D. M. Ceperley, S. Chiesa, B. K. Clark *et al.*, *Qmcpack: an open source ab initio quantum monte carlo package for the electronic structure of atoms, molecules and solids*, Journal of Physics: Condensed Matter **30**(19), 195901 (2018), doi:[10.1088/1361-648X/aab9c3](https://doi.org/10.1088/1361-648X/aab9c3).
- [5] A. Scemama, M. Caffarel, E. Oseret and W. Jalby, *Qmc=chem: A quantum monte carlo program for large-scale simulations in chemistry at the petascale level and beyond*, In M. Dayde, O. Marques and K. Nakajima, eds., *High Performance Computing for Computational Science - VECPAR 2012*, pp. 118–127. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-38718-0, doi:[10.1007/978-3-642-38718-0_14](https://doi.org/10.1007/978-3-642-38718-0_14) (2013).
- [6] S. Shepard, R. L. Panadés-Barrueta, S. Moroni, A. Scemama and C. Filippi, *Double excitation energies from quantum monte carlo using state-specific energy optimization*, Journal of Chemical Theory and Computation **18**(11), 6722 (2022), doi:[10.1021/acs.jctc.2c00769](https://doi.org/10.1021/acs.jctc.2c00769), PMID: 36314602, <https://doi.org/10.1021/acs.jctc.2c00769>.
- [7] A. Cuzzocrea, S. Moroni, A. Scemama and C. Filippi, *Reference excitation energies of increasingly large molecules: A qmc study of cyanine dyes*, Journal of Chemical Theory and Computation **18**(2), 1089 (2022), doi:[10.1021/acs.jctc.1c01162](https://doi.org/10.1021/acs.jctc.1c01162), PMID: 35080893, <https://doi.org/10.1021/acs.jctc.1c01162>.
- [8] A. Cuzzocrea, A. Scemama, W. J. Briels, S. Moroni and C. Filippi, *Variational principles in quantum monte carlo: The troubled story of variance minimization*, Journal of Chemical Theory and Computation **16**(7), 4203 (2020), doi:[10.1021/acs.jctc.0c00147](https://doi.org/10.1021/acs.jctc.0c00147), PMID: 32419451, <https://doi.org/10.1021/acs.jctc.0c00147>.
- [9] M. Dash, S. Moroni, A. Scemama and C. Filippi, *Perturbatively selected configuration-interaction wave functions for efficient geometry optimization in quantum monte carlo*, Journal of Chemical Theory and Computation **14**(8), 4176 (2018), doi:[10.1021/acs.jctc.8b00393](https://doi.org/10.1021/acs.jctc.8b00393), PMID: 29953810, <https://doi.org/10.1021/acs.jctc.8b00393>.
- [10] E. Sloatman, I. Poltavsky, R. Shinde, J. Cocomello, S. Moroni, A. Tkatchenko and C. Filippi, *Accurate quantum monte carlo forces for machine-learned force fields: Ethanol as a benchmark*, Journal of Chemical Theory and Computation **20**(14), 6020 (2024), doi:[10.1021/acs.jctc.4c00498](https://doi.org/10.1021/acs.jctc.4c00498), PMID: 39003522, <https://doi.org/10.1021/acs.jctc.4c00498>.
- [11] B. M. Austin, D. Y. Zubarev and W. A. J. Lester, *Quantum Monte Carlo and Related Approaches*, Chemical Reviews **112**(1), 263 (2012), doi:[10.1021/cr2001564](https://doi.org/10.1021/cr2001564), Publisher: American Chemical Society.

- [12] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, Science **355**(6325), 602–606 (2017), doi:[10.1126/science.aag2302](https://doi.org/10.1126/science.aag2302).
- [13] P.-J. Yang, M. Sugiyama, K. Tsuda and T. Yanai, *Artificial neural networks applied as molecular wave function solvers*, Journal of Chemical Theory and Computation **16**(6), 3513 (2020), doi:[10.1021/acs.jctc.9b01132](https://doi.org/10.1021/acs.jctc.9b01132), PMID: 32320233, <https://doi.org/10.1021/acs.jctc.9b01132>.
- [14] J. Lin, G. Goldshlager and L. Lin, *Explicitly antisymmetrized neural network layers for variational monte carlo simulation*, Journal of Computational Physics **474**, 111765 (2023), doi:[10.1016/j.jcp.2022.111765](https://doi.org/10.1016/j.jcp.2022.111765).
- [15] Z. Schätzle, J. Hermann and F. Noé, *Convergence to the fixed-node limit in deep variational Monte Carlo*, The Journal of Chemical Physics **154**(12), 124108 (2021), doi:[10.1063/5.0032836](https://doi.org/10.1063/5.0032836), https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0032836/14009445/124108_1_online.pdf.
- [16] K. Inui, Y. Kato and Y. Motome, *Determinant-free fermionic wave function using feed-forward neural networks*, Phys. Rev. Res. **3**, 043126 (2021), doi:[10.1103/PhysRevResearch.3.043126](https://doi.org/10.1103/PhysRevResearch.3.043126).
- [17] J. Kessler, F. Calcavecchia and T. D. Kühne, *Artificial neural networks as trial wave functions for quantum monte carlo*, Advanced Theory and Simulations **4**(4), 2000269 (2021), doi:[10.1002/adts.202000269](https://doi.org/10.1002/adts.202000269), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adts.202000269>.
- [18] J. Han, L. Zhang and W. E, *Solving many-electron schrödinger equation using deep neural networks*, Journal of Computational Physics **399**, 108929 (2019), doi:[10.1016/j.jcp.2019.108929](https://doi.org/10.1016/j.jcp.2019.108929).
- [19] K. Choo, A. Mezzacapo and G. Carleo, *Fermionic neural-network states for ab-initio electronic structure*, Nature Communications **11**(1), 2368 (2020), doi:[10.1038/s41467-020-15724-9](https://doi.org/10.1038/s41467-020-15724-9).
- [20] G. Cassella, W. M. C. Foulkes, D. Pfau and J. S. Spencer, *Neural network variational Monte Carlo for positronic chemistry*, Nature Communications **15**(1), 5214 (2024), doi:[10.1038/s41467-024-49290-1](https://doi.org/10.1038/s41467-024-49290-1).
- [21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf et al., *Pytorch: An imperative style, high-performance deep learning library* (2019), [1912.01703](https://arxiv.org/abs/1912.01703).
- [22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg et al., *Tensorflow: A system for large-scale machine learning* (2016), [1605.08695](https://arxiv.org/abs/1605.08695).
- [23] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne and Q. Zhang, *JAX: composable transformations of Python+NumPy programs* (2018).
- [24] D. Pfau, J. Spencer, A. de G. Matthews and W. Foulkes, *Ab-initio solution of the many-electron schrödinger equation with deep neural networks*, Phys. Rev. Research **2**, 033429 (2020), doi:[10.1103/PhysRevResearch.2.033429](https://doi.org/10.1103/PhysRevResearch.2.033429).

- [25] J. Hermann, Z. Schätzle and F. Noé, *Deep-neural-network solution of the electronic schrödinger equation*, Nature Chemistry **12**(10), 891 (2020), doi:[10.1038/s41557-020-0544-y](https://doi.org/10.1038/s41557-020-0544-y).
- [26] M. T. Entwistle, Z. Schätzle, P. A. Erdman, J. Hermann and F. Noé, *Electronic excited states in deep variational Monte Carlo*, Nature Communications **14**(1), 274 (2023), doi:[10.1038/s41467-022-35534-5](https://doi.org/10.1038/s41467-022-35534-5).
- [27] D. Pfau, S. Axelrod, H. Sutterud, I. von Glehn and J. S. Spencer, *Accurate computation of quantum excited states with neural networks*, Science **385**(6711) (2024), doi:[10.1126/science.adn0137](https://doi.org/10.1126/science.adn0137).
- [28] P. B. Szabó, Z. Schätzle, M. T. Entwistle and F. Noé, *An improved penalty-based excited-state variational monte carlo approach with deep-learning ansatzes*, Journal of Chemical Theory and Computation (2024), doi:[10.1021/acs.jctc.4c00678](https://doi.org/10.1021/acs.jctc.4c00678).
- [29] N. Renaud, *Qmctorch: a pytorch implementation of real-space quantum monte carlo simulations of molecular systems*, Journal of Open Source Software **8**(91), 5472 (2023), doi:[10.21105/joss.05472](https://doi.org/10.21105/joss.05472).
- [30] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters and G. K.-L. Chan, *The python-based simulations of chemistry framework (pyscf)* (2017), [1701.08223](https://doi.org/10.26434/chemrxiv-2017-1701).
- [31] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders and T. Ziegler, *Chemistry with adf*, J. Comput. Chem. **22**(9), 931 (2001), doi:[10.1002/jcc.1056](https://doi.org/10.1002/jcc.1056).
- [32] C. Filippi, R. Assaraf and S. Moroni, *Simple formalism for efficient derivatives and multi-determinant expansions in quantum monte carlo*, The Journal of Chemical Physics **144**(19) (2016), doi:[10.1063/1.4948778](https://doi.org/10.1063/1.4948778).
- [33] R. Assaraf, S. Moroni and C. Filippi, *Optimizing the Energy with Quantum Monte Carlo: A Lower Numerical Scaling for Jastrow–Slater Expansions*, Journal of Chemical Theory and Computation **13**(11), 5273 (2017), doi:[10.1021/acs.jctc.7b00648](https://doi.org/10.1021/acs.jctc.7b00648), Publisher: American Chemical Society.
- [34] P. López Ríos, A. Ma, N. D. Drummond, M. D. Towler and R. J. Needs, *Inhomogeneous backflow transformations in quantum monte carlo calculations*, Physical Review E **74**(6) (2006), doi:[10.1103/physreve.74.066701](https://doi.org/10.1103/physreve.74.066701).
- [35] M. Holzmann and S. Moroni, *Orbital-dependent backflow wave functions for real-space quantum monte carlo*, Physical Review B **99**(8) (2019), doi:[10.1103/physrevb.99.085121](https://doi.org/10.1103/physrevb.99.085121).
- [36] D. Luo and B. K. Clark, *Backflow transformations via neural networks for quantum many-body wave functions*, Physical Review Letters **122**(22) (2019), doi:[10.1103/physrevlett.122.226401](https://doi.org/10.1103/physrevlett.122.226401).
- [37] A.-J. Liu and B. K. Clark, *Neural network backflow for ab initio quantum chemistry*, Physical Review B **110**(11) (2024), doi:[10.1103/physrevb.110.115137](https://doi.org/10.1103/physrevb.110.115137).
- [38] Z. Liu and B. K. Clark, *Unifying view of fermionic neural network quantum states: From neural network backflow to hidden fermion determinant states*, Phys. Rev. B **110**, 115124 (2024), doi:[10.1103/PhysRevB.110.115124](https://doi.org/10.1103/PhysRevB.110.115124).

- [39] I. Romero, J. Nys and G. Carleo, *Spectroscopy of two-dimensional interacting lattice electrons using symmetry-aware neural backflow transformations*, Communications Physics **8**(1), 46 (2025), doi:[10.1038/s42005-025-01955-z](https://doi.org/10.1038/s42005-025-01955-z).
- [40] P. López Ríos, P. Seth, N. D. Drummond and R. J. Needs, *Framework for constructing generic jastrow correlation factors*, Phys. Rev. E **86**, 036703 (2012), doi:[10.1103/PhysRevE.86.036703](https://doi.org/10.1103/PhysRevE.86.036703).
- [41] D. Ceperley, G. V. Chester and M. H. Kalos, *Monte carlo simulation of a many-fermion study*, Phys. Rev. B **16**, 3081 (1977), doi:[10.1103/PhysRevB.16.3081](https://doi.org/10.1103/PhysRevB.16.3081).
- [42] S. Pathak and L. K. Wagner, *A light weight regularization for wave function parameter gradients in quantum monte carlo*, AIP Advances **10**(8), 085213 (2020), doi:[10.1063/5.0004008](https://doi.org/10.1063/5.0004008), https://pubs.aip.org/aip/adv/article-pdf/doi/10.1063/5.0004008/9048174/085213_1_online.pdf.
- [43] C. Attaccalite and S. Sorella, *Stable liquid hydrogen at high pressure by a novel ab initio molecular-dynamics calculation*, Phys. Rev. Lett. **100**, 114501 (2008), doi:[10.1103/PhysRevLett.100.114501](https://doi.org/10.1103/PhysRevLett.100.114501).
- [44] Y. Min and N. Azizan, *Hardnet: Hard-constrained neural networks with universal approximation guarantees* (2025), [2410.10807](https://arxiv.org/abs/2410.10807).
- [45] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko and K.-R. Müller, *Schnet: A continuous-filter convolutional neural network for modeling quantum interactions* (2017), [1706.08566](https://arxiv.org/abs/1706.08566).
- [46] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization* (2017), [1412.6980](https://arxiv.org/abs/1412.6980).
- [47] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization* (2019), [1711.05101](https://arxiv.org/abs/1711.05101).
- [48] J. Martens and R. Grosse, *Optimizing neural networks with kronecker-factored approximate curvature* (2020), [1503.05671](https://arxiv.org/abs/1503.05671).
- [49] C. J. Umrigar, *Accelerated metropolis method*, Phys. Rev. Lett. **71**, 408 (1993), doi:[10.1103/PhysRevLett.71.408](https://doi.org/10.1103/PhysRevLett.71.408).
- [50] C. J. Umrigar, M. P. Nightingale and K. J. Runge, *A diffusion monte carlo algorithm with very small time-step errors*, The Journal of Chemical Physics **99**(4), 2865 (1993), doi:[10.1063/1.465195](https://doi.org/10.1063/1.465195), https://pubs.aip.org/aip/jcp/article-pdf/99/4/2865/19121424/2865_1_online.pdf.
- [51] W. Kolos and L. Wolniewicz, *Potential-energy curves for the $x1\sigma g+$, $b3\sigma u+$, and $c1\pi u$ states of the hydrogen molecule*, The Journal of Chemical Physics **43**(7), 2429 (1965), doi:[10.1063/1.1697142](https://doi.org/10.1063/1.1697142), https://pubs.aip.org/aip/jcp/article-pdf/43/7/2429/18840418/2429_1_online.pdf.
- [52] W. Cencek and J. Rychlewski, *Benchmark calculations for $he2+$ and lih molecules using explicitly correlated gaussian functions*, Chemical Physics Letters **320**(5), 549 (2000), doi:[https://doi.org/10.1016/S0009-2614\(00\)00303-1](https://doi.org/10.1016/S0009-2614(00)00303-1).
- [53] C. Filippi and C. J. Umrigar, *Multiconfiguration wave functions for quantum monte carlo calculations of first-row diatomic molecules*, The Journal of Chemical Physics **105**(1), 213 (1996), doi:[10.1063/1.471865](https://doi.org/10.1063/1.471865), https://pubs.aip.org/aip/jcp/article-pdf/105/1/213/19076740/213_1_online.pdf.

- [54] A. Annarelli, D. Alfè and A. Zen, *A brief introduction to the diffusion monte carlo method and the fixed-node approximation*, The Journal of Chemical Physics **161**(24) (2024), doi:[10.1063/5.0232424](https://doi.org/10.1063/5.0232424).
- [55] W. Ren, W. Fu, X. Wu and J. Chen, *Towards the ground state of molecules via diffusion Monte Carlo on neural networks*, Nature Communications **14**(1), 1860 (2023), doi:[10.1038/s41467-023-37609-3](https://doi.org/10.1038/s41467-023-37609-3).
- [56] C. Huang and B. M. Rubenstein, *Machine Learning Diffusion Monte Carlo Forces*, The Journal of Physical Chemistry A **127**(1), 339 (2023), doi:[10.1021/acs.jpca.2c05904](https://doi.org/10.1021/acs.jpca.2c05904), Publisher: American Chemical Society.
- [57] J. Feldt and C. Filippi, *Excited-state calculations with quantum monte carlo* (2020), [2002.03622](https://doi.org/10.26434/chemrxiv-2020-03622).
- [58] C. P. Robert, *The metropolis-hastings algorithm* (2016), [1504.01896](https://doi.org/10.1007/978-1-4939-9837-9).
- [59] N. D. Drummond, M. D. Towler and R. J. Needs, *Jastrow correlation factor for atoms, molecules, and solids*, Physical Review B **70**(23) (2004), doi:[10.1103/physrevb.70.235119](https://doi.org/10.1103/physrevb.70.235119).
- [60] L. Bertini, M. Mella, D. Bressanini and G. Morosi, *Explicitly correlated trial wavefunctions in quantum monte carlo calculations of excited states of be and be-*, Journal of Physics B: Atomic, Molecular and Optical Physics **34**(3), 257–266 (2001), doi:[10.1088/0953-4075/34/3/304](https://doi.org/10.1088/0953-4075/34/3/304).
- [61] Y. Wu, H. Wang, B. Zhang and K.-L. Du, *Using radial basis function networks for function approximation and classification*, International Scholarly Research Notices **2012**(1), 324194 (2012), doi:<https://doi.org/10.5402/2012/324194>, <https://onlinelibrary.wiley.com/doi/pdf/10.5402/2012/324194>.
- [62] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*, Version 20121115 (2012).