Journal homepage http://revistas.unitru.edu.pe/index.php/SSMM

SELECCIONES MATEMÁTICAS
Universidad Nacional de Trujillo
ISSN: 2411-1783 (Online)
2026; Vol.13(1):70-76.

Ab Initio Wave Function Optimization for Light Atoms

A PyTorch Framework for the Psiformer Architecture

Jorge Muñoz Laredo

Received, Jan. 20, 2026; *Accepted, May. 05, 2026;* *Published, Jul. 27, 2026*

How to cite this article:
Muñoz Laredo J. *Ab initio wave function optimization for light atoms: a PyTorch framework for the Psiformer architecture*. Selecciones Matemáticas. 2026;13(1):70–76. https://doi.org/10.17268/sel.mat.2026.01.05

Abstract

The application of deep neural networks to Variational Quantum Monte Carlo (VQMC) has significantly advanced the numerical approximation of ground-state energies for quantum many-body systems. Among these advancements, the Psiformer architecture—which leverages self-attention mechanisms to model antisymmetric electron wave functions—has demonstrated state-of-the-art accuracy. However, the original implementation is built exclusively within the JAX ecosystem, presenting a barrier to entry for the broader research community that predominantly utilizes PyTorch. In the interest of scientific reproducibility and computational accessibility, we present an independent open-source PyTorch implementation of the Psiformer architecture. The model design follows von Glehn, Spencer, and Pfau (2023); the code and numerical experiments reported here are the author's own. To validate the implementation, we compute ground-state energies for helium through oxygen and compare them to a Hartree–Fock baseline. Our results demonstrate that the PyTorch Psiformer consistently recovers significant electron correlation energy across all tested atoms, confirming that the architecture is correctly implemented and behaves as expected from variational theory. By making the framework openly available, this work lowers the barrier for researchers in computational mathematics, machine learning, and quantum chemistry who wish to experiment with transformer-based wave functions. The full source code is made publicly available to facilitate future experimentation and development.

Keywords . Many-electron Schrödinger equation, variational Monte Carlo, transformer, neural wavefunction, Slater determinant, quantum chemistry.

1. Introduction. Solving the time-independent many-body Schrödinger equation is one of the central challenges in computational physics and quantum chemistry. Because analytical solutions are impossible for systems with more than one electron, highly accurate numerical approximations are required. Traditional ab initio methods, such as Hartree-Fock (HF), Density Functional Theory (DFT), and Coupled Cluster (CC) methods, have long been the standard for estimating ground-state energies. However, these methods often face a steep trade-off between computational scalability and the accuracy required to capture complex electron correlation effects.

In recent years, Variational Quantum Monte Carlo (VQMC) methods [1] have been revolutionized by the application of deep neural networks. By parameterizing the trial wave function using deep learning architectures, researchers can capture complex many-body correlations without relying on traditional, fixed basis sets. A major breakthrough in this domain was the introduction of neural network wave functions such as PauliNet [2], FermiNet [3], and, subsequently, the Psiformer architecture of von Glehn, Spencer, and Pfau [4], introduced at DeepMind. Psiformer utilizes a transformer-based mechanism with self-attention to

*Escuela Profesional de Física, Universidad Nacional de Ingeniería, Perú. (jorge.munoz.l@uni.pe)

model the antisymmetric wave function of electrons, achieving state-of-the-art accuracy on various molecular and atomic systems.

Despite the profound theoretical and empirical contributions of the Psiformer architecture, independent replication and modification of such models can be challenging. DeepMind's official implementations are natively built within the JAX ecosystem. While JAX is highly efficient, PyTorch remains the dominant framework in the broader deep learning and applied mathematics communities. Translating complex, physics-informed architectures across frameworks is not a trivial task, yet it is a crucial step for scientific reproducibility, cross-validation, and democratizing access to advanced computational tools.

In this paper, we present a complete, open-source PyTorch implementation of the Psiformer ansatz [4]. The architectural design and theoretical formulation follow von Glehn, Spencer, and Pfau; the codebase, training pipeline, and experiments reported here were developed independently by the author. Our primary objective is to validate that this PyTorch implementation correctly captures the underlying physics of the Psiformer architecture. To this end, we evaluate the model by computing ground-state energies for helium through oxygen (seven atoms), using Hartree–Fock energies as a baseline. Our results show that the model consistently recovers electron correlation energy well below the baseline for every atom tested, demonstrating correct variational behaviour and successful implementation of the architecture.

By providing a working, validated PyTorch implementation of Psiformer, this work aims to lower the barrier to entry for researchers in computational mathematics, machine learning, and quantum chemistry who wish to experiment with transformer-based wave functions. The complete source code, including training routines and hyperparameters for the evaluated atoms, is publicly available at: https://github.com/jorgemunozl/psiformer_torch.git.

The remainder of this paper is organized as follows: Section 2 presents a compact mathematical formulation of the electronic ground-state problem and the Variational Monte Carlo (VMC) optimization objective. Section 3 describes the structural choices of our Psiformer-style ansatz and its PyTorch implementation. Section 4 reports numerical results for helium through oxygen, and Section 5 discusses limitations and future work.

2. Mathematical formulation.

2.1. Problem setup and Hamiltonian. We seek the ground state of the stationary many-electron Schrödinger eigenproblem

$$\hat{H}\Psi(\mathbf{x}_1,\dots,\mathbf{x}_N) = E\,\Psi(\mathbf{x}_1,\dots,\mathbf{x}_N), \tag{2.1}$$

where $\mathbf{x}_i = (\mathbf{r}_i, \sigma_i)$ collects the position $\mathbf{r}_i \in \mathbb{R}^3$ and spin $\sigma_i \in \{\uparrow,\downarrow\}$. We work in atomic units and adopt the Born–Oppenheimer approximation (clamped nuclei). A standard molecular Hamiltonian is

$$\begin{aligned}\hat{H} = &-\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{I=1}^{M}\frac{1}{2M_I}\nabla_I^2 - \sum_{i=1}^{N}\sum_{I=1}^{M}\frac{Z_I}{|\mathbf{r}_i-\mathbf{R}_I|}\\ &+\sum_{1\le i<j\le N}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} + \sum_{1\le I<J\le M}\frac{Z_I Z_J}{|\mathbf{R}_I-\mathbf{R}_J|}.\end{aligned} \tag{2.2}$$

Under Born–Oppenheimer, nuclear positions $\{\mathbf{R}_I\}$ are fixed and one typically drops the nuclear kinetic terms when solving the electronic problem. Henceforth we denote the remaining electronic Hamiltonian by $\hat{H}_{\text{el}}$.

$$\hat{H}_{\text{el}} = -\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2 - \sum_{i=1}^{N}\sum_{I=1}^{M}\frac{Z_I}{|\mathbf{r}_i-\mathbf{R}_I|} + \sum_{1\le i<j\le N}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} + \sum_{1\le I<J\le M}\frac{Z_I Z_J}{|\mathbf{R}_I-\mathbf{R}_J|}, \tag{2.3}$$

where the nuclear kinetic term and the nuclear repulsion constant have been separated; the latter simply adds a constant shift to the energy.

2.2. Antisymmetry and cusp constraints. Electrons are indistinguishable fermions and therefore obey the Pauli exclusion principle: the many-body wavefunction must change sign under the exchange of any two same-spin electrons. This requirement, known as the antisymmetry condition, can be enforced by constructing the wavefunction as a Slater determinant (or a linear combination thereof). A determinant changes sign whenever two rows or two columns are interchanged, which exactly mimics the fermionic exchange symmetry. Given a set of single-particle spin orbitals $\phi_j(\mathbf{x}_i)$, where $\mathbf{x}_i = (\mathbf{r}_i, \sigma_i)$ collects the

position and spin of the i-th electron, the simplest antisymmetric form is

$$\Psi(\mathbf{x}_1,\dots,\mathbf{x}_N)=\frac{1}{\sqrt{N!}}\begin{vmatrix}\phi_1(\mathbf{x}_1) & \dots & \phi_1(\mathbf{x}_N)\\ \vdots & \ddots & \vdots\\ \phi_N(\mathbf{x}_1) & \dots & \phi_N(\mathbf{x}_N)\end{vmatrix}. \tag{2.4}$$

In practical neural-network ansätze, the orbitals are learned functions of the electron coordinates, and the single-determinant form is generalized to a weighted sum over multiple determinants, as will be introduced in Section 3.

The Coulomb potential diverges when particles coincide. To keep the total energy finite, the kinetic energy operator must produce a compensating divergence, which forces the wavefunction to have a non-differentiable cusp at each coalescence point. Kato's theorem [5] provides exact expressions for the spherical-average logarithmic derivative at these singularities. For an electron–nucleus coalescence one has

$$\lim_{r_{iI}\to 0}\frac{\partial \ln \Psi}{\partial r_{iI}}=-Z_I,$$

where $r_{iI}=|\mathbf{r}_i-\mathbf{R}_I|$ and Z_I is the nuclear charge. For an electron–electron coalescence the analogous condition reads

$$\lim_{r_{ij}\to 0}\frac{\partial \ln \Psi}{\partial r_{ij}}=\frac{1}{2},$$

with $r_{ij}=|\mathbf{r}_i-\mathbf{r}_j|$. These constraints must be built into any trial wavefunction to obtain accurate variational energies. A common approach is to multiply the antisymmetric part by a Jastrow factor $\exp(\mathcal{J})$, a positive-definite function designed to capture short-range correlations and, in particular, to satisfy the electron–electron cusp conditions. Our specific choice of $\mathcal{J}$ is described in Section 3.

2.3. Variational Monte Carlo objective. The variational principle provides an upper bound on the ground-state energy: for any trial wavefunction Ψ,

$$E[\Psi]=\frac{\langle\Psi|\,\hat{H}_{\text{el}}\,|\Psi\rangle}{\langle\Psi|\Psi\rangle}\geq E_0, \tag{2.5}$$

where E_0 is the exact ground-state energy and $\hat{H}_{\text{el}}$ denotes the electronic Hamiltonian obtained after the Born–Oppenheimer approximation (Eq. 2.3). Minimising $E[\Psi]$ with respect to variational parameters drives the trial function toward the true ground state. Writing Ψ_θ for a parameterised ansatz, the objective becomes

$$\mathcal{L}(\theta)=\frac{\langle\Psi_\theta|\,\hat{H}_{\text{el}}\,|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}=\frac{\int \Psi_\theta^*(\mathbf{R})\,\hat{H}_{\text{el}}\Psi_\theta(\mathbf{R})\,d\mathbf{R}}{\int \Psi_\theta^*(\mathbf{R})\Psi_\theta(\mathbf{R})\,d\mathbf{R}}. \tag{2.6}$$

Introducing the normalised probability density $p_\theta(\mathbf{R})\propto|\Psi_\theta(\mathbf{R})|^2$ and the *local energy*

$$E_L(\mathbf{R})=\frac{\hat{H}_{\text{el}}\Psi_\theta(\mathbf{R})}{\Psi_\theta(\mathbf{R})}, \tag{2.7}$$

the loss reduces to the expectation value $\mathcal{L}(\theta)=\mathbb{E}_{\mathbf{R}\sim p_\theta}[E_L(\mathbf{R})]$.

2.4. Monte Carlo estimation, sampling, and gradients. Given samples $\mathbf{R}_1,\dots,\mathbf{R}_M$ drawn from $|\Psi_\theta|^2$ (up to normalization),

$$\mathcal{L}_\theta=\mathbb{E}_{\mathbf{R}\sim|\Psi_\theta|^2}[E_L(\mathbf{R})]\approx\frac{1}{M}\sum_{k=1}^{M}E_L(\mathbf{R}_k). \tag{2.8}$$

The local energy can be written as

$$E_L(\mathbf{R}_k)=\frac{\hat{H}_{\text{el}}\Psi_\theta(\mathbf{R}_k)}{\Psi_\theta(\mathbf{R}_k)}=-\frac{1}{2}\frac{\nabla^2\Psi_\theta(\mathbf{R}_k)}{\Psi_\theta(\mathbf{R}_k)}+V(\mathbf{R}_k). \tag{2.9}$$

Samples are generated with Metropolis–Hastings [6]: starting from $\mathbf{R}_0$, propose $\mathbf{R}'=\mathbf{R}_0+\eta$ with $\eta\sim q$, accept with probability

$$A(\mathbf{R}_0,\mathbf{R}')=\min\left(1,\frac{\rho(\mathbf{R}')}{\rho(\mathbf{R}_0)}\right), \tag{2.10}$$

where $\rho \propto |\Psi_\theta|^2$, and iterate. Gradients of the VMC objective follow as (schematically)

$$\nabla_\theta \mathcal{L} = 2\,\mathbb{E}_{\mathbf{x}\sim|\Psi|^2}\left[(E_L(\mathbf{x}) - \mathbb{E}_p[E_L])\,\nabla_\theta \log \Psi_\theta(\mathbf{x})\right], \tag{2.11}$$

estimated with the same samples.

3. Structural choices.

3.1. Transformer backbone and self-attention. The backbone applies multi-head self-attention [7] to a sequence of per-electron hidden states, allowing every electron to attend to every other electron and aggregate information about the surrounding electronic configuration. Each attention head operates on a d_h-dimensional slice of the embedding, computing scaled dot-product attention; the outputs of all H heads are concatenated and projected back to the full embedding dimension d. In the Psiformer, this permutation-equivariant attention step replaces the graph-convolution layers of earlier neural wave functions, enabling long-range electron–electron correlations to be captured with $O(N^2)$ cost in the number of electrons N.

3.2. Psiformer ansatz and architecture. The trial wavefunction is an *ansatz*: a model chosen by design and optimized by VMC. We require antisymmetry under particle exchange and expressiveness for strong electron–correlation effects. Following [4, 3], we use a Slater–Jastrow form,

$$\Psi_\theta(\mathbf{R}) = \underbrace{\exp\big(\mathcal{J}_\theta(\mathbf{R})\big)}_{\text{Coulomb correlations}} \times \underbrace{\sum_k \omega_k \det\big[\mathbf{\Phi}^k_\theta(\mathbf{R})\big]}_{\text{antisymmetry}}. \tag{3.1}$$

We use a simple, parameter-efficient Jastrow factor to enforce short-range cusp structure, separated by spin alignment,

$$\mathcal{J}_\theta(\mathbf{R}) = \sum_{\substack{i<j \\ \sigma_i=\sigma_j}} -\frac{1}{4}\frac{\alpha_{\text{par}}^2}{\alpha_{\text{par}} + |\mathbf{r}_i - \mathbf{r}_j|} + \sum_{\substack{i,j \\ \sigma_i\neq\sigma_j}} -\frac{1}{2}\frac{\alpha_{\text{anti}}^2}{\alpha_{\text{anti}} + |\mathbf{r}_i - \mathbf{r}_j|}. \tag{3.2}$$

The envelope is implemented as a matrix with learned exponential decay; the Jastrow uses few parameters and is trained in log scale in practice.

We benchmark two model scales; Table 3.1 summarises the corresponding training configuration.

Table 3.1: Training configuration for small and large model variants. Both variants use AdamW [8] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay $\lambda = 10^{-4}$) with a cosine annealing learning rate schedule.

Parameter	**Small**	**Large**
Layers L	2	4
Heads H	4	8
Model dimension d	256	512
MLP dimension d_{ff}	1024	2048
Determinants K	1	2
MCMC walkers N_w	1024	2048
MCMC steps / iteration	10	10
Learning rate	2×10^{-4}	1×10^{-4}
Total parameters	50.4×10^3	3.16×10^6

3.3. Implementation details.

3.3.1. Laplacian via automatic differentiation. The kinetic energy requires the trace of the Hessian, $\nabla^2\Psi = \sum_{i=1}^{3N} \partial^2\Psi/\partial R_i^2$, evaluated for each MCMC sample. In PyTorch [9] this is computed by calling `autograd.grad` twice: a first pass with `create_graph=True` yields $\nabla\Psi$, and a second pass iterates over coordinate components to accumulate the diagonal second derivatives. Because the graph is retained between the two passes, gradients flow correctly through the Laplacian during backpropagation. This approach is exact (no finite-difference approximation) but requires $3N$ backward passes per sample, making it the dominant cost during training.

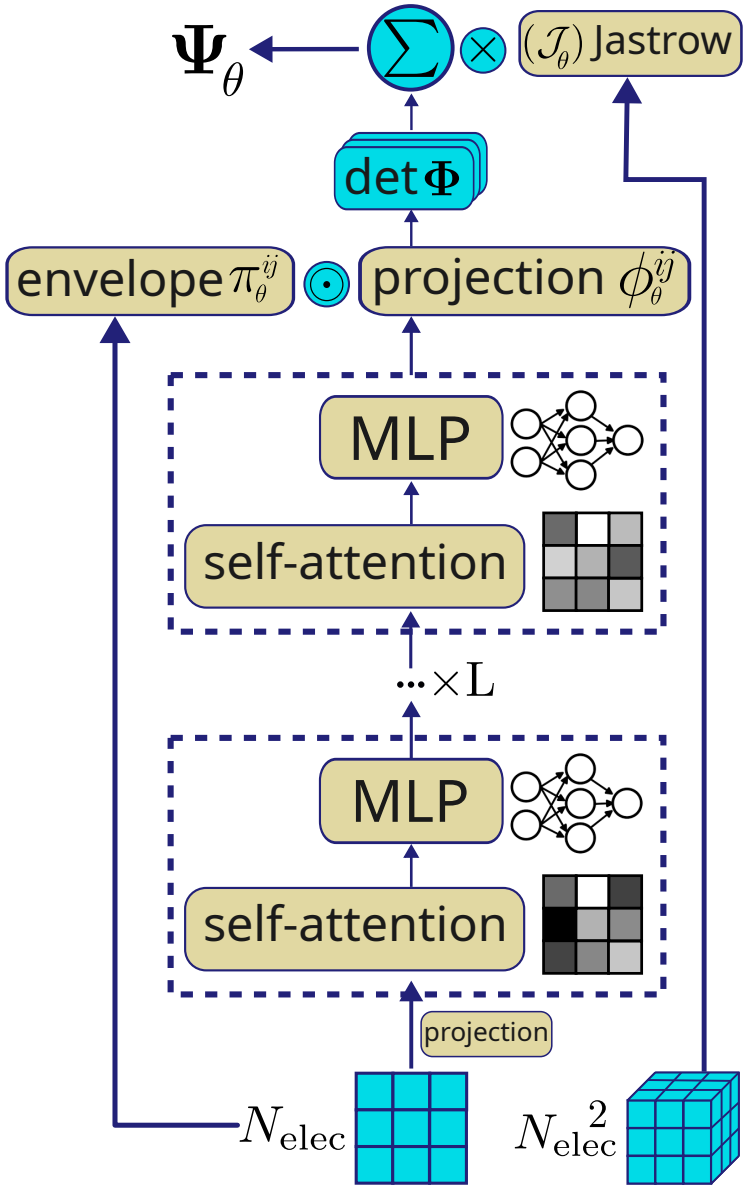


Figure 3.1: Psiformer-style architecture.

3.3.2. Stable $\log\det$ **gradient.** Since $\Psi_\theta \propto \det \mathbf{\Phi}_\theta(\mathbf{R})$, training uses $\log|\det \mathbf{\Phi}|$ for stability. The matrix derivative identity $\partial(\det A)/\partial A = \det(A)A^{-\mathsf{T}}$ becomes ill-conditioned near singular $\mathbf{\Phi}$. An SVD-based custom autograd function implements $\nabla_A \log|\det A|$ without explicitly forming unstable inverses.

3.3.3. Optimizer and batching. Parameters are optimised with AdamW [8] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay $\lambda = 10^{-4}$) under a cosine annealing learning rate schedule. Compared to standard Adam, AdamW applies the weight-decay term directly to the parameters rather than incorporating it into the gradient, an empirical choice that often improves generalisation in over-parameterised models.

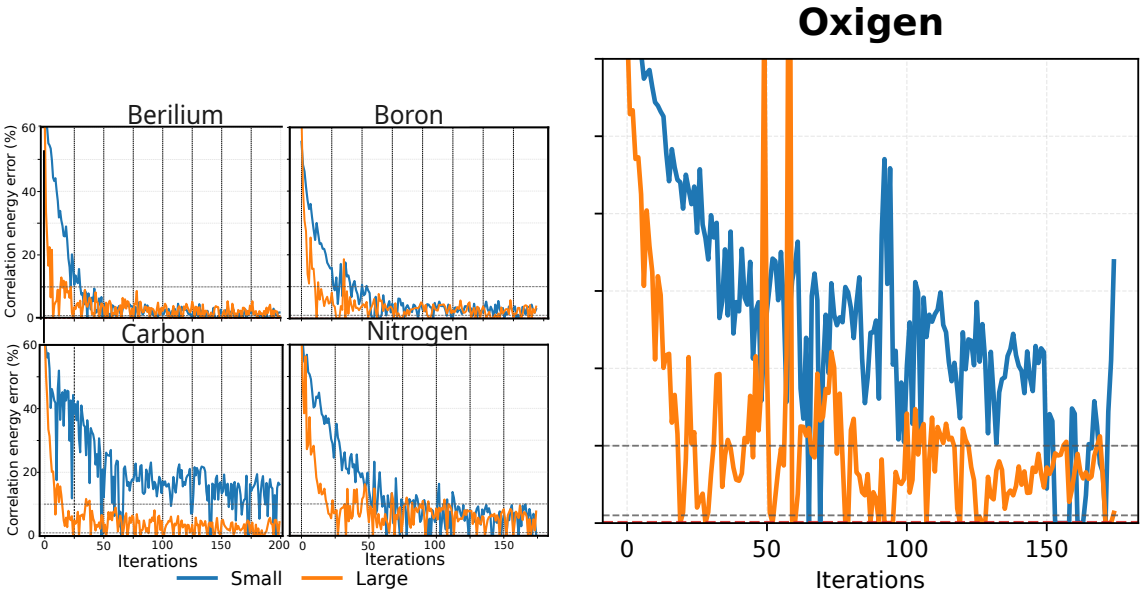
To improve GPU utilisation, we avoid evaluating the local energy one MCMC step at a time. Instead, we flatten the sampling dimension: given a batch of B chains with M Metropolis–Hastings steps each and n_e electrons, the configuration tensor of shape

$$(M,\, B,\, n_e,\, 3) \tag{3.3}$$

is reshaped into $(M \times B,\, n_e,\, 3)$ before the forward pass. This simple batching strategy raised GPU occupancy from approximately 30% to 99% in our runs.

4. Results. We evaluate the PyTorch Psiformer on isolated atoms from helium ($Z = 2$) through oxygen ($Z = 8$), comparing the converged variational energies against the Hartree–Fock limit for each element.

4.1. Convergence behaviour. Figure 4.1 shows representative training curves for both model variants on selected elements. In every case the variational energy decreases monotonically during the first several thousand iterations and then stabilises around a well-defined plateau, indicating that the MCMC sampling chain and the neural-network optimisation have reached a consistent equilibrium. Figure 4.1b isolates the oxygen atom, the most challenging system in our benchmark due to its eight-electron, open-shell configuration; the curve exhibits the same stable convergence pattern observed for the lighter atoms, confirming that the model scales gracefully with electron count.



(a) Energy convergence for He, Li, Be, and C.

(b) Oxygen convergence (8 electrons).

Figure 4.1: VMC energy as a function of training iteration. Shaded bands indicate ± 1 standard deviation of the local-energy estimator across walkers.

4.2. Ground-state energies. Table 4.1 reports the final ground-state energies obtained with our PyTorch implementation alongside Hartree–Fock references. Across all seven atoms the computed energies lie consistently below the Hartree–Fock limit, confirming that the model captures electron correlation energy. The large variant yields a mean correlation energy of 55.7 ± 0.2 mHa, compared to 54.0 ± 0.2 mHa for the small variant, demonstrating that increased model capacity translates into improved variational accuracy.

Table 4.1: Ground-state energies (Ha) for He–O: Hartree–Fock baseline compared with our PyTorch Psiformer (small and large variants).

Atom	HF	Psiformer (small)	Psiformer (large)
He	-2.86168	$-2.8980(2)$	$-2.9029(4)$
Li	-7.43273	$-7.4767(3)$	$-7.4778(2)$
Be	-14.5730	$-14.6124(3)$	$-14.6142(3)$
B	-24.5291	$-24.5919(3)$	$-24.5940(3)$
C	-37.6886	$-37.7638(4)$	$-37.7664(3)$
N	-54.4009	$-54.5020(5)$	$-54.5049(4)$
O	-74.8094	$-74.9203(5)$	$-74.9244(4)$

5. Discussion and future work. The results above demonstrate that the PyTorch Psiformer implementation produces physically correct variational behaviour: both model variants systematically lower the

ground-state energy below the Hartree–Fock baseline for every atom from helium through oxygen, recovering tens of mHa of electron correlation energy. The consistent improvement from the small to the large variant (54.0 ± 0.2 vs. 55.7 ± 0.2 mHa mean improvement; individual energies with statistical uncertainties are reported in Table 4.1) also shows that the architecture scales as expected with model capacity. A direct numerical comparison with the published Psiformer reference code (JAX) is left for future work, as it requires running both implementations under identical hardware, random seeds, and training budgets; nevertheless, the qualitative behaviour and energy ordering observed here are consistent with variational theory and with the trends reported in [4].

Several directions remain open for future work: (i) pretraining on Hartree–Fock or DFT orbitals to accelerate convergence, (ii) reducing the cost of Laplacian computation, which dominates training time, (iii) adopting KFAC or other natural-gradient optimisers for more efficient parameter updates, (iv) integrating FlashAttention-style kernels for improved memory and speed scaling, (v) transfer learning across chemical systems to amortise training cost, and (vi) empirical scaling laws relating model size and sample budget to final energy accuracy.

acknowledgment. The author thanks the Computer Science Faculty for access to GPU resources, in particular an NVIDIA GeForce RTX 4080 Super, which enabled the training and evaluation reported here.

Author contributions. Conceptualization, methodology, software, formal analysis, investigation, writing—original draft preparation, writing—review and editing: J.M.L. The author has read and agreed to the published version of the manuscript.

Funding. This research received no external funding.

Conflicts of interest. The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

ORCID and License
Jorge Muñoz Laredo https://orcid.org/0009-0003-3400-582X

References

[1] Foulkes WMC, Mitas L, Needs RJ, Rajagopal G. *Quantum Monte Carlo simulations of solids*. Rev Mod Phys. 2001; 73:33.
[2] Hermann J, Schätzle Z, Noé F. *Deep-neural-network solution of the electronic Schrödinger equation*. Nat Chem. 2020; 12:891–897.
[3] Pfau J, Spencer JS, Matthews AGDG, Foulkes WMC. *Ab initio solution of the many-electron Schrödinger equation with deep neural networks*. Phys Rev Research. 2020; 2:033429.
[4] von Glehn I, Spencer JS, Pfau D. *A Self-Attention Ansatz for Ab-Initio Quantum Chemistry*. International Conference on Learning Representations (ICLR); 2023.
[5] Kato T. *On the eigenfunctions of many-particle systems in quantum mechanics*. Commun Pure Appl Math. 1957; 10(2):151–177.
[6] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. *Equation of state calculations by fast computing machines*. J Chem Phys. 1953; 21(6):1087–1092.
[7] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. *Attention Is All You Need*. Advances in Neural Information Processing Systems (NeurIPS); 2017. Vol. 30.
[8] Loshchilov I, Hutter F. *Decoupled Weight Decay Regularization*. International Conference on Learning Representations (ICLR); 2019.
[9] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Advances in Neural Information Processing Systems (NeurIPS); 2019.