A structural optimization algorithm with stochastic forces and stresses

Siyuan Chen & Shiwei Zhang

We propose an algorithm for optimizations in which the gradients contain stochastic noise. This arises, for example, in structural optimizations when computations of forces and stresses rely on methods involving Monte Carlo sampling, such as quantum Monte Carlo or neural network states, or are performed on quantum devices that have intrinsic noise. Our proposed algorithm is based on the combination of two ingredients: an update rule derived from the steepest-descent method, and a staged scheduling of the targeted statistical error and step size, with position averaging. We compare it with commonly applied algorithms, including some of the latest machine learning optimization methods, and show that the algorithm consistently performs efficiently and robustly under realistic conditions. Applying this algorithm, we achieve full-degree optimizations in solids using ab initio many-body computations, by auxiliary-field quantum Monte Carlo with plane waves and pseudopotentials. A potential metastable structure in Si is discovered using density-functional calculations with synthetic noisy forces.

Geometry optimization is the procedure to locate the structure with energy or free-energy minimum in a solid or molecular system given the atomic compositions. Such a local or global minimum state is usually a naturally existing structure under common or extreme conditions. As an essential ingredient in materials discovery and design, structural search and geometry optimization have important applications from quantum materials to catalysis to protein folding to drug design, covering wide-ranging areas including condensed-matter physics, materials science, chemistry, biology and so on. The problems involved are fundamental, connecting applied mathematics, algorithms and computing with quantum chemistry and physics. With the rapid advent of computational methods and computing platforms, they have become a growing component of the scientific research repertoire, complementing and in some cases supplementing experimental efforts.

The vast majority of geometry optimization efforts to date have been performed with an effective ion–ion potential (force fields) or ab initio molecular dynamics based on density-functional theory (DFT). Force fields are obtained empirically from experimental data, or derived from DFT calculations for fixed structures, or learned from combinations of theoretical or experimental data. Geometry optimization using force fields is computationally low cost and convenient, and allows a variety of realistic calculations to be performed. The development of ab initio molecular dynamics signaled a fundamental step forward in accuracy and predictive power, where the interatomic forces are obtained more accurately from DFT on the fly, allowing the structural optimization to better capture the underlying quantum mechanical nature. With either force fields or ab initio DFT, the total energy and forces can be obtained deterministically without any statistical noise, and a well tested set of optimization procedures has been developed and applied.

In many quantum materials, however, Kohn–Sham DFT is still not sufficiently accurate, because of its underlying independent-electron framework, and a more advanced treatment of electronic correlations is needed to provide reliable structural predictions. Examples of such materials include so-called strongly correlated systems, which encompass a broad range of materials with great fundamental and technological importance. One of the frontiers in quantum science is to develop computational methods that can go beyond DFT-based methods in accuracy, with reasonable computational cost. Progress has been made on several fronts: for example, with the combination of DFT and GW.
approaches based on dynamical mean-field theory, quantum Monte Carlo (QMC) methods, quantum chemistry methods, and so on. For instance, the computation of forces and stresses with plane-wave auxiliary-field quantum Monte Carlo (PW-AFQMC) has recently been demonstrated (S.C. and S.Z., manuscript in preparation), paving the way for ab initio geometry optimization in this many-body framework.

One crucial aspect of geometry optimization with most of the post-DFT methods is that information on the potential-energy surface (PES) obtained from such approaches contains statistical uncertainties. The post-DFT methods, because of the exponential scaling of the Hilbert space in a many-body treatment, often involve stochastic sampling. This includes the various classes of QMC methods, but other approaches such as dynamical mean-field theory may also contain ingredients that rely on Monte Carlo sampling. Neural network wave-function approaches also typically involve stochastic ingredients. Additionally, if the many-body computation is performed on a quantum device, noise may also be present.

Geometry optimization under these situations, namely with noisy PES information, presents both challenges and opportunities. As we illustrate below, the presence of statistical noise can fundamentally change the behavior of the optimization, which can challenge existing algorithms. On the other hand, the fact that the size of the statistical error bar can be controlled by the amount of Monte Carlo sampling affords opportunities to tune and adapt the algorithm to minimize the integrated computational cost in the optimization process. To date, work on structural optimization with noisy PESs has not been widespread. One class of studies focuses on applications using variational Monte Carlo, which allows computations of forces and Hessians besides the total energy, mostly applying standard optimization algorithms. Another class focuses on using total energies to explore the PES, since computing forces and other gradients remains challenging in QMC, especially with projection methods beyond variational Monte Carlo. General and more systematic applications of structural optimization in correlated materials will in all likelihood require going beyond variational Monte Carlo in its current state, and effectively exploiting accurate forces and other gradients to efficiently scale up to high dimensions. This Article investigates optimization algorithms with this as the background. In principle, a number of algorithms widely used in the machine learning (ML) community can be adopted for the geometry optimization problem. However, we find that, in a variety of realistic situations under general conditions, the performance of these algorithms is often suboptimal. Given that the many-body computational methods tend to have higher computational costs, it is essential to minimize the number of times that force or stress needs to be evaluated, and the amount of sampling in each evaluation, before the optimized structure is reached.

In this Article, we propose an algorithm for optimization when the computed gradients have intrinsic statistical noise. The algorithm is found to consistently yield efficient and robust performance in geometry optimization using stochastic forces and stresses, often outperforming the best existing methods. We apply the method to realize a full geometry optimization in solids. In analyzing and testing the method, we also unexpectedly discovered a (meta-)stable orthorhombic Cmca structure in solid silicon.

Results

We have proposed a structural optimization algorithm to work with stochastic forces and gradients. The presence of statistical error bars in the gradients is a common characteristic in many quantum many-body computations. We find that existing optimization algorithms all experience substantial difficulties in such situations. This is a fundamental problem, whose importance is magnified by both the growing demand for the higher predictive power and the generally high cost of ab initio many-body calculations. Our algorithm addresses this problem by the combination of a fixed-step steepest descent (FSSD) and a staged error-targeting (SET) workflow with position averaging, with details described in Methods. Here we first compare FSSD with standard optimization methods, followed by a performance analysis of SET and the combination of FSSD × SET. The algorithm is then applied in an actual ab initio many-body computation, using PW-AFQMC to realize a full structural optimization. The optimization algorithm can be applied to atomic position and lattice structure optimizations, as well as a full geometry optimization combining the two. We demonstrated the combined approach for a full geometry optimization, which resulted in the discovery of a metastable structure in Si. Furthermore, we illustrate that the presence of statistical noise sometimes creates opportunities for discoveries in the optimization. This can be in the form of tuning the target statistical error to minimize the computational cost, or exploiting the noise to alter the optimization paths and expand the scope of the search, in the spirit of simulated annealing.

FSSD versus line-search and ML algorithms

In the presence of noise in the gradients, standard line-search algorithms such as steepest descent and conjugate gradient can suffer efficiency loss or even fail to find the correct local minimum. (Supplementary Section 1 provides an illustration.) Many ML methods, which avoid line search and incorporate advanced optimization algorithms for low-quality gradients, are an obvious choice as an alternative in such situations. Our expectation was that these would be the best option to serve as the engine in our optimization. However, to our surprise we found that the FSSD was consistently competitive with or even outperformed the ML algorithms in geometry optimizations in solids.

Below we describe two sets of tests in which we characterize the performance of FSSD in comparison with other methods. For line-search methods, we use the standard steepest descent, and the conjugate gradient with a Polak–Ribiére formula, which showed the best performance within several conjugate-gradient variants in our experiments. For the ML algorithms we choose three: RMSProp, ADEDELTA and Adam, which are well known and generally found to be among the best-performing methods for a variety of problems. For each algorithm, we have experimented with the choice of step size or learning rate to choose an optimal setting for the comparison. (Details of the parameter choices can be found in Methods.)

Figure 1 shows a convergence analysis of FSSD and other algorithms in solid Si in which the targeted minimum is the so-called β-tin structure, reached under a pressure-induced phase transition from the diamond structure, as illustrated in Fig. 1a; see Supplementary Section 2 for further details). Three random runs are shown for each method. As seen in Fig. 1b, the performance of line-search methods, in which one line-search iteration can take several steps, is lowered by the statistical noise. The convergence of FSSD is not only much faster but also more robust than the two line-search methods. The ML algorithms are shown in Fig. 1c. RMSProp shows slightly worse convergence speed and quality than FSSD. These methods have conceptual similarities: both involve averaging over gradient history, and both become a fixed-step approach when this averaging is turned off. ADEDELTA has excellent convergence quality, but slower convergence. Adam performs worse than the other algorithms here.

We next compared FSSD and the three ML algorithms in a two-dimensional solid, the MoS2 monolayer, which has an interesting energy landscape: the global minimum (2H) and a nearby local minimum (1T) are separated by a ridge, as depicted in Fig. 2 (system details in Supplementary Section 2). We observe that the original ML algorithms all lead to the local-minimum structure, while FSSD finds the global minimum. We then modified the ML algorithms and introduced a ‘by-norm’ variant (Methods). As shown in Fig. 2, this resulted in different behaviors from the original ‘element-wise’ algorithms, crossing over the ridge and finding the global minimum instead. These by-norm algorithms, similarly to FSSD, follow paths that are almost perpendicular.
by a filled symbol. The automatic script also identifies, after the fact, five optimization runs. In each stage, the end of each run is indicated scheduling is applied in SET. The convergence process is shown for integration and demonstrate the efficiency gain by their synergy, using inclusion of the first momentum Adam produces a path that is more 'flatter' landscape, which slows down its final convergence; due to its steep surfaces but reduces the step size more markedly when entering a damping dynamics, delaying its convergence speed.

These tests also confirm the characteristics of the ML algorithms seen in the Si phase transition problem. It is worth emphasizing that the observation here should not be taken as a general conclusion over any energy landscape. The proximity of the initial structure to the convergence boundary is a key factor, but the markedly different behaviors from the different variants are still interesting to note.

The convergence speed of each method in MoS$_2$ can be seen on the contour plot, where each arrow represents a single optimization step; a more direct comparison is shown in Fig. 2c. FSSD remains the fastest method, again closely followed by RMSProp and ADADELTA. These tests also confirm the characteristics of the ML algorithms seen in the Si test: RMSProp is similar to FSSD, and shows relatively fast convergence on the shortest route; ADADELTA optimizes efficiently on steep surfaces but reduces the step size more markedly when entering a ‘flatter’ landscape, which slows down its final convergence; due to its inclusion of the first momentum Adam produces a path that is more like a damping dynamics, delaying its convergence speed.

Performance and analysis of SET

When the FSSD is applied under the SET approach, a qualitative leap in capability and efficiency is achieved. In Fig. 3, we illustrate their integration and demonstrate the efficiency gain by their synergy, using the example of optimization in MoS$_2$. In Fig. 3a, a simple two-stage scheduling is applied in SET. The convergence process is shown for five optimization runs. In each stage, the end of each run is indicated by a filled symbol. The automatic script also identifies, after the fact, an initial position of convergence (step (4) of SET in Methods); the average of this position in each stage is indicated by the open diamond. A clear lag is seen between the two, leaving a considerable number of steps for position averaging in each run. Position averaging ensures that these steps are not wasted but effectively utilized. This is reflected by the initial positions in stage II being markedly better than the corresponding end positions in stage I, as seen in the lowering of the error in the energy. One of the runs is discontinued after stage I, because it is trapped in a local minimum, as identified by the clustering of the converged positions from all the runs. In stage II, the step size and error target are both reduced by a factor of 10.

Figure 3b shows the convergence plot without SET. The step size and error target are fixed at the values used in stage II above, so that the same convergence quality is achieved as in Fig. 3a. We see that all five runs converge in this setting. Figure 3c compares the computational costs with and without SET, and we see that the two-stage SET procedure resulted in a 90% saving, or tenfold gain in efficiency in the optimization.

There are two key ingredients in the SET approach: position averaging at the end of each stage, and discrete, staged scheduling instead of adapting the error bar and step size continuously with time. In FSSD, a larger step size will generally lead to faster convergence; however, it will result in worse final convergence quality, because the atomic positions will fluctuate with larger magnitudes around the minimum. Position (or parameter) averaging helps to markedly improve the convergence quality of FSSD. The idea of averaging parameters over an optimization trajectory has a long history$^{43–45}$ and has been applied in previous structural optimizations in QMC (for example, by Wagner and Grossman$^{24}$, Barborini et al.$^{23}$ and Guareschi and Filippi$^{27}$). Our algorithm defines a precise and efficient scheme to apply position averaging retroactively after convergence has been detected. It allows a wide range of choices for step size, with almost no effect on the convergence quality (illustrated in Supplementary Section 3). The convergence quality within this range is dictated by the target error-bar sizes $s$. This makes it more natural to introduce the concept of a separate stage, in which we target a smaller error bar (with increased computational cost), and reduce the step size at the same time to account for the reduced system scale. Comparing with a smooth scheduling procedure, we find this staged scheduling to be efficient, more robust, and resilient to saddle points.

We mention some possible improvements to the SET algorithm over our present implementation. We have chosen to reduce $s$ and the step size $L$ by the same scale when entering a new stage. Around the minimum, the optimal $L$ is essentially proportional to the distance $D$ to the minimum, suggesting a choice of $0.1D$–$0.2D$ for $L$. The target error bar $s$ on the force should also be reduced with $D$, but, as illustrated in Supplementary Section 3, $D$ decreases more slowly than $s$. This indicates that it would be more optimal to reduce $s$ faster than $L$. A related point is how much to reduce $s$ in each stage of the scheduling. If the choice is too aggressive, a large reduction in $L$ would be required to reach convergence, which in turn would require a large number of steps, and hence high computational cost. If a very small reduction of $s$ is used, a large number of stages will be needed, which is less optimal since there is a threshold of steps to identify convergence in each stage. Our empirical choice of $\sim 10$ is based on the balance of these two extremes.

It is worth emphasizing that SET can be employed in combination with other algorithms. For example, we find that position averaging can improve the convergence quality in (by-norm) RMSProp by a similar extent to what is seen with FSSD. The RMSProp × SET approach, although slightly slower than FSSD × SET in the examples we studied, would provide more freedom in the choice of the step size, as RMSProp allows for small auto-adaptations.

Finally, we comment on the computational cost and scaling of the overall FSSD × SET algorithm. Under optimal step size and error-bar sequence choices, the number of steps taken within each stage is roughly the same. The last stage dominates the computational cost associated with the force or gradient computation (Fig. 3c), and
We have performed a number of additional studies in different systems for comparison and to further characterize the behavior of the algorithm. These studies are described in Supplementary Information. In the comparisons, we have included more algorithms (for example SGD + momentum (Ref. 46,47), which, to avoid cluttering, are not shown here.

A realistic application in AFQMC
We next apply our algorithm to perform a fully ab initio quantum many-body geometry optimization in Si. Recent progress has made possible the direct computation of atomic forces and stresses by PW-AFQMC (S.C. and S.Z., manuscript in preparation). Employing this framework, we study the pressure-induced structure phase transition from the insulating diamond phase to the semimetallic β-tin phase.

Figure 4 shows the energy difference and Euclidean distance relative to the target β-tin structure at each step during the geometry optimization process. The run is divided into two stages. In stage I, our convergence analysis identified convergence at step 26. Atom positions are accumulated and averaged starting from this step, yielding a lower and more stable Euclidean distance curve. This averaged position is taken to be the starting point for the second stage. In the second stage the statistical error and the step size are reduced to two-sevenths of those of the first stage. The optimization quickly converges and approaches the correct β-tin structure. The total energy in the final structure is consistent with the ground-state energy computed by AFQMC at the ideal β-tin structure, and the final structure is in agreement with the ideal structure within our targeted precision (Euclidean distance of -0.1 bohr).

Discovery of a Si (meta-)stable structure
A (meta-)stable orthorhombic structure in Si was discovered unexpectedly in our study. In this section we present this structure. It emerged in tests of our algorithm for full-degree-of-freedom optimization in solids allowing both the atomic positions and the lattice structure to relax. Our approach for a full-degree-of-freedom optimization can be found in Methods.
Stage I Stage II

Energy difference (eV)

Not averaged Position averaged

Distance to glob. min. (nm)

Fig. 4 | A direct PW-AFQMC geometry optimization with the FSSD × SET algorithm. a. The total energy relative to the target β-tin structure versus the optimization step. For each of the steps, PW-AFQMC total energy is presented as mean value ± statistical error from 27 uncorrelated measurements of PW-AFQMC. The two insets show the starting and target structures. b. The Euclidean distance to the expected β-tin structure. Position averaging starts when convergence is identified in either stage.

Discussion

In addition to geometry optimization, the algorithm can potentially be applied to other problems in which the gradients contain stochastic noise. The two components of the algorithm can be applied independently or combined with other methods. Insights from them can also stimulate further developments. With the intense effort in many-body methods, we highlight the importance of first-order gradients in allowing predictive calculations for structural properties. The computation of such gradients should be an important direction of future research. Our algorithm emphasizes maximizing the capac-

ity while minimizing the computing budget for forces and gradients.

Methods

A noisy gradient, such as an interatomic force evaluated from a QMC calculation, can be written as

\[ \mathbf{F} = \mathbf{F} + \mathbf{\epsilon}, \]

where \( \mathbf{F} \) is the true force, and \( \mathbf{\epsilon} \) is the (expectation) value computed by the numerical method with stochastic components. The vector \( \mathbf{\epsilon} \) denotes stochastic noise; for example, the statistical error bar estimated from the QMC computation. In the case of a sufficiently large number of Monte Carlo samples (realized in most cases but not always), the central limit theorem dictates that the noise is given by a Gaussian

\[ \mathbf{\epsilon}_i = \mathcal{N}(0, \sigma_i^2), \]

where \( i \) denotes a component of the gradient (for example a combination of the atom number and the Cartesian direction in the case of interatomic forces), and \( \sigma_i \) is the s.d., which can be reduced as the square root of the number of effective samples \( N_e \). The computational cost is typically proportional to \( N_e \).

To facilitate our study, we create DFT models to simulate actual many-body computations with noise. We consider a number of real solids and realistic geometry optimizations, but use forces and stresses computed from DFT, which is substantially less computationally costly than many-body methods. Synthetic noise is introduced on the forces, defining \( \mathbf{\epsilon} \) according to the targeted statistical errors of the many-body computation, and sampling \( \mathbf{F} = \{ \mathbf{F}_i \} \) from \( \mathcal{N}((\mathbf{F}_i, \sigma_i^2)) \) where \( \{ \mathbf{F}_i \} \) are the corresponding forces or stresses computed from DFT. As indicated above, we have chosen the noise to be isotropic in all directions on the basis of our observations from AFQMC, but this can be generalized as needed. The DFT model replaces the many-body computation, and is called to produce \( \{ \mathbf{F}_i \} \) as the input to the optimization algorithm. This provides a controlled, flexible and convenient emulator for systematic studies of the performance of the optimization algorithm.

Our algorithm consists of two key components. In each step of the optimization, we follow an update rule using the current \( \mathbf{F} \), which is a fixed-step size modification of the gradient descent with momentum method, which we will refer to as ‘fixed-step steepest descent’ (FSSD). (Ideas similar to ‘fixed step’, or ‘fixing the norm of displacement’, are discussed in other papers in a different framework or context.) Globally, the optimization process is divided into stages, each with a target statistical error \( \sigma \) for \( \mathbf{\epsilon} \) (hence controlling the computational cost per gradient evaluation) and specific choice of step size, called a staged error-targeting (SET) workflow. The SET is complemented by a self-averaging procedure within each stage, which further accelerates convergence. We outline the two ingredients separately below, and provide analysis and discussion in the following sections.

FSSD update rule

The SET approach discussed later defines the overall algorithm. Each step in each stage of SET is taken with the FSSD algorithm, which works as follows. Let \( n \) denote the current step number, and \( \mathbf{x}_n \) denote the atom positions at the end of this step. Here, \( \mathbf{x} \) is an \( N_s \)-dimensional vector, with \( N_s \) being the number of degrees of freedom in the optimization.

(1) Calculate the force at the atomic configuration from the previous step: \( \mathbf{F}_n = -\nabla E(\mathbf{x}_{n-1}) \). In the case of quantum many-body computations, the loss function \( E \) is \( F \), and the force is typically computed as the estimator of an observable directly, for example via the Hellmann–Feynman theorem (S.C. and S.Z., manuscript in preparation).

(2) The search direction is then chosen as

\[ \mathbf{d}_n = \frac{\mathbf{a} \mathbf{d}_{n-1} + \mathbf{F}_{n-1}}{\sqrt{\mathbf{a} + 1}}, \]

where \( \mathbf{d}_{n-1} \) is the displacement direction of the step \( n - 1 \), which encodes the forces from past steps and thus serves as a ‘historic force’. We experiment with the choice of the parameter \( a \) (Supplementary Fig. 8), but typically set it to \( a = 1/e \).
(3) The displacement vector is now set to the chosen direction from (2), with \(L\) fixed throughout the stage:

\[
\Delta x_n = \epsilon \left( \frac{d_n}{|d_n|} \right).
\]

(4) Obtain the new \(x_n = x_{n-1} + \Delta x_n\). Account for symmetries and constraints such as periodic boundary conditions or restricting degrees of freedom as needed.

SET scheduling approach

The SET workflow can be described as follows.

(1a) Initialize the stage. At the beginning of each stage of SET, the step count \(n\) is set to 1, and an initial position \(x_n\) is given, which is either the input at the beginning of the optimization or inherited from the previous stage (see (5) below). We also set \(d_n = 0\) in (2) of FSSD (thus the first step within each stage is a standard steepest descent).

(1b) Use a fixed \(L\), and target a fixed average statistical error bar \(s\) for the force computation throughout this stage. The values of \(L\) and \(s\) are either input (first stage at the beginning of the optimization) or set at the end of the previous stage (see (5) below). From \(s\) we obtain an estimate of the computational resources needed, \(C(s) = s^{-2}\) for each force evaluation, which helps to set the run parameters during this stage (for example population size and projection time in AFQMC). We have used the average \(\sum_{\beta} F_{\beta}^2 / N_E\) for \(s\), but clearly other choices are possible. To initialize the optimization we have typically used \(L = 0.1 \sqrt{N_E}\) [bohr]. For \(s\), we have typically used an initial value of -20% of the average of each component of the initial force. These choices are ad hoc and can be replaced by other input values: for example, from an estimate by a less computationally costly approach such as DFT.

(2) Carry out a step of FSSD with the current \(L\) and the rationed \(C(s)\). This consists of the steps described in FSSD.

(3) Perform convergence analysis if a threshold number of steps has been reached. Our detailed convergence analysis algorithm is discussed below.

(4) If convergence is not reached in (3), loop back to (2) for the next step within this stage; otherwise, the analysis will reveal a previous step count \(m (m < n)\) where convergence was reached. Take the position average (see below) of \([x_m, x_{m+1}, \ldots, x_n]\) to obtain the final position of this stage, \(x\).

(5) If overall objective of optimization is reached, stop; otherwise, set \(x_0 = x\), modify \(L\) and \(s\), and return to (1). For the latter, we typically lower \(s\) and \(L\) by the same ratio.

Parameter choices in the optimization methods

The optimization methods employed in this work all have some free parameters or variations. We have not attempted to perform the most detailed optimization of these parameters. The following describes our choices.

FSSD uses a step size of 0.5 bohr for Fig. 1, 0.3 bohr for Fig. 2, 0.5 bohr for Fig. 3 (stage I) and 0.7 bohr for Fig. 4 (stage II). A mixing parameter of \(\alpha = 1/e\) is used throughout the work. Our tests show that a value between 0.35 and 0.5 yields good convergence speed and final convergence accuracy (Supplementary Fig. 8).

Steepest descent and conjugate gradient are based on a Newtonian line search that finds the root of \(\langle |F| \rangle_{\beta} = 90^\circ\), as in Supplementary Section 1. Conjugate gradient uses the Polak–Ribière formula and restarts every five steps. Note that for a noisy PES specially designed methods\(^{29}\) can result in better performance for conjugate gradient. This was not pursued here, since the required automatic differentiation is not always available in the many-body computations with which the optimization algorithm is expected to couple.

ML-based algorithms discussed in this work have an element-wise version and a by-norm version. The version in the original literature of RMSProp, ADADELTA and Adam is element-wise;\(^{40–42}\) for example, the RMSProp algorithm is

\[
\langle F_i^2 \rangle_{\beta} = \beta \langle F_i^2 \rangle_{\beta-1} + (1 - \beta) F_{\beta m}^2,
\]

\[
x_{\beta i}\beta+1 = x_{\beta i} + \frac{\eta}{\sqrt{\langle F_i^2 \rangle_{\beta}}} F_{\beta m}.
\]

where \(F_m\) is the force computed at \(x_n\) on the \(i\)-th degree of freedom, \(\beta\) is the mixing factor, \(\eta\) is the initial step size and \(\epsilon\) is a small number to prevent singularity. \(\langle |F| \rangle_{\beta}\) is a 'historical average' of all squared forces on the \(i\)-th degree of freedom. This original element-wise algorithm treats each dimension separately. A variant of this algorithm, which we call the by-norm algorithm, is given by replacing the equations above with

\[
\langle |F| \rangle_{\beta} = \beta \langle |F| \rangle_{\beta-1} + (1 - \beta) F_{\beta m}^2.
\]
\[ x_{n+1} = x_n + \frac{\eta}{\sqrt{\|F(x_n)\|}} F_n, \]  
(8)

where the force is now treated as a whole for all dimensions, as each dimension receives the same value as the prefactor for the force. By analogy, the FSSD algorithm we use should be classified as a by-norm algorithm.

Our application of the ADADELTA algorithm has one small modification from its original form\(^5\), which forces \(E(\Delta x)^2 \approx 0\) and appears to have poor efficiency in our optimizations. Replacing this value with a finite number gives the algorithm an initial boost and specifies the initial step size with \(\eta = \sqrt{E(\Delta x)^2}/(1 - \rho)\).

We used the following parameters for ML-based algorithms. RMSProp (element-wise) used the suggested mixing factor \(\beta = 0.9\), with an initial step size \(\eta = 0.2\) bohr in each dimension for Si (Fig. 1) and \(\eta = 0.1\) bohr in each dimension for MoS\(_2\) (Fig. 2). By-norm RMSProp used a stepsize of 0.7 bohr for MoS\(_2\) (Fig. 2). ADADELTA used a mixing parameter of \(\rho = 0.9\) with \(\eta = 0.4\) bohr in Si. ADADELTA (element-wise or by-norm) used the same \(\eta\) as RMSProp in MoS\(_2\). Adam (element-wise) used the same initial step size as RMSProp for all systems, and \(\beta_1 = 0.9, \beta_2 = 0.999\) as suggested in the original paper. By-norm Adam had a slower performance, hence we switched the initial step size from 0.7 bohr to 2.0 bohr in MoS\(_2\).

### Distance metric and position averaging

The Euclidean distance metric between two atom-position arrays \(x\) and \(x_{\text{ref}}\) is

\[ D(x, x_{\text{ref}}) = |x - x_{\text{ref}}| = \sqrt{\sum_{i=1}^{N_{\text{At}}} (x_i - x_{\text{ref},i})^2}, \]  
(9)

where \(N_{\text{At}}\) is the number of atoms. Periodicity, global translations on all \(N_{\text{At}}\) atoms, and atom permutations are applied to \(x\) to minimize \(D(x, x_{\text{ref}})\). If \(x_{\text{ref}}\) is a known structure with some crystal symmetry, then the crystal symmetry operations are also applied. This is the case when we measure the ‘distance to global minimum’ to compare algorithm efficiencies. On the other hand, in convergence analysis where we have no knowledge of the final structure, only the periodicity, translations and permutations are used.

Position averaging is performed with similar consideration for symmetries: the position of the last optimization step is chosen as \(x_{\text{ref}}\), and symmetry operations including periodicity, translation and permutation are applied to all earlier steps to minimize the distance to \(x_{\text{ref}}\) before the actual averaging takes place.

### Convergence analysis algorithm

A sketch of our convergence analysis algorithm can be found in Supplementary Fig. 9. The Euclidean distance metric is used to build a one-dimensional distance function of the steps in the optimization history, with \(x_n\) being the ‘current best guess’. At step \(N\), \(x_{\text{ref}}\) is selected as the position average of the last \(N_{\text{ave}}\) steps of the convergence procedure.

We compute \(D_n = D(x_n, x)\) for all \(0 \leq n \leq N - N_{\text{ave}}\), and then search for a step number \(m\) between \(N_{\text{ave}}\) and \(N - N_{\text{ave}} - N_0\) that divides the entire run into two phases (\(N_{\text{ave}}, N_0\) are the ‘minimum phase lengths’) of phases A and B, such that the ratio of the standard errors of the distances in the first phase and in the second phase is maximized:

\[ m = \arg \max_{n \in \{N_{\text{ave}}, N_0\}} \frac{R_n}{R_{n-m}} \equiv \frac{\text{s.e.m.}_{n=N_{\text{ave}}} \left(D_n\right)}{\text{s.e.m.}_{n=m} \left(D_n\right)} \]  
(10)

where \(\text{s.e.m.}_{n=m} \left(D_n\right)\) denotes the standard error of \(\{D_n, D_{n+1}, \ldots, D_m\}\). Convergence is reached if \(R_m > R_0\), where \(R_0\) is a threshold.

There are a few tunable parameters in this analysis algorithm. By default we choose \(N_{\text{ave}} = 5\), \(N_0 = 10\), and \(R_0 = 5\). These parameters can be varied. Note that low \(N_{\text{ave}}, N_0, N_0\), \(R_0\) might lead to misidentification of the saddle points as equilibrium, while high parameters can result in longer runs.

### Full-degree-of-freedom optimization

To apply our algorithm to a full-degree-of-freedom (atom geometry, lattice shape and volume) optimization, we combine the atomic position vectors and strain tensor into a single generalized position

\[ x = \left( x; \epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12}, \epsilon_{13}, \epsilon_{23} \right), \]  
(12)

and the interatomic forces and stress tensor into a single gradient

\[ f = \left( F; \sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23} \right), \]  
(13)

such that \( f = -\partial E(x)/\partial x \) as before. The cell volume \(\Omega\) appears above, and is included in the definition of the stress tensor: \(\sigma_{ij} = -\Omega (\partial E(\Sigma)/\partial e_{ij})\) (ref. 2). Care must be taken with metrics; for example, \(L\) in the algorithm should be defined as

\[ L = \sqrt{\Delta x^2 + \sum_{i=1}^{3} (v^{-1} \Delta e_{i})^2}, \]  
(14)

where \(v\) has the dimension of inverse length.

An additional role of \(v\) is to tune the optimization procedure, as it controls the relative step size for optimizing the atomic positions versus the overall lattice structure. Different choices thus can result in different optimization trajectories. As we describe in detail in Supplementary Section 4, there is considerable sensitivity of the optimized structure (local minimum) to the choice of \(v\), as well as an interplay with the particular stochastic realization of the optimization trajectory. In general this would seem to be an additional disadvantage of optimization in the presence of stochastic gradients. However, it provides a natural realization of statistical sampling of the landscape, which could broaden the search in the optimization. It is this feature that led to the surprise discovery of the structure shown in Fig. 5.

### Data availability

All data in this Article were generated with Quantum Espresso 5.0.1 and/or plane-wave AFQMC v26-2020.04.02 developed in our group (unpublished). Source data for Figs. 1–4 are available at https://github.com/schen24wm/fssd-set and Zenodo\(^6\) and also available with this Article. Detailed parameters of the \(\text{Cnca}\) structure (Fig. 5) we found are available in Supplementary Section 5.

### Code availability

Code of FSSD × SET is available at https://github.com/schen24wm/fssd-set and Zenodo\(^6\), and available as Supplementary Software 1 with this Article.

### References

1. Frenkel, D. & Smit, B. Understanding Molecular Simulation: from Algorithms to Applications (Academic, 2002).

2. Leach, A. Molecular Modelling: Principles and Applications 2nd edn (Prentice Hall, 2001).

3. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. Phys. Rev. 136, B864 (1964).

4. Jones, R. O. Density functional theory: its origins, rise to prominence, and future. Rev. Mod. Phys. 87, 897 (2015).

5. Becke, A. D. Perspective: Fifty years of density-functional theory in chemical physics. J. Chem. Phys. 140, 18A301 (2014).

6. Burke, K. Perspective on density functional theory. J. Chem. Phys. 136, 150901 (2012).
7. Car, R., & Parrinello, M. Unified approach for molecular dynamics and density-functional theory. Phys. Rev. Lett. 55, 2471 (1985).
8. Hedin, L. New method for calculating the one-particle Green’s function with application to the electron-gas problem. Phys. Rev. 139, A796 (1965).
9. Georges, A., Kotliar, G., Krauth, W. & Rozenberg, M. J. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. Rev. Mod. Phys. 68, 13 (1996).
10. Foulkes, W. M. C., Mitas, L., Needs, R. J. & Rajagopal, G. Quantum Monte Carlo simulations of solids. Rev. Mod. Phys. 73, 33 (2001).
11. Zhang, S. & Krakauer, H. Quantum Monte Carlo method using phase-free random walks with Slater determinants. Phys. Rev. Lett. 90, 136401 (2003).
12. Rillo, G., Morales, M. A., Ceperley, D. M. & Pierleoni, C. Coupled electron–ion Monte Carlo simulation of hydrogen molecular crystals. J. Chem. Phys. 148, 102314 (2018).
13. Tirelli, A., Tenti, G., Nakano, K. & Sorella, S. High-pressure hydrogen by machine learning and quantum Monte Carlo. Phys. Rev. B 106, L041105 (2022).
14. Levine, I. N. Quantum Chemistry (Prentice Hall, 1991).
15. Cramer, C. J. Essentials of Computational Chemistry (Wiley, 2002).
16. Suewattana, M., Purwanto, W., Zhang, S., Krakauer, H. & Walter, E. J. Phaseless auxiliary-field quantum Monte Carlo calculations with plane waves and pseudopotentials: applications to atoms and molecules. Phys. Rev. B 75, 245123 (2007).
17. Jia, Z.-A. et al. Quantum neural network states: a brief review of methods and applications. Adv. Quantum Technol. 2, 1800077 (2019).
18. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. Science 355, 602 (2017).
19. Lanyon, B. P. et al. Towards quantum chemistry on a quantum computer. Nat. Chem. 2, 106–111 (2010).
20. Huggins, W. J. et al. Unbiasing fermionic quantum Monte Carlo with a quantum computer. Nature 603, 416–420 (2022).
21. Guareschi, R. & Filippi, C. Ground- and excited-state geometry optimization of small organic molecules with quantum Monte Carlo. J. Chem. Theory Comput. 9, 5513–5525 (2013).
22. Zen, A., Zhelyazov, D. & Guidoni, L. Optimized structure and vibrational properties by error affected potential energy surfaces. J. Chem. Theory Comput. 8, 4204–4215 (2012).
23. Barborini, M., Sorella, S. & Guidoni, L. Structural optimization by quantum Monte Carlo: investigating the low-lying excited states of ethylene. J. Chem. Theory Comput. 8, 1260–1269 (2012).
24. Wagner, L. K. & Grossman, J. C. Quantum Monte Carlo calculations for minimum energy structures. Phys. Rev. Lett. 104, 210201 (2010).
25. Tihonen, J., Kent, P. R. C. & Krogel, J. T. Surrogate Hessian accelerated structural optimization for stochastic electronic structure theories. J. Chem. Phys. 156, 054104 (2022).
26. Chiesa, S., Ceperley, D. M. & Zhang, S. Accurate, efficient, and simple forces computed with quantum Monte Carlo methods. Phys. Rev. Lett. 94, 036404 (2005).
27. Assaraf, R. & Caffarel, M. Zero-variance zero-bias principle for observables in quantum Monte Carlo: application to forces. J. Chem. Phys. 119, 10536 (2003).
28. Robbins, H. & Monro, S. A stochastic approximation method. Ann. Math. Stat. 22, 400–407 (1951).
29. Armijo, L. Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. 16, 1–3 (1966).
30. Wolfe, P. Convergence conditions for ascent methods. SIAM Rev. 11, 226–235 (1969).
31. Wolfe, P. Convergence conditions for ascent methods. II: Some corrections. SIAM Rev. 13, 185–188 (1971).
32. Bertsekas, D. P. & Tsitsiklis, J. N. Gradient convergence in gradient methods with errors. SIAM J. Optim. 10, 627–642 (2000).
33. Bertsekas, D. P. Nonlinear Programming (Athena, 2016).
34. Debye, P. Näherungsformeln für die zylinderfunktionen für große werte des arguments und unbeschränkt veränderliche werte des index. Math. Ann. 67, 535–558 (1909).
35. Hestenes, M. R. & Stiefel, E. Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. 49, 409–436 (1952).
36. Shewchuk, J. R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf (Carnegie Mellon Univ., 1994).
37. Fletcher, R. & Reeves, C. M. Function minimization by conjugate gradients. Comput. J. 7, 149–154 (1964).
38. Polak, E. & Ribière, G. Note sur la convergence de méthodes de directions conjuguées. ESAIM Math. Model. Numer. Anal. 3, 35–43 (1969).
39. Schraudolph N. N. & Graepel, T. Combining conjugate direction methods with stochastic approximation of gradients. Proc. Mach. Learning Res. 44, 248–253 (2003).
40. Tieleman T. & Hinton, G. Lecture 6a: Neural Networks for Machine Learning (Computer Science, Univ. Toronto, 2012). https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
41. Zeiler, M. D. ADADELTA: an adaptive learning rate method. Preprint at https://doi.org/10.48550/arxiv.1212.5701 (2012).
42. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
43. Poljak, B. T. & Juditsky, A. B. Acceleration of stochastic approximation by averaging. SIAM J. Control Optim. 30, 838–855 (1992).
44. Poljak, B. New method of stochastic approximation type. Autom. Remote Control 51, 937–1008 (1990).
45. Ruppert, D. Efficient Estimations from a Slowly Convergent Robbins–Monro Process Technical Report (Cornell Univ. Operations Research and Industrial Engineering, 1988).
46. Qian, N. On the momentum term in gradient descent learning algorithms. Neural Netw. 12, 145–151 (1999).
47. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. Nature 323, 533–536 (1986).
48. Chen, S. schen24wm/geoopt-srscdata: FSSDxSET source data v0.1. Zenodo https://doi.org/10.5281/zenodo.7157782 (2022).
49. Chen, S. schen24wm/fssd-set: FSSDxSET v0.1. Zenodo https://doi.org/10.5281/zenodo.7157673 (2022).
50. Chen, M., Yu, T.-Q. & Tuckerman, M. E. Locating landmarks on high-dimensional free energy surfaces. Proc. Natl Acad. Sci. USA 112, 3235–3240 (2015).
51. Laio, A. & Parrinello, M. Escaping free-energy minima. Proc. Natl Acad. Sci. USA 99, 12562–12566 (2002).
52. Martin, R. M. Electronic Structure: Basic Theory and Practical Methods (Cambridge Univ. Press, 2020).
53. Momma, K. & Izumi, F. VESTA: a three-dimensional visualization system for electronic and structural analysis. J. Appl. Crystallogr. 41, 653–658 (2008).

Acknowledgements
We thank B. Busemeyer, M. Lindsey, F. Ma, M. A. Morales, M. Motta, A. Sengupta, S. Sorella and Y. Yang for discussions. S.C. thanks the Center for Computational Quantum Physics, Flatiron Institute, for support and hospitality. We acknowledge financial support from the following grant: US Department of Energy (DOE) grant DE-SC0001303 (S.C.). The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript. We thank William & Mary Research Computing and Flatiron Institute Scientific Computing Center for computational resources and technical support. The Flatiron Institute is a division of the Simons Foundation.

https://doi.org/10.1038/s43588-022-00350-w
Author contributions
S.C. and S.Z. conceived the project. S.C. executed the research with support and supervision from S.Z. S.C. and S.Z. wrote the manuscript. S.Z. obtained funding.

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/s43588-022-00350-w.

Correspondence and requests for materials should be addressed to Siyuan Chen.

Peer review information Nature Computational Science thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Jie Pan, in collaboration with the Nature Computational Science team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2022