nature computational science

Article

https://doi.org/10.1038/s43588-025-00872-z

# Transferable neural wavefunctions for solids

Received: 4 June 2024

Accepted: 20 August 2025

Published online: 22 October 2025

Check for updates

L. Gerard [1,4], M. Scherbela [1,4], H. Sutterud[2,4], W. M. C. Foulkes [2] & P. Grohs [1,3]

Deep-learning-based variational Monte Carlo has emerged as a highly accurate method for solving the many-electron Schrödinger equation. Despite favorable scaling with the number of electrons, $\mathcal{O}({n_{\mathrm{el}}}^4)$, the practical value of deep-learning-based variational Monte Carlo is limited by the high cost of optimizing the neural network weights for every system studied. Recent research has proposed optimizing a single neural network across multiple systems, reducing the cost per system. Here we extend this approach to solids, which require numerous calculations across different geometries, boundary conditions and supercell sizes. We demonstrate that optimization of a single ansatz across these variations significantly reduces optimization steps. Furthermore, we successfully transfer a network trained on $2 \times 2 \times 2$ supercells of LiH, to $3 \times 3 \times 3$ supercells, reducing the number of optimization steps required to simulate the large system by a factor of 50 compared with previous work.

Many interesting material properties, such as magnetism and superconductivity, depend on the material's electronic structure as given by the ground-state wavefunction. The wavefunction may in principle be found by solving the time-independent Schrödinger equation, but doing so with sufficient accuracy is challenging because the computational cost grows dramatically with the number of particles. The challenge is particularly pronounced in solid state physics, where accurate calculations for periodic systems require the use of large supercells—and, consequently, many particles—to minimize finite-size effects.

Over the past few decades, density functional theory (DFT) has emerged as the primary workhorse of solid-state physics. When using local or semi-local exchange–correlation functionals, DFT calculations have a favorable scaling of $\mathcal{O}({n_{\mathrm{el}}}^3)$ or better, where $n_{\mathrm{el}}$ is the number of electrons in the system, and an accuracy that is often sufficient to help guide and predict experiments[1,2]. However, the choice of functional is in practice an uncontrolled approximation, and DFT sometimes yields quantitatively or even qualitatively wrong results, especially for strongly correlated materials[3,4].

Another approach, known as variational Monte Carlo (VMC), uses an explicit parameterized representation of the full many-body wavefunction and optimizes the parameters using the variational principle. This method has a favorable scaling of $\mathcal{O}({n_{\mathrm{el}}}^{3-4})$ (refs. 5,6) but is limited in accuracy by the expressivity of the ansatz used. Recently, deep neural networks have been used as wavefunction ansatze[6–8] and used to study a large variety of systems including small molecules[6,9,10], periodic model systems described by lattice Hamiltonians[7,11–13], the homogeneous electron gas[14,15] and Fermi liquids[16,17]. Due to their flexibility and expressive power, deep-learning-based VMC (DL-VMC) approaches provide the best current estimates for the ground-state energies of several small molecules[9,10] In DL-VMC, the wavefunction ansatz $\psi_\theta$ is represented as a neural network, with the variational parameters $\theta$ being the network weights and biases. An approximation of the ground state is obtained by minimizing the energy expectation value of this ansatz (Fig. 1a). In each optimization step, electron coordinates $\mathbf{r}$ are sampled from the probability density $|\psi_\theta|^2$, and these samples are used to estimate the energy expectation value $E_\theta$. Using automatic differentiation, the energy gradient is computed, and the network parameters $\theta$ are updated to minimize this energy.

Despite the success for small molecules, efforts to apply DL-VMC to real solids[18,19] have been limited by the high computational cost involved. While a single calculation may be feasible, studying real solids requires many similar but distinct calculations. First, it is necessary to perform calculations involving increasingly larger supercells to estimate finite-size errors and extrapolate results to the thermodynamic

[1]Faculty of Mathematics, University of Vienna, Vienna, Austria. [2]Department of Physics, Imperial College London, London, UK. [3]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria. [4]These authors contributed equally: L. Gerard, M. Scherbela, H. Sutterud. e-mail: philipp.grohs@univie.ac.at

https://doi.org/10.1038/s43588-025-00872-z

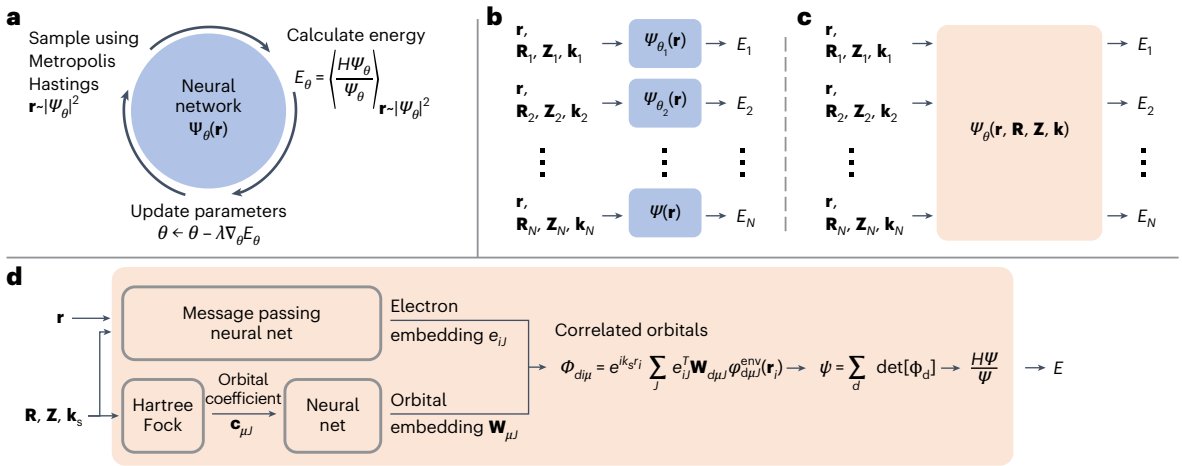


**Fig. 1 | Schematic overview of our approach. a**, A schematic overview of the VMC optimization loop. **b**, The conventional approach of training separate, geometry- and twist-specific wavefunctions. **c**, Our approach of training a single, transferable wavefunction across system variations. **d**, A schematic of our transferable wavefunction ansatz: starting from electron and nucleus coordinates, nucleus charges $\mathbf{r}$, $\mathbf{R}$, $\mathbf{Z}$ and twist $\mathbf{k}_s$, we compute high-dimensional representations $\mathbf{e}_i$ for each electron $i$ and $\mathbf{W}_\mu$ for each orbital $\mu$. We combine them to square matrices $\Phi_d$ and use determinants to obtain an antisymmetric wavefunction $\Psi_\theta$ with trainable parameters $\theta$. From that, we compute the total energy $E$, applying the Hamiltonian operator $H$ to the wavefunction.

limit (TDL). Second, twist-averaged boundary conditions (TABC) are used to accelerate the rate at which the finite-size errors reduce as the supercell size increases[20]. This requires averaging the results for each supercell over many calculations using different boundary conditions. Lastly, studying a given system often requires calculations for different geometries and lattice constants. As most existing DL-VMC ansatze require optimizing a new wavefunction from scratch for each new system (Fig. 1b), the computational cost quickly becomes prohibitive even for systems of moderate size. For example, Li et al. proposed DeepSolid[18], an ansatz capable of accurately modeling periodic wavefunctions with up to 100 electrons, but it required over 80,000 GPU hours to study a single system.

In this work, we implement a transferable DL-VMC ansatz for real solids that takes as input not only the electron positions but also other parameters of the system, such as its geometry or boundary condition. When computing energies for multiple systems, we do not optimize separate ansatze for each system, but instead optimize a single wavefunction able to represent all these systems (Fig. 1c). The transferability of this wavefunction across systems yields two large speed-ups in practice. First, optimizing a single ansatz for many variations of unit-cell geometry, boundary condition and supercell requires typically much fewer optimization steps than optimizing ansatze separately for each system. Second, because the ansatz learns to generalize across systems, we can use models pretrained on small systems as highly effective initializers for new systems or larger supercells. The key idea, based on Scherbela et al.[21], sketched in Fig. 1d and detailed in the Methods, is to map computationally cheap, uncorrelated mean-field orbitals to expressive neural network orbitals that depend on the positions of all electrons.

Compared with previous DL-VMC work without transferability, our approach yields more accurate results, gives access to denser twist averaging (reducing finite-size effects) and requires a fraction of the computational resources. For example, for lithium hydride, transferring a 32-electron calculation to one with 108 electrons yields more accurate results than previous work[18] at approximately 1/50 of the computational cost.

## Results

### One-dimensional hydrogen chains

Chains of hydrogen atoms with periodic boundary conditions provide a simple one-dimensional toy system that nevertheless exhibits rich physics such as dimerization, a lattice-constant-dependent metal–insulator transition and strong correlation effects. A collaborative effort[3,22] has obtained results for this system using a large variety of high-accuracy methods, providing a trustworthy benchmark.

The first test is to obtain the total energy per atom for a fixed atom spacing, $R = 1.8a_0$ (where $a_0$ is the Bohr radius), in the TDL attained as the number of atoms in the supercell tends to infinity. To this end, we train two distinct models on periodic supercells with $N_{\text{atoms}} = 4, 6, \ldots, 22$. The first model is trained at twist $k = 0$ (the $\Gamma$-point) only. The second is trained using all twists from a $\Gamma$-centered four-point Monkhorst–Pack grid[23]. The three inequivalent twists are $k = 0$, $\frac{1}{4}$ and $\frac{1}{2}$ in units of $2\pi/R$, and their weights are $w = 1, 2$ and 1, respectively. Once the model has been pretrained on these relatively short chains, we fine-tune it on larger chains with $N_{\text{atoms}} = 32$ and 38. We use the extrapolation method described in ref. 22 to obtain the energy $E_\infty$ in the TDL. Previous authors have extrapolated the energy using only chain lengths of the form $N_{\text{atoms}} = 4n + 2$, $n \in \mathbb{N}$, which have filled electronic shells. We also report extrapolations using chain lengths $N_{\text{atoms}} = 4n$, which lead to partially filled shells.

Figure 2a shows that all of our extrapolations ($\Gamma$-point filled shells, $\Gamma$-point unfilled shells and TABC) are in good qualitative agreement with previous results obtained using methods such as lattice-regularized diffusion Monte Carlo (LR-DMC)[22] and DeepSolid[18], a FermiNet-based[6] neural wavefunction for solids. Quantitatively, we achieve slightly lower (and, thus, more accurate) energies than DeepSolid for all values of $N_{\text{atoms}}$. Using TABC, we obtain $E_\infty = -565.24(2)$ mHa, which is 0.2–0.5 mHa lower than the estimate obtained using LR-DMC and DeepSolid, and agrees within uncertainty with the extrapolated energy computed using the auxiliary-field quantum Monte Carlo (AFQMC) method[22]. Most notably, however, we obtain these results at a fraction of the computational cost of DeepSolid. Whereas DeepSolid required a separate calculation with 100,000 optimization steps for each value of $N_{\text{atoms}}$ (and would have required even more calculations for twist-averaged energies), we obtain results for all 10 chain lengths and values of $N_{\text{atoms}} = 4, \ldots, 22$, with 3 twists for each system, using only 50,000 optimization steps in total. Furthermore, by reusing the model pretrained on smaller chains, we obtain results for the larger chains with $N_{\text{atoms}} = 32$ and 38 using only 2,000 additional steps of fine tuning. This reduces the cost of simulating the large chains by a factor of approximately 50. We note that, as expected, the use of TABC reduces finite-size errors, allows us to combine results for filled and unfilled

https://doi.org/10.1038/s43588-025-00872-z

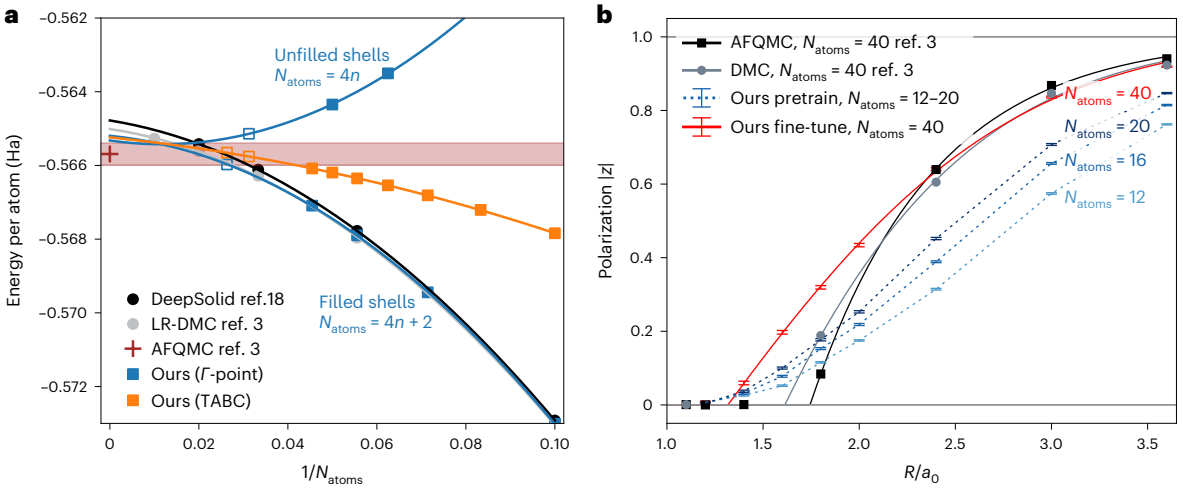


**Fig. 2 | One-dimensional hydrogen chain. a**, Extrapolation of the energy per atom to the TDL for $R = 1.8a_0$. Results obtained using DeepSolid (neural wavefunction), LR-DMC, AFQMC and our transferable neural wavefunction are shown. Open markers indicate energies computed by fine-tuning a model pretrained on smaller supercells. The shaded area depicts the statistical uncertainty in the AFQMC result. The Monte Carlo uncertainty of our results is approximately 10 μHa, well below the marker size. **b**, The complex polarization $|z|$ as a function of the interatomic separation, $R$, showing a phase transition between a metal at small $R$ and an insulator at large $R$. AFQMC and DMC results are taken from the work of the Simons Collaboration[3]. The error bars for our results represent Monte Carlo uncertainty. DeepSolid results are taken from Li et al.[18].

shells in the extrapolation and leads to faster convergence of the energy per atom. By contrast, when using only $\Gamma$-point calculations, there is a strong even/odd effect in the energy, requiring separate extrapolations for unfilled and filled shells.

Beyond energies, we study the hydrogen chain's phase transition from an insulating phase at large interatomic separation, $R$, to a metallic phase at small $R$. The transition can be quantified by evaluating the complex polarization along the length of the chain

$$z = \left\langle e^{i\frac{2\pi}{RN_{atoms}}\sum_{i=1}^{n_{el}} x_i} \right\rangle, \quad (1)$$

where $x_i$ is the position of electron $i$ in the direction of the chain. The expectation value is defined as $\langle \ldots \rangle \equiv \int \Psi^*(\mathbf{r}) \ldots \Psi(\mathbf{r}) d\mathbf{r}$, where $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{n_{el}})$ is a $3n_{el}$-dimensional vector of electron positions, $\Psi$ is the (approximate) ground-state wavefunction, and the integral is over all $3n_{el}$ electronic degrees of freedom. Although the polarization is easy to evaluate in principle, studying the transition is computationally costly because it requires many similar but distinct calculations: multiple values of $R$ are required to locate the transition; multiple twists $k$ are required to obtain accurate twist-averaged polarizations; and multiple chain lengths $N_{atoms}$ are required to allow extrapolation to the TDL. Even for a modest selection of all of these variations, studying the phase transition in detail requires hundreds of calculations. Using our transferable wavefunction, on the other hand, allows us to train a single model to represent the wavefunction for all parameter variations at once.

We trained a single ansatz to describe all 120 combinations of: (1) 3 distinct chain lengths, $N_{atoms} = 12$, 16 and 20; (2) 5 symmetry-reduced $k$-points of an 8-point $\Gamma$-centered Monkhorst–Pack grid; and (3) 8 distinct atom spacings between $R = 1.2a_0$ and $R = 3.6a_0$. A total of 200,000 optimization steps were carried out, after which the complex polarization was evaluated using equation (1). To improve our estimates for $N_{atoms} \to \infty$, we fine-tuned this pretrained model for 2,000 steps on chain lengths of $N_{atoms} = 40$ and a denser 20-point Monkhorst–Pack grid containing 11 symmetry-reduced twists. Figure 2b shows that our approach qualitatively reproduces the results obtained using DMC and AFQMC. In agreement with Motta et al.[3], we observe a second-order metal–insulator transition. However, where Motta estimates the critical atom spacing $R_{crit} = 1.70(5)a_0$, our results are more consistent with $R_{crit} = 1.32(5)a_0$. A possible explanation for the disagreement is that our neural wavefunction may be less accurate (and may therefore produce relatively higher energies) for metals than insulators, disfavoring the metallic phase. Another possible explanation follows from the observation that, unlike the VMC method used here, the DMC and AFQMC methods yield biased estimates of the expectation values of operators, such as the complex polarization, that do not commute with the Hamiltonian[5,24].

Also in agreement with Motta et al.[3], we find that the hydrogen chain shows quasi-long-range antiferromagnetic correlation at at large lattice constant $R$. The expected atomic spins are zero on every atom, but the spins on neighboring atoms are antiferromagnetically correlated. As the lattice constant gets smaller and the system transitions to the metallic phase, these correlations decrease as shown in Supplementary Fig. 3.

## Graphene

To demonstrate the application of our transferable DL-VMC ansatz to a two-dimensional solid, we compute the cohesive energy of graphene in a 2 × 2 supercell and compare against the DL-VMC results of DeepSolid by Li et al.[18]. We use TABC, apply structure-factor-based finite-size corrections[25] as detailed in the Methods and add zero-point vibrational energies (ZPVE). The DeepSolid results were restricted to a Monkhorst–Pack grid of 3 × 3 twists, yielding three symmetry-reduced twists in total. In our case, because we are able to compute multiple twists at once with minimal extra cost, we increase the grid density to 12 × 12. This increases the number of symmetry-reduced twists from 3 to 19. Our denser twist grid contains a subset of the twists considered by DeepSolid, allowing a direct comparison with their independent energy calculations. We stress that we require only a single neural network, optimized for 120,000 steps, to obtain energies for all twists (both the 12 × 12 grid and the 3 × 3 subset). DeepSolid, on the other hand, optimized for 900,000 steps in total, obtaining energies only for the 3 × 3 twist grid. We find that our transferable ansatz has an approximately 2× higher per-step cost compared with DeepSolid (Supplementary Fig. 2), but the large reduction in the required number of optimization steps (Fig. 3a) far outweighs this cost.

https://doi.org/10.1038/s43588-025-00872-z

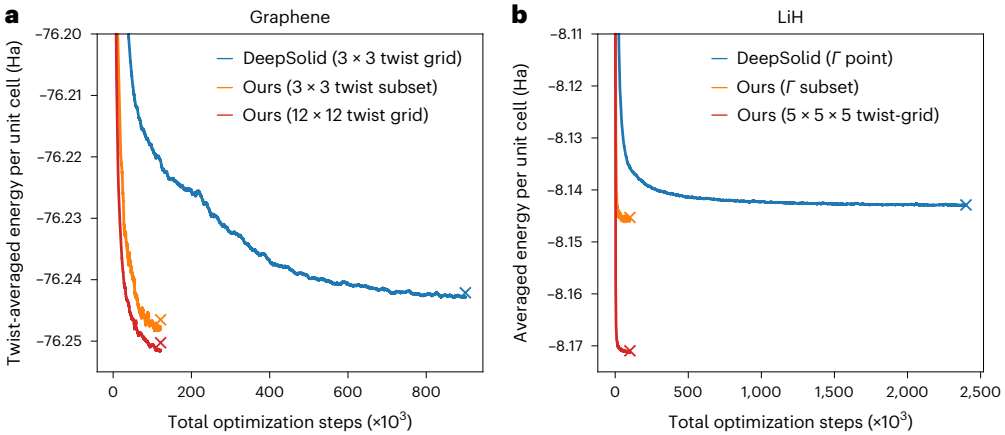


**Fig. 3 | Optimization curves.** Mean energy as a function of total optimization steps across all geometries and twists. Energies are the running average over the last 1,000 steps. Crosses mark final evaluation energies. **a**, Energy of 2 × 2 supercell of graphene. **b**, Mean energy of the potential energy surface of LiH in a 2 × 2 × 2 supercell.

Our twist-averaged energy using the 3 × 3 twist grid is 4 mHa per primitive cell lower than the DeepSolid energy. Looking at individual twists (Table 1), we find that our energies for $k_1$ and $k_2$ are lower than the energies obtained by DeepSolid by 1 mHa and 7 mHa, respectively, while our energy for $k_3$ is higher by 4 mHa. This twist-dependent accuracy is expected, because we allocate optimization steps proportional to the twist's symmetry weight (see 'Sampling' in the Methods), thereby potentially optimizing $k_1$ and $k_2$ more stringently than $k_3$. This procedure ensures that more optimization steps are spent on twists with high contribution to the final energy, thus improving efficiency.

To check for finite size effects, we also compute cohesive energies on a larger 3 × 3 supercell with a 12 × 12 twist grid. Due to the transferability of our wavefunction, we can use wavefunction parameters obtained from the 2 × 2 supercell as initialization for the 3 × 3 supercell calculation, thereby reducing the number of required optimization steps.

When computing cohesive energies and correcting for finite-size effects using a structure-factor-based correction and ZPVE, we obtain energies that are 7 mHa lower than experimental values for the 2 × 2 supercell, that is, we predict slightly stronger binding than experiment. For the 3 × 3 supercell, we predict 15 mHa higher energies than experimental values (Supplementary Table 1). We hypothesize that the remaining discrepancy may be a finite-size artifact and that even for the 3 × 3 supercell energies may not yet be converged. An alternative hypothesis is that a larger, more expressive network may be needed to represent the true ground-state wavefunction for the 3 × 3 supercell.

With a network that has been trained across the entire Brillouin zone, we can evaluate observables along arbitrary paths in $k$ space. Figure 4 is a bandstructure-like diagram, showing how the total energy varies along a path passing through the high-symmetry $k$-points $\Gamma = (0, 0)$, $M = (0, 1/2)$ and $K = (1/3, 2/3)$ in units of the supercell reciprocal lattice vectors. We use the pretrained model from the 12 × 12 Monkhorst–Pack grid and transfer it to the bandstructure-like diagram with $k$-points previously unseen during optimization, requiring only a few additional optimization steps. We fine-tune the pretrained model for the $k$-points on the path, using around 100 optimization steps per twist and then evaluate the energies along the path. Analogously to the Dirac cone visible in the one-electron bandstructure, our many-electron bandstructure displays a characteristic cusp at the $K$-point.

## Lithium hydride

We have also used the transferable DL-VMC ansatz to evaluate the energy–volume curve of LiH in the rock-salt crystal structure. As shown in Fig. 5 (see also Supplementary Section 7), we obtain the energy–volume curve by fitting a Birch–Murhaghan equation of state to the total energies of a 2 × 2 × 2 supercell at eight different lattice parameters. To reduce finite-size errors, the eight total energies are twist averaged using a 5 × 5 × 5 $\Gamma$-centered Monkhorst–Pack grid and include structure-factor-based finite-size corrections. For comparison, DeepSolid performed a $\Gamma$-point calculation only and estimated finite-size errors by converging a Hartree–Fock calculation with an increasingly dense twist grid[18]. To all results we add ZPVE taken from ref. 26, making the calculated cohesive energy less negative by approximately 8 mHa. The DeepSolid results by Li et al.[18] took no account of the ZPVE, explaining the slight difference between our depiction of their results, shown in Fig. 5, and their original publication[18].

We trained a single neural network wavefunction across 8 lattice constants and 10 symmetry-reduced twists, making 80 systems in total. By comparison, DeepSolid required a separate calculation for each geometry.

The Birch–Murnaghan fit gives an equilibrium lattice constant of $7.66(1)a_0$ (dotted orange line), which agrees well with the experimental value of $7.674(2)a_0$ (ref. 26). Our Birch–Murnaghan estimate of the cohesive energy of −177.3(1) mHa per primitive cell deviates from the experimental value of −175.3(4) mHa by −2.0(5) mHa. This marks an improvement over the DeepSolid results[18] of −166.8(1) mHa, which differ from experiment by 8.5(5) mHa. Because we are able to optimize all systems at once, our results were obtained with roughly 5% of the compute required by DeepSolid, and the speed-up is evident in Fig. 3b. Similar improvements can be observed in the variance of the local energy (Supplementary Section 8).

**Table 1 | Total energies of graphene in Hartrees for a primitive cell, as computed by VMC, after the structure factor correction (SFC) and after adding ZPVE**

| | Twist | Weight | Total energy | Total energy + SFC | Total energy + SFC + ZPVE |
|---|---|---|---|---|---|
| DeepSolid | $k_1 = (0, 0)$ | 1/9 | −76.1559 | −76.1534 | −76.1406 |
| | $k_2 = (1/3, 1/3)$ | 2/3 | −76.2495 | −76.2470 | −76.2342 |
| | $k_3 = (2/3, 1/3)$ | 2/9 | −76.2631 | −76.2607 | −76.2479 |
| Our work | $k_1 = (0, 0)$ | 1/9 | −76.1572(2) | −76.1542(2) | −76.1414(2) |
| | $k_2 = (1/3, 1/3)$ | 2/3 | −76.2572(2) | −76.2543(2) | −76.2415(2) |
| | $k_3 = (2/3, 1/3)$ | 2/9 | −76.2590(2) | −76.2560(2) | −76.2432(2) |

The table compares our results against the total energies computed with DeepSolid[18] at the three symmetry-inequivalent twists on the 3×3 Monkhorst–Pack grid. The twists are expressed in the basis of the reciprocal lattice vectors.

https://doi.org/10.1038/s43588-025-00872-z

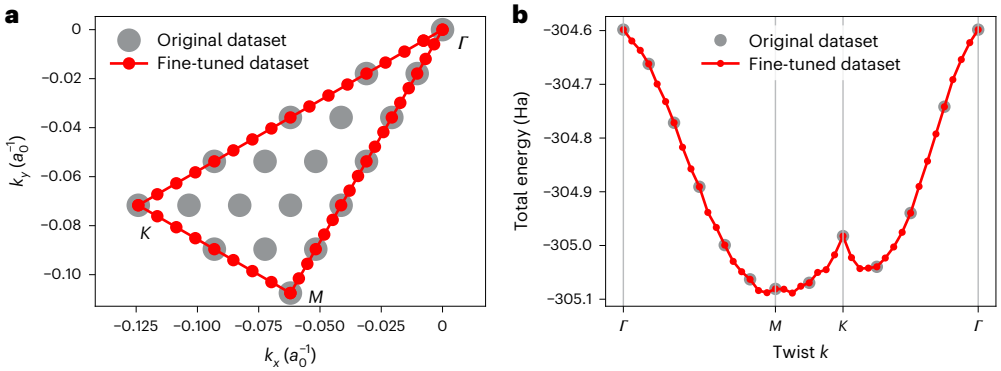


**Fig. 4 | Twist-dependent energy of Graphene. a**, Grid of pretrained twists and path of fine-tuned values through Brillouin zone. **b**, Fine-tuned energies of graphene along the path of twists across the Brillouin zone, computed using shared optimization and around 100 additional optimization iterations per twist. The error bars are smaller than the size of the markers.

Although we improve on the DeepSolid baseline, the cohesive energy might still be impacted by finite-size effects because of the small size of the 2 × 2 × 2 supercell used. To check this, we also studied a larger supercell containing 3 × 3 × 3 primitive unit cells. This 108-electron system is one of the largest to have been studied using neural wavefunctions so far. DeepSolid used 400,000 optimization steps to get a $\Gamma$-point estimate for the cohesive energy and overestimated the energy by around 7 mHa per primitive cell compared with the experimental results[18,26]. By contrast, we can exploit the transferability of our wavefunction and use the parameters obtained from pretraining on the 2 × 2 × 2 supercells as initialization for the much larger 3 × 3 × 3 supercell. Due to the good generalization of our ansatz, we are able to calculate the cohesive energy for the 3 × 3 × 3 supercell with only 8,000 additional optimization steps shared across ten different twists. Using twist averaging, a structure-factor correction and a ZPVE correction as before, we obtain a cohesive energy of −174.6 mHa per primitive cell, deviating from experiment by only 0.7(5) mHa per primitive cell. The magnitude of this deviation is close to the 0.4-mHa spread of experimental data obtained from different thermochemistry experiments[26]. Our twist-averaged 3 × 3 × 3 calculation required only ~2% of the computational resources used by DeepSolid for a single $\Gamma$-point calculation[18].

Furthermore, we compared our approach to the case of pretraining on a single system and fine-tuning the pretrained wavefunction on the remaining systems with independent calculations (similar to DeepSolid) for a 2 × 2 × 2 supercell of LiH. This comparison confirms that the approach of training a single neural network wavefunction across different systems converges much faster then fine-tuning independent wavefunctions (Supplementary Fig. 7).

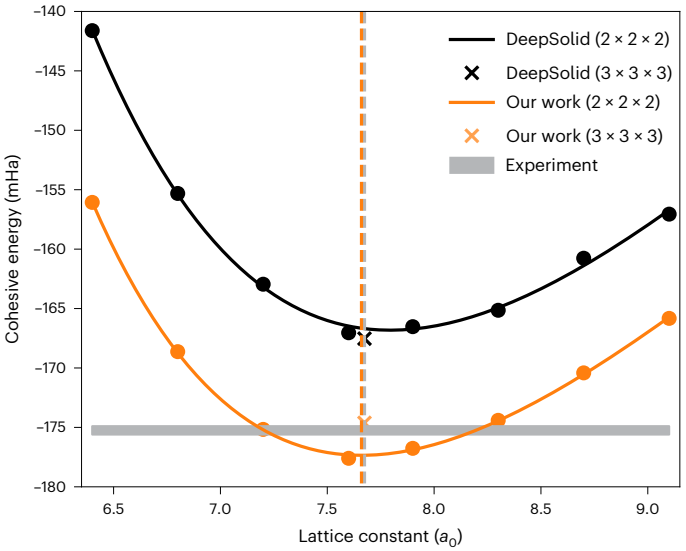


**Fig. 5 | Energy–volume curve of LiH per primitive cell.** The curve is for a 2 × 2 × 2 supercell as calculated using DeepSolid[18] and our transferable DL-VMC method. The DeepSolid results (black circles, with a Birch–Murnaghan fit represented as a black line) were obtained at a single twist, the $\Gamma$-point. Hartree–Fock corrections were applied, as discussed in ref. 18, and a ZPVE correction was added. Our results (orange circles, with a Birch–Murnaghan fit represented as an orange line) are twist averaged, using a 5 × 5 × 5 Monkhorst–Pack grid per lattice constant. Structure-factor-based corrections were applied, and a ZPVE correction was added. The gray bar indicates the experimental uncertainty[26]. The statistical error bars are too small to be visible on this scale and therefore have been omitted. The vertical dashed orange line indicates the equilibrium lattice constant as calculated from the Birch–Murnaghan fit to our data. The vertical dashed gray line indicates the experimental value of the equilibrium lattice constant[26]. The orange cross shows the twist-averaged cohesive energy of a 3 × 3 × 3 simulation cell, again using structure factor correction. This was obtained by transferring the network pretrained for the 2 × 2 × 2 system to a 3 × 3 × 3 supercell, using only 8,000 additional optimization steps. A 5 × 5 × 5 Monkhorst–Pack grid of twists was used. The black cross shows the result of DeepSolid's 3 × 3 × 3 $\Gamma$-point calculation with a Hartree–Fock finite-size correction.

## Discussion

By training a single transferable wavefunction across system sizes, geometries and boundary conditions, our approach substantially reduces the computational cost of applying DL-VMC to solids. Combining this approach with other acceleration techniques—such as the efficient forward evaluation of the Laplacian by Li et al.[27] or pseudo-potentials[28]—might enable the study of strongly correlated materials with DL-VMC. Our approach could also be extended to grand-canonical twist averaging[29], in which the number of electrons in the supercell varies with the twist. Because our ansatz already supports a variable number of particles, this extension should be easy to incorporate.

Our approach shares many of the limitations of other DL-VMC methods, including the sensitivity with regard to MCMC initialization. A standard practice in DL-VMC is to assign each electron a spin and initialize it close to the nuclei at the beginning of the calculation. If the electrons are initialized in an anti-ferromagnetic pattern, that is, alternating the spins of neighboring atoms, but the ground state is ferromagnetic, as can be the case for the hydrogen chain when the interatomic separation is small, our approach tends to converge to local minima. FermiNet suffers from similar problems.

Another limitation arises from the allocation of compute budget between the multiple geometries or systems described by a single neural network. We allocate more compute during optimization to twists with a larger weight, which has a positive effect on twist-averaged results in general, because twists with higher contribution are

converged to higher accuracy (Table 1). However for individual twists, when plotting, for example, the band structure (Fig. 4), not all twists are optimized to the same accuracy, potentially skewing results.

While this work demonstrates the transferability of a wavefunction across variations of a system (lattice constant, supercell size and twist), more research is needed to develop wavefunctions that reliably transfer to entirely new systems, such as different compositions or lattices. Prior work on molecules in the gas phase has shown that it is possible to pretrain a single wavefunction on a diverse set of molecules and transfer the results to new, unseen molecules[30]. There are several open challenges to applying this approach to solids: First, the effectiveness decreases when transferring the pretrained model to systems substantially larger than those in the pretraining set. This issue is particular problematic for solids, where finite-size scaling may often require transferability to large systems. Second, successful pretraining typically requires wavefunction optimization for a large, diverse set of systems. This is challenging for calculations of solids, which are inherently more costly than calculations for molecules owing to the need for supercells. In practice, while pretraining on hundreds of qualitatively different systems is achievable on a moderate compute budget for gas-phase molecules, this scale is currently out of reach for solids.

# Methods

## Notation

All vectors, matrices and tensors are denoted by bold letters, except for functions. We use lower-case indices $i,j = 1,\ldots,n_{\mathrm{el}}$ for electron positions and upper-case indices $I,J = 1,\ldots,N_{\mathrm{atoms}}$ for atom positions, where $n_{\mathrm{el}}$ and $N_{\mathrm{atoms}}$ are the numbers of electrons and atoms in the supercell. Orbitals are enumerated by the indices $\mu$ and $\nu$, which range from 1 to $n_{\mathrm{el}}$. The position of the $i$th electron is $\boldsymbol{r}_i \in \mathbb{R}^3$. When $i$ is not used as a subscript, it denotes the imaginary unit. By $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_{n_{\mathrm{el}}})$, we denote the $3n_{\mathrm{el}}$-dimensional vector of all electron positions. Similarly, nuclear positions and charges are represented by $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_{N_{\mathrm{atoms}}})$ and $\mathbf{Z} = (Z_1, \ldots, Z_{N_{\mathrm{atoms}}})$. The matrix $L \in \mathbb{R}^{3\times3}$ contains the supercell lattice vectors in its columns. The twist vector, which may always be reduced into the first Brillouin zone of the supercell, is denoted by $\mathbf{k}_{\mathrm{s}}$. The dot product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is written $\mathbf{a}\cdot\mathbf{b}$, and by $\odot$ we refer to the element-wise multiplication (Hadamard product).

## Deep-learning VMC

The time-independent Schrödinger equation for a solid takes the form

$$\hat{H}\Psi = E\Psi, \qquad \hat{H} = -\frac{1}{2}\sum_i \nabla^2_{\mathbf{r}_i} + \hat{V}_{\mathrm{Coulomb}} \tag{2}$$

with the Hamiltonian in the Born–Oppenheimer approximation and Coulomb potential $\hat{V}_{\mathrm{Coulomb}}$. A finite supercell is used to approximate the bulk solid, and the Coulomb potential is evaluated using the Ewald method, as described in refs. 14,31.

In this work, we are interested in finding the lowest eigenvalue of the Schrödinger equation—the ground-state energy, $E_0$—and the corresponding energy eigenfunction. To find an approximate solution, one can reformulate the Schrödinger equation as a minimization problem using the Rayleigh–Ritz variational principle. Given an arbitrary anti-symmetric trial wavefunction, $\Psi_{\boldsymbol{\theta}}$, with $\boldsymbol{\theta}$ denoting, for example, the trainable parameters of a neural network, the best attainable approximation to the ground state may be found by minimizing the energy expectation value

$$L(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{r}\sim|\Psi_{\boldsymbol{\theta}}|^2}\left[\frac{\hat{H}\Psi_{\boldsymbol{\theta}}}{\Psi_{\boldsymbol{\theta}}}\right] \geq E_0 \tag{3}$$

with respect to $\boldsymbol{\theta}$. An important constraint for the construction of the trial wavefunction arises from the Pauli exclusion principle, which states that the wavefunction must be antisymmetric with respect to the permutations of different electron coordinates[6]. As in previous work, we approximate the expectation value in equation (3) using Monte Carlo integration with samples drawn from the $3n_{\mathrm{el}}$-dimensional probability density $|\Psi_{\boldsymbol{\theta}}(\mathbf{r})|^2$ (refs. 6,8).

A list of all relevant hyperparameters can be found in Supplementary Table 2.

## Architecture

**Overview.** Our ansatz can be broken down into the computation of periodic input features, the computation of embeddings $\mathbf{e}_{ij}$ for each electron–nucleus pair, the computation of correlated orbitals and the assembly of the final wavefunction $\Psi_\theta$ as a sum of Slater determinants. Each step serves a distinct purpose.

The input features enforce the periodic boundary conditions of the supercell. To capture correlation effects, we use a neural network to map single-electron coordinates to vectors in a latent space. These vectors, also known as embeddings, depend on the positions of all of the other electrons in a permutation equivariant way. Each embedding therefore contains information about the corresponding electron as well as its environment. The embeddings are subsequently mapped to many-electron orbitals as outlined below.

**Ansatz.** Our wavefunction ansatz is a sum of Slater determinants multiplied by a Jastrow factor,

$$\Psi(\mathbf{r},\mathbf{R},\mathbf{Z},\mathbf{k}_{\mathrm{s}}) = e^{J(\mathbf{r})}\sum_{d=1}^{n_{\mathrm{det}}}\det\Phi_d(\mathbf{r},\mathbf{R},\mathbf{Z},\mathbf{k}_{\mathrm{s}}). \tag{4}$$

The optimization is free to adjust the relative normalizations of the determinants in the unweighted sum, making it equivalent to a weighted sum of normalized determinants, as might be used in a configuration-interaction expansion. The Jastrow factor $e^{J(\mathbf{r})}$ is node-less and follows the work of Hermann et al.[8], while the determinant enforces the fermionic antisymmetry. Instead of using single-particle orbitals in the determinant, as in most quantum chemical approaches, we follow other neural wavefunction methods[6] and promote every entry $\Phi_{d,i\mu}$ in the orbital matrix $\Phi_d$ from a one-electron orbital, $\phi_{d,\mu}(\mathbf{r}_i)$, to a many-electron orbital, $\Phi_{d,i\mu}(\mathbf{r})$ (temporarily dropping the dependency on $\mathbf{R}$, $\mathbf{Z}$ and $\mathbf{k}_{\mathrm{s}}$ for the sake of brevity). The many-electron orbitals are permutation equivariant, such that applying a permutation $\pi$ to the electron position vectors permutes the rows of $\Phi_d$ by $\pi$, that is, $\Phi_{d,i\mu}(\mathbf{r}_{\pi(1)},\ldots,\mathbf{r}_{\pi(n_{\mathrm{el}})}) = \Phi_{d,\pi(i)\mu}(\mathbf{r}_1,\ldots,\mathbf{r}_{n_{\mathrm{el}}})$This ensures that the determinant has the correct fermionic symmetry. Each entry is constructed as a linear combination of atom-centered functions with permutation equivariant dependencies on both electrons and atoms

$$\Phi_{d,i\mu}(\mathbf{r},\mathbf{R},\mathbf{Z},\mathbf{k}_{\mathrm{s}}) = e^{i\mathbf{k}_{\mathrm{s}}\cdot\mathbf{r}_i}\sum_{J=1}^{N_{\mathrm{atoms}}}\varphi_{d\mu}(\mathbf{r}_i,\{\mathbf{r}\},\mathbf{R}_J,\{\mathbf{R}\}). \tag{5}$$

Here, $\{\mathbf{r}\}$ and $\{\mathbf{R}\}$ denote the (permutation invariant) set of electron and atom positions, respectively. The phase factor enforces the twisted boundary conditions. To construct the $\varphi_{d\mu iJ} \equiv \varphi_{d\mu}(\mathbf{r}_i,\{\mathbf{r}\},\mathbf{R}_J,\{\mathbf{R}\})$ using a neural network, we use an adaptation of the recently proposed transferable atomic orbital ansatz[21,30]. The orbitals are written as the inner product of an electron–nuclear embedding $\boldsymbol{e}_{ij} \in \mathbb{R}^{n_{\mathrm{emb}}}$ and an orbital embedding $\mathbf{W}_{d\mu J} \in \mathbb{C}^{n_{\mathrm{emb}}}$, multiplied by an exponential envelope $\varphi^{\mathrm{env}}_{d\mu J}$,

$$\varphi_{d\mu iJ} = (\mathbf{W}_{d\mu J}\cdot\mathbf{e}_{ij})\varphi^{\mathrm{env}}_{d\mu J}(\mathbf{r}_i) \tag{6}$$

$$\varphi^{\mathrm{env}}_{d\mu J}(\mathbf{r}_i) = e^{-a_{d\mu J}||\mathrm{L}^{-1}\mathbf{r}_{ij}||^{\mathrm{per}}}, \tag{7}$$

where $a_{d\mu J}$ is a learnable decay rate, $\mathbf{s}_{ij}$ is the vector from nucleus $J$ to electron $i$, expressed in the basis of the supercell lattice vectors, and $||\mathbf{s}_{ij}||^{\mathrm{per}}$ is the modulus of $\mathbf{s}_{ij}$ in a periodic norm explained below. Both the orbital embedding $\mathbf{W}_{d\mu J}$ and the decay length $a_{d\mu J}$ depend on the orbital $\mu$ and atom $J$ and are different for each determinant $d$.

To obtain $\mathbf{W}_{d\mu J}$ and $a_{d\mu J}$ in a transferable way, we do not parameterize them directly but represent them as functions of some orbital-specific descriptor $\tilde{\mathbf{c}}_{\mu J} \in \mathbb{R}^{d_{\text{orb}}}$:

$$\mathbf{W}_{d\mu J} = f_d^{\text{W}}(\tilde{\mathbf{c}}_{\mu J}), \qquad a_{d\mu J} = f_d^{\text{a}}(\tilde{\mathbf{c}}_{\mu J}), \tag{8}$$

with $f^{\text{W}} : \mathbb{R}^{d_{\text{orb}}} \to \mathbb{C}^{n_{\text{det}} \times d_{\text{emb}}}$ and $f^{\text{a}} : \mathbb{R}^{d_{\text{orb}}} \to \mathbb{R}^{n_{\text{det}}}$ denoting simple multilayer perceptrons. The orbital embedding includes information about single-particle orbitals of the system calculated with a mean-field method, which is key for the transferability of the ansatz. The inputs are the orbital features $\tilde{\mathbf{c}}_{\mu J} \in \mathbb{R}^{d_{\text{orb}}}$, which are concatenations of the expansion coefficients of the localized mean-field orbitals in an atom-centered basis set, the twist $\mathbf{k}_s$, the mean position of orbital $\mu$ and the position of atom $J$, with a combined dimensionality of $d_{\text{orb}}$. While all parameters and intermediate computations of our network are real-valued, the last layer of $f^{\text{W}}$ is complex-valued to allow the network to represent complex-valued wavefunctions.

An important difference with respect to previous neural network-based wavefunctions is the use of electron–nuclear embeddings $\mathbf{e}_{ij}$, which describe the interaction between electron $i$ and nucleus $J$. Other architectures such FermiNet, but also the more closely related transferable atomic orbital ansatz[21], use embeddings to represent the interactions of a single electron $i$ with all nuclei instead. However, when the embeddings are both invariant under permutation of nuclei (which we require for efficient transferability) and invariant under translation of particles by a supercell lattice vector (which we require to enforce boundary conditions), they become periodic on the primitive lattice (Supplementary Section 4), not just the supercell lattice. This is too restrictive to represent correlation beyond a single primitive cell. We therefore opt to use electron–nucleus embeddings that are equivariant under permutation of nuclei at some additional computational cost explained in Supplementary Section 4.

**Input.** We require our representation of the difference vectors $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{r}_{iI} = \mathbf{r}_i - \mathbf{R}_I$ and $\mathbf{r}_{Ij} = \mathbf{r}_I - \mathbf{R}_J$ to be periodic with respect to the supercell lattice. This is accomplished using the approach introduced by Cassella et al.[14]. The first step is to transform the coordinates into supercell fractional coordinates with $\mathbf{s}_{ij} = L^{-1}\mathbf{r}_{ij}$, $\mathbf{s}_{iI} = L^{-1}\mathbf{r}_{iI}$ and $\mathbf{s}_{Ij} = L^{-1}\mathbf{r}_{Ij}$. Periodic versions of the difference vectors are then obtained by applying sine and cosine element-wise,

$$\omega(\mathbf{s}) := [\sin(2\pi\mathbf{s}), \cos(2\pi\mathbf{s})], \qquad \omega : \mathbb{R}^3 \to \mathbb{R}^6, \tag{9}$$

$$\mathbf{x}_{ij} := \omega(\mathbf{s}_{ij}), \quad \mathbf{x}_{iJ} := \omega(\mathbf{s}_{iJ}), \quad \mathbf{x}_{Ij} := \omega(\mathbf{s}_{Ij}), \tag{10}$$

where square brackets denote the concatenation operator. For the distance, we use the periodic norm

$$(||\mathbf{s}||^{\text{per}})^2 = \sum_{l,p=1}^{3} \left((1-\cos(2\pi s_l))A_{lp}(1-\cos(2\pi s_p)) + \sin(2\pi s_l)A_{lp}\sin(2\pi s_p)\right) \tag{11}$$

for a vector $\mathbf{s} \in \mathbb{R}^3$ with the lattice metric $A := LL^T$. This norm is used to define the periodic distance features

$$x_{ij} = ||\mathbf{s}_{ij}||^{\text{per}}, \quad x_{iJ} = ||\mathbf{s}_{iJ}||^{\text{per}}, \quad x_{Ij} = ||\mathbf{s}_{Ij}||^{\text{per}}. \tag{12}$$

**Embedding.** The periodic input features are used to generate high-dimensional embeddings $\mathbf{e}_{ij}$ for the construction of the orbital matrix. The following embedding is a slight adaption of the appraoch used in the recently proposed Moon architecture[32]. We start by aggregating the electron–electron features into message vectors $\mathbf{m}_i^0$ for each electron $i$

$$\mathbf{m}_i^0 = \sum_{j=1}^{n_{\text{el}}} \Gamma^{\text{e-e}}(x_{ij}, \mathbf{x}_{ij}) \odot \sigma\left(W^{\text{m}}\tilde{\mathbf{x}}_{ij} + \mathbf{b}^{\text{m}}\right), \tag{13}$$

and compute the initial electron embeddings $\mathbf{h}_i^0$ as a trainable function of these messages

$$\mathbf{h}_i^0 = \sigma\left(W^0\mathbf{m}_i^0 + \mathbf{b}^0\right). \tag{14}$$

The matrices $W^{\text{m}}$ and $W^0$ and vectors $\mathbf{b}^{\text{m}}$ and $\mathbf{b}^0$ are trainable parameters, $\sigma$ is an activation function that is applied elementwise and $\odot$ denotes the elementwise product. The filter function $\Gamma^{\text{e-e}}$

$$\Gamma^{\text{e-e}}(x_{ij}, \mathbf{x}_{ij}) = \sigma\left(W^{\text{env}}\mathbf{x}_{ij} + \mathbf{b}\right) \odot \exp\left(-x_{ij}^2\boldsymbol{\alpha}\right), \tag{15}$$

ensures an exponential decay with a trainable vector of length scales $\boldsymbol{\alpha}$ and a trainable matrix $W^{\text{env}}$. Furthermore, the input features $\tilde{\mathbf{x}}_{ij} = [x_{ij}, \mathbf{x}_{ij}, \mathbf{k}_s]$ make the embedding twist dependent to allow for better transferability across twists.

To initialize the atomic features, we first one-hot encode the nuclear charges $\mathbf{Z}$ into a matrix $\tilde{H} \in \mathbb{R}^{N_{\text{atoms}} \times n_{\text{species}}}$. With one-hot encoding we refer to the common machine-learning practice of encoding categorical data (in this case, the type of atom), using a vector that is zero everywhere, except in the one dimension corresponding to the category it encodes. We then initialize the atom embeddings $\mathbf{H}_I^0$ analogously to the electron embeddings, by aggregating atom–atom features for each atom $I$

$$\mathbf{H}_I^0 = \sum_{J=1}^{N_{\text{atoms}}} \Gamma^{\text{a-a}}(x_{IJ}, \mathbf{x}_{IJ}) \odot \sigma\left(W^{\text{a}}\tilde{\mathbf{H}}_J + \mathbf{b}^{\text{a}}\right), \tag{16}$$

using a trainable weight matrix $W^{\text{a}}$ and bias vector $\mathbf{b}^{\text{a}}$. We then incorporate electron–atom information by contracting across all electrons

$$\mathbf{H}_I^1 = \sum_{i=1}^{n_{\text{el}}} \mathbf{e}_{iI}^0 \odot \left(W^{\text{e-a}}\,\Gamma^{\text{e-a}}(x_{iI}, \mathbf{x}_{iI})\right) \tag{17}$$

$$\mathbf{e}_{iI}^0 = \sigma\left(\mathbf{h}_i^0 + \mathbf{H}_I^0 + W^{\text{edge}}\tilde{\mathbf{x}}_{iI} + \mathbf{b}^{\text{edge}}\right), \tag{18}$$

with $\tilde{\mathbf{x}}_{iI} = [x_{iI}, \mathbf{x}_{iI}, \mathbf{k}_s]$ and trainable matrices $W^{\text{e-a}}$, $\mathbf{W}^{\text{edge}}$ and bias $\mathbf{b}^{\text{edge}}$. Subsequently, the atom embeddings are updated with $L$ dense layers

$$\mathbf{H}_I^{l+1} = \sigma\left(W^l\mathbf{H}_I^l + \mathbf{b}^l\right) + \mathbf{H}_I^l, \tag{19}$$

to finally diffuse them to electron-atom embeddings $\mathbf{e}_{iI}$ of the form

$$\mathbf{e}_{iI} = \sigma\left(W^{\text{out}_1}\mathbf{e}_{iI}^0 + \mathbf{H}_I^L + W^{\text{out}_2}\mathbf{h}_i^0 + \mathbf{b}^{\text{out}}\right) \odot \left(W^{\text{out}_3}\,\Gamma^{\text{out}}(x_{iI}, \mathbf{x}_{iI})\right). \tag{20}$$

with trainable matrix $W^{\text{out}_1}$, $W^{\text{out}_2}$, $W^{\text{out}_3}$ and trainable bias vector $\mathbf{b}^{\text{out}}$. For the sake of simplicity, we omitted the spin dependence in this presentation of the different embedding stages. Compared with the original Moon embedding[32], we use separate filters $\Gamma$ for the intermediate layers and the output layer, include the twist as input feature and omit the final aggregation step from electron–ion embeddings $\mathbf{e}_{iI}$ to electron embeddings $\mathbf{e}_i$.

**Orbitals.** The orbital features $\tilde{\mathbf{c}}_{\mu J}$ are a concatenation of four different types of features. First, as proposed by Scherbela et al.[21], we rely on mean-field coefficients from a Hartree–Fock calculation. The mean-field orbitals $\phi_\mu$ are localized as described in 'Orbital localization' and expanded in periodic, atom-centered, basis functions $b_\eta$

$$\phi_\mu(\mathbf{r}_i) = \sum_{I=1}^{N_{\text{atoms}}} \sum_{\eta=1}^{n_b} c_{I\mu,\eta}\, b_\eta(\mathbf{r}_i - \mathbf{R}_I), \tag{21}$$

where $n_b$ represents the per-atom basis set size of the Hartree–Fock calculation. We use a periodic version of the cc-pVDZ basis set[33] and

find no strong dependence of our results on the basis set used (Supplementary Fig. 5). In addition, we include relative atom positions $\tilde{\mathbf{R}}_I$

$$\tilde{\mathbf{R}}_I = \mathbf{R}_I - \frac{\sum_{J=1}^{N_{\text{atoms}}} \mathbf{R}_J Z_J}{\sum_{K=1}^{N_{\text{atoms}}} Z_K} \tag{22}$$

and analogously relative orbital positions $\tilde{\mathbf{R}}_\mu^{\text{orb}}$

$$\tilde{\mathbf{R}}_\mu^{\text{orb}} = \mathbf{R}_\mu^{\text{orb}} - \frac{\sum_{J=1}^{N_{\text{atoms}}} \mathbf{R}_J Z_J}{\sum_{K=1}^{N_{\text{atoms}}} Z_K}, \tag{23}$$

where $\mathbf{R}_\mu^{\text{orb}}$ is the position of the localized orbital $\mu$ as outlined in 'Orbital localization'. This allows the network to differentiate between different atoms and orbitals within the supercell. As a final feature, we include the twist of the system

$$\tilde{\mathbf{k}}_I^{\text{s}} = [\mathbf{k}_{\text{s}}, \sin(\mathbf{R}_I \cdot \mathbf{k}_{\text{s}}), \cos(\mathbf{R}_I \cdot \mathbf{k}_{\text{s}})] \in \mathbb{R}^5. \tag{24}$$

The final orbital features $\tilde{\mathbf{c}}_{I\mu}$ are obtained as a concatenation

$$\tilde{\mathbf{c}}_{I\mu} = [\mathbf{c}_{I\mu}, \tilde{\mathbf{R}}_I, \tilde{\mathbf{R}}_\mu^{\text{orb}}, \tilde{\mathbf{k}}_I^{\text{s}}] \in \mathbb{R}^{d_{\text{orb}}}, \tag{25}$$

where $d_{\text{orb}} = n_b + 11$, resulting from the concatenation of the $n_b$ basis coefficient features, 3 atom position features, 3 orbital position features and 5 twist features.

## Sampling

We use the Metropolis Hastings algorithm[34] to draw samples $\mathbf{r}$ from our unnormalized density $|\Psi_{\boldsymbol{\theta}}|^2$. We use Gaussian all-electron proposals $\mathbf{r}^{\text{prop}}$ of the form

$$\mathbf{r}^{\text{prop}} = \mathbf{r} + s\boldsymbol{\delta}, \tag{26}$$

where $\boldsymbol{\delta}$ is drawn from a $3n_{\text{el}}$-dimensional standard normal distribution. We continuously adjust the stepsize $s$ to obtain a mean acceptance probability of approximately 50%. Empirically, we find no strong dependence of autocorrelations on this acceptance target, as long as it is roughly between 30% and 70%. While it can be shown that under simplifying assumptions 23% is the optimal acceptance rate[35], we do not find this to be optimal in practice. Performance is more strongly impacted by too small acceptance rates, and thus, we opt for the larger ~50%.

When calculating properties of the hydrogen chain for different lattice constants $R$, special care must be given to the treatment of spins. The hydrogen chain has two phases with different arrangements of spins. In the insulating phase at large lattice constant, the ground state is antiferromagnetic, that is, neighboring spins prefer to be aligned antiparallel. In the metallic phase at small lattice constant, this antiferromagnetic ordering decreases and the system may even show ferromagnetic domains[3]. Moving between these two configurations is difficult using local Monte Carlo updates as given by equation (26), so we modify our Metropolis Hastings proposal function. In addition to moving electrons in real space, we occasionally propose moves that swap the positions of two electrons with opposite spin. To avoid biasing our sampling toward either spin configuration, we initialize half our Monte Carlo walkers in the antiferromagnetic configuration (neighboring electrons having opposite spin) and half our Monte Carlo walkers in a ferromagnetic configuration (all spin-up electrons in one half of the chain and all spin-down electrons in the other half). We found that, on the contrary, initializing all walkers in the antiferromagnetic configuration (as might be indicated, for example, by a mean-field calculation) can cause the optimization to fall into local energy minima during wavefunction optimization.

When optimizing a transferable wavefunction across multiple systems, we must also sample these systems during training. To simplify implementation, we sample only a single system per gradient step. We choose this system randomly, with its probability being either proportional to the systems weight (in the case of twist averaging) or proportional to the variance per electron.

## Complex KFAC

We use the Kronecker factored approximate curvature (KFAC) method[36] to optimize the trainable parameters of our ansatz. KFAC uses the Fisher information matrix as a metric in the space of wavefunction parameters. For real wavefunctions, the Fisher matrix is equivalent to the preconditioner used in the stochastic reconfiguration method [6], but this is not the case for complex wavefunctions. Instead, the Fubini–Study metric should be used, given by

$$F_{ij} = \text{Re}\left\{\left\langle \frac{\partial \ln \psi}{\partial \theta_i}^* \frac{\partial \ln \psi}{\partial \theta_j} \right\rangle\right\} \tag{27}$$

Writing the complex wavefunction in polar form, $\psi = \rho e^{i\phi}$, this becomes

$$F = \left\langle \frac{\partial \ln \rho}{\partial \theta_i} \frac{\partial \ln \rho}{\partial \theta_j} + \frac{\partial \phi}{\partial \theta_i} \frac{\partial \phi}{\partial \theta_j} \right\rangle, \tag{28}$$

where the first term is the Fisher information matrix and the second term is the new contribution due to the phase of the wavefunction. The second term is zero if the phase is a global constant, such as for a purely real-valued wavefunction. For our wavefunction, the phase is generally nonzero, due to the complex-valued orbitals and the phase factor introduced to enforce twist-averaged boundary conditions.

## Orbital localization

To obtain orbital features that generalize well across system sizes, we do not use the canonical mean-field coefficients $\mathbf{c}$ as network inputs. Rather, we use the coefficients $\mathbf{c}^{\text{loc}}$ of maximally localized Wannier orbitals computed from $\mathbf{c}$. We follow the procedure of ref. 37 to find a unitary rotation $U$ within the subspace spanned by the occupied orbitals. Given a set of mean-field orbitals $\phi_\mu(\mathbf{r})$, $\mu = 1, \ldots, n_{\text{el}}$, expanded in periodic, atom-centered basis functions $b_{I\eta}(\mathbf{r})$, $I = 1, \ldots, N_{\text{atoms}}$, $\eta = 1, \ldots, n_{\text{b}}$, as described in 'Architecture', we compute the complex polarization matrix

$$\chi_{\alpha,\nu\mu} = \int \phi_\nu^*(\mathbf{r}) e^{i\mathbf{r}^T \mathbf{G}_\alpha} \phi_\mu(\mathbf{r})\, \mathrm{d}\,\mathbf{r}, \quad \chi \in \mathbb{C}^{3\times n_{\text{orb}}\times n_{\text{orb}}} \tag{29}$$

where $G = 2\pi L^{-T}$ is the matrix of reciprocal lattice vectors. Given a unitary transformation $\mathbf{U} \in \mathbb{C}^{n_{\text{orb}}\times n_{\text{orb}}}$, the transformed polarization matrix $\hat{\chi}$ and the corresponding localization loss $\mathcal{L}$ are given by

$$\boldsymbol{\Omega}_{\alpha\mu} = \hat{\boldsymbol{\chi}}_{\alpha,\mu\mu} = (U^\dagger \chi_\alpha U)_{\mu\mu} \tag{30}$$

$$\mathcal{L}(U) = -\,\|\,\boldsymbol{\Omega}(U)\|_2^2, \tag{31}$$

where $\|\cdot\|_2$ denotes the $L_2$ norm. To facilitate unconstrained optimization, we parameterize the unitary matrix $U$ as the complex matrix exponential of a symmetrized, unconstrained complex matrix $A$:

$$U = e^{\frac{i}{2}(\mathbf{A}+\mathbf{A}^\dagger)}. \tag{32}$$

We obtain the optimal $U^{\text{loc}}$, and corresponding orbital coefficients $\mathbf{c}^{\text{loc}}$ via gradient-based optimization

$$U^{\text{loc}} = \text{argmin}_U \mathcal{L}(U), \qquad c^{\text{loc}}_{I\eta,\mu} = \sum_m c_{I\eta,\nu} U^{\text{loc}}_{\nu\mu}, \tag{33}$$

using the Adam[38] optimizer. For orthorombic supercells, the position of the Wannier center $\mathbf{R}_\mu^{\mathrm{orb}}$ of the localized orbital $\mu$ can be inferred from the localized polarization matrix $\hat{\boldsymbol{\chi}}$ as

$$R_{l\alpha}^{\mathrm{orb}} = -\frac{L_{\alpha\alpha}}{2\pi}\,\mathrm{Im}\,\log\hat{\boldsymbol{\chi}}^{\alpha}_{\mu\mu}, \qquad \alpha = 1\ldots 3,\ \ \mu = 1\ldots n_{\mathrm{orb}}. \tag{34}$$

For other supercells, we follow the generalization given in ref. 37.

### Observables and postprocessing

**TABC.** In a finite system, there are finite-size errors related to both the artificial constraint of periodicity in the supercell and the lack of correlations of longer range than the supercell. The effects of the former on the single-particle contributions to the Hamiltonian, namely the kinetic energy, the Hartree energy and the electron–ion interaction, can be reduced using TABC[20,25]. Twisted boundary conditions require that the wavefunction obeys

$$\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_i+\mathbf{L}_\alpha,\ldots,\mathbf{r}_N) = e^{i\mathbf{k}\cdot\mathbf{L}_\alpha}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_i,\ldots,\mathbf{r}_N), \tag{35}$$

where $\mathbf{L}_\alpha$ is the $\alpha$th supercell lattice vector. Equation (35) is enforced by adding a position-dependent phase $e^{i\mathbf{k}_s\cdot\mathbf{r}_i}$ for each electron in the transferable atomic orbitals. as seen in equation (5). To obtain twist-averaged observables, we compute observables across a grid of twists $\mathbf{k}_s$ spanning the first Brillouin zone and average the results.

**Structure factor correction.** To handle finite-size errors in the Ewald energy, we use the finite-size corrections proposed by ref. 25. Writing the Ewald energy in terms of Fourier series, we get

$$\langle\hat{V}_{\mathrm{E}}\rangle = \frac{N}{2}\left\{\upsilon_{\mathrm{M}} + \frac{1}{\Omega}\sum_{\mathbf{G}_s\neq 0}\upsilon_{\mathrm{E}}(G_s)[S(\mathbf{G}_s)-1]\right\} + \frac{1}{2\Omega}\sum_{\mathbf{G}_p\neq 0}\upsilon_{\mathrm{E}}(G_p)\rho(\mathbf{G}_p)\rho^*(\mathbf{G}_p). \tag{36}$$

Here, $\upsilon_{\mathrm{M}}$ is the Madelung energy, $\Omega$ is the supercell volume, $\upsilon_{\mathrm{E}}(\mathbf{k}) = 4\pi/k^2$ is the Fourier transform of the Coulomb interaction, and $\mathbf{G}_s$ ($\mathbf{G}_p$) is a simulation (primitive) cell reciprocal lattice vector. The translationally averaged structure factor $S(\mathbf{G}_p)$ is defined by

$$S(\mathbf{G}_s) = \frac{1}{N}\left[\langle\hat{\rho}(\mathbf{G}_s)\hat{\rho}^*(\mathbf{G}_s)\rangle - \langle\hat{\rho}(\mathbf{G}_s)\rangle\langle\hat{\rho}^*(\mathbf{G}_s)\rangle\right], \tag{37}$$

where $\hat{\rho}(\mathbf{G}_s) = \sum_i \exp(-i\mathbf{G}_s\cdot\mathbf{r}_i)$ is the Fourier representation of the operator for the electron density. The structure factor converges fairly rapidly with supercell size, so we can assume that $S_\Omega(\mathbf{k}) \approx S_\infty(\mathbf{k})$. In this limit, the largest contribution to the error is the omission of the $G_s = 0$ term in the first sum. In cubic systems, we have $S(\mathbf{k}) \propto \eta k^2 + O(k^4)$, with odd terms missing due to inversion symmetry, and the $\mathbf{k}\to 0$ limit of $S(\mathbf{k})\upsilon_{\mathrm{E}}(k)$ is well defined. As such, to a first approximation, the Ewald finite-size error is given by

$$\Delta V_{\mathrm{E}} \approx \frac{N}{2\Omega}\lim_{k\to 0}\upsilon_{\mathrm{E}}(k)S(\mathbf{k}) = \frac{4\pi N}{2\Omega}\lim_{k\to 0}\frac{S(\mathbf{k})}{k^2}. \tag{38}$$

Sampling $S(\mathbf{G}_s)$ at supercell reciprocal lattice vectors $\mathbf{G}_s$, we approximate the limit $k\to 0$ by fitting the function

$$S(k) \approx f(k) = 1 - e^{-a_0k^2 - a_1k^4}, \tag{39}$$

with $a_0$ and $a_1$ greater than zero. The form of the fit ensures that $S(k)$ has the correct $k^2$ behavior at small $k$ and that $\lim_{k\to\infty} S(\mathbf{k}) = 1$. The finite-size correction $\Delta V_{\mathrm{E}}$ is given by $\Delta V_{\mathrm{E}} \approx 4\pi N a_0/2\Omega$.

**ZPVE.** To estimate the ZPVE contribution for graphene, we obtained the phonon density of states $D(\omega)$ calculated within DFT using the Perdew–Burke–Ernzerhof functional[2] from ref. 39. The ZPVE energy per primitive cell, $E_{\mathrm{ZPVE}}$, is then given as

$$E_{\mathrm{ZPVE}} = \frac{3N_{\mathrm{atoms}}^{\mathrm{prim}}}{\int D(\omega)\,\mathrm{d}\,\omega}\int D(\omega)\frac{1}{2}\hbar\omega\,\mathrm{d}\,\omega, \tag{40}$$

where $N_{\mathrm{atoms}}^{\mathrm{prim}} = 2$ is the number of atoms per primitive unit cell of graphene. This yields a ZPVE of 12.8 mHa per primitive cell for graphene. For LiH, we use ZPVE data published in ref. 26.

## Data availability

All data, including geometries, configurations and the figure source data, are available via GitHub at https://github.com/mdsunivie/deeperwin and via Zenodo at https://doi.org/10.5281/zenodo.16084892 (ref. 40). Source data are provided with this paper.

## Code availability

All code is available via GitHub at https://github.com/mdsunivie/deeperwin and via Zenodo at https://doi.org/10.5281/zenodo.16084892 (ref. 40).

## References


1. Sherrill, C. D. Frontiers in electronic structure theory. *J. Chem. Phys.* **132**, 110902 (2010).
2. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
3. Simons Collaboration on the Many-Electron Problem. Ground-state properties of the hydrogen chain: dimerization, insulator-to-metal transition, and magnetic phases. *Phys. Rev. X* **10**, 031058 (2020).
4. Burke, K. Perspective on density functional theory. *J. Chem. Phys.* **136**, 150901 (2012).
5. Foulkes, W. M. C., Mitas, L., Needs, R. J. & Rajagopal, G. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.* **73**, 33–83 (2001).
6. Pfau, D., Spencer, J. S., Matthews, Alexander, G. D. G. & Foulkes, W. M. C. Ab initio solution of the many-electron Schr"dinger equation with deep neural networks. *Phys. Rev. Res* **2**, 033429 (2020).
7. Carleo, G. & Matthias, T. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
8. Hermann, J., Schätzle, Z. & Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.* **12**, 891–897 (2020).
9. Gerard, L. et al. Gold-standard solutions to the schrödinger equation using deep learning: how much physics do we need?. *Adv. Neural Inf. Process. Syst.* **35**, 10282–10294 (2022).
10. von Glehn, I., Spencer, J. S. & Pfau, D. A self-attention ansatz for ab initio quantum chemistry. Preprint at https://doi.org/10.48550/arXiv.2211.13672 (2022).
11. Choo, K., Carleo, G., Regnault, N. & Neupert, T. Symmetries and many-body excitations with neural-network quantum states. *Phys. Rev. Lett.* **121**, 167204 (2018).
12. Sharir, O., Levine, Y., Wies, N., Carleo, G. & Shashua, A. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.* **124**, 020503 (2020).
13. Luo, D. & Clark, B. K. Backflow transformations via neural networks for quantum many-body wave functions. *Phys. Rev. Lett.* **122**, 226401 (2019).
14. Cassella, G. et al. Discovering quantum phase transitions with fermionic neural networks. *Phys. Rev. Lett.* **130**, 036401 (2023).

15. Pescia, G., Nys, J., Kim, J., Lovato, A. & Carleo, G. Message-passing neural quantum states for the homogeneous electron gas. *Phys. Rev. B* **110**(3), 035108 (2024).
16. Kim, J. et al. Neural-network quantum states for ultra-cold Fermi gases. *Commun. Phys.* **7**, 1–12 (2024).
17. Lou, WanTong et al. Neural wave functions for superfluids. *Phys. Rev. X* **14**, 021030 (2024).
18. Li, X., Li, Z. & Chen, J. Ab initio calculation of real solids via neural network ansatz. *Nat. Commun.* **13**, 7895 (2022).
19. Li, X., Qian, Y. & Chen, J. Electric polarization from a many-body neural network ansatz. *Phys. Rev. Lett.* **132**, 176401 (2024).
20. Lin, C., Zong, F.-H. & Ceperley, D. M. Twist-averaged boundary conditions in continuum quantum Monte Carlo. *Phys. Rev. E* **64**, 016702 (2001).
21. Scherbela, M., Gerard, L. & Grohs, P. Towards a transferable fermionic neural wavefunction for molecules. *Nat. Commun.* **15**, 120 (2024).
22. Simons Collaboration on the Many-Electron Problem. Towards the solution of the many-electron problem in real materials: equation of state of the hydrogen chain with state-of-the-art many-body methods. *Phys. Rev. X* **7**, 031059 (2017).
23. Monkhorst, H. J. & Pack, J. D. Special points for Brillouin-zone integrations. *Phys. Rev. B* **13**, 5188–5192 (1976).
24. Shi, H. & Zhang, S. Some recent developments in auxiliary-field quantum Monte Carlo for real materials. *J. Chem. Phys.* **154**, 024107 (2021).
25. Chiesa, S., Ceperley, D. M., Martin, R. M. & Holzmann, M. Finite-size error in many-body simulations with long-range interactions. *Phys. Rev. Lett.* **97**, 076404 (2006).
26. Nolan, S. J., Gillan, M. J., Alfé, D., Allan, N. L. & Manby, F. R. Calculation of properties of crystalline lithium hydride using correlated wave function theory. *Phys. Rev. B* **80**, 165109 (2009).
27. Li, R. et al. A computational framework for neural network-based variational Monte Carlo with forward Laplacian. *Nat. Mach. Intell.* **6**, 209–219 (2024).
28. Li, X., Fan, C., Ren, W. & Chen, J. Fermionic neural network with effective core potential. *Phys. Rev. Res.* **4**, 013021 (2022).
29. Azadi, S. & Foulkes, W. M. C. Efficient method for grand-canonical twist averaging in quantum Monte Carlo calculations. *Phys. Rev. B* **100**, 245142 (2019).
30. Scherbela, M., Gerard, L. & Grohs, P. Variational Monte Carlo on a budget—fine-tuning pre-trained neural wavefunctions. In *37th Conference Neural Information Processing Systems* (eds Oh, A. et al) 23902–23920 (Curran Associates, 2023).
31. Ewald, P. P. Die Berechnung Optischer und Elektrostatischer Gitterpotentiale. *Ann. Phys.* **369**, 253–287 (1921).
32. Gao, N. & Günnemann, S. Generalizing neural wave functions. In *Proc. 40th International Conference on Machine Learning* vol. 202, 10708–10726 (PMLR, 2023).
33. Dunning Jr, T. H. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J. Chem. Phys.* **90**, 1007–1023 (1989).
34. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092 (1953).
35. Sherlock, C. Optimal scaling of the random walk Metropolis: general criteria for the 0.234 acceptance rule. *J. Appl. Prob.* **50**, 1–15 (2013).
36. Martens, J. & Grosse, R. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proc. 32nd International Conference on Machine Learning* 2408–2417 (PMLR, 2015).
37. Silvestrelli, P. L. Maximally localized Wannier functions for simulations with supercells of general symmetry. *Phys. Rev. B* **59**, 9703–9706 (1999).
38. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2017).
39. Diery, W. A., Moujaes, E. A. & Nunes, R. W. Nature of localized phonon modes of tilt grain boundaries in graphene. *Carbon* **140**, 250–258 (2018).
40. Scherbela, M., Grohs, P. & Gerard, L. DeepErwin for solids. *Zenodo* https://doi.org/10.5281/zenodo.16084892 (2025).


## Acknowledgements

We acknowledge helpful discussions with J. Chen, X. Li, T. Lou and G. Arctaedius and thank X. Li for sharing data on DeepSolid. We gratefully acknowledge financial support from the following grants: Austrian Science Fund FWF Project I 3403 (P.G.), WWTF-ICT19-041 (L.G.) and Aker Scholarship (H.S.). The computational results have been partially achieved using the Vienna Scientific Cluster and Leonardo (Project L-AUT 005). H.S. gratefully acknowledges the Gauss Centre for Supercomputing e.V. (https://www.gauss-centre.eu) for providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC); the HPC RIVR consortium and EuroHPC JU for resources on the Vega high-performance computing system at IZUM, the Institute of Information Science in Maribor; and the UK Engineering and Physical Sciences Research Council for resources on the Baskerville Tier 2 HPC service. Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure program (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. The funders had no role in the study design, data collection and analysis, decision to publish or preparation of the manuscript.

## Author contributions

H.S. proposed the idea of shared twist averaging. M.S. and L.G. proposed concrete architecture and approach. L.G., M.S. and H.S. jointly worked on implementation: L.G. worked on the periodic Hamiltonian and contributed to the model. M.S. implemented/extended the model and orbital localization and contributed to the mean-field orbitals. H.S. extended KFAC to complex wavefunctions, implemented the evaluation of the mean-field orbitals and contributed to the periodic Hamiltonian. M.S. designed and ran experiments on the H chains. L.G. designed and ran experiments for LiH and graphene with support from H.S. and M.S. L.G., M.S., H.S. and W.M.C.F. jointly wrote the paper with supervision and funding from P.G. and W.M.C.F.

## Funding

Open access funding provided by University of Vienna.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s43588-025-00872-z.

**Correspondence and requests for materials** should be addressed to P. Grohs.

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