Learn2Hop: Learned Optimization on Rough Landscapes With Applications to Atomic Structural Optimization

Amil Merchant [1 2] Luke Metz [1] Sam Schoenholz [1] Ekin Dogus Cubuk [1]

Abstract

Optimization of non-convex loss surfaces containing many local minima remains a critical problem in a variety of domains, including operations research, informatics, and material design. Yet, current techniques either require extremely high iteration counts or a large number of random restarts for good performance. In this work, we propose adapting recent developments in meta-learning to these many-minima problems by *learning* the optimization algorithm for various loss landscapes. We focus on problems from atomic structural optimization—finding low energy configurations of many-atom systems—including widely studied models such as bimetallic clusters and disordered silicon. We find that our optimizer learns a 'hopping' behavior which enables efficient exploration and improves the rate of low energy minima discovery. Finally, our *learned optimizers* show promising generalization with efficiency gains on never before seen tasks (e.g. new elements or compositions). Code is available at `https://learn2hop.page.link/github`.

1. Introduction

Efficient global optimization remains a problem of general research interest, with applications to a range of fields including operations design (Ryoo & Sahinidis, 1995), network analysis (Abebe & Solomatine, 1998), and bioinformatics (Liwo et al., 1999). Within the fields of chemical physics and material design, efficient global optimization is particularly important for finding low potential energy configurations of isolated groups of atoms (clusters) and periodic systems (crystals); identifying low energy minima in these cases can yield new stable structures to be experimentally produced and tested for a wide variety of industrial or scientific applications (Wales & Doye, 1997; Flikkema & Bromley, 2004). However, even simple examples with a few atoms have quite complex minima structures. Numeric approximations suggest that systems of only 147 atoms could have 10^{60} distinct minima (Tsai & Jordan, 1993).

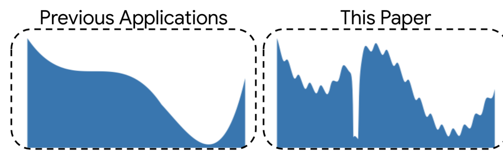

Figure 1. Schematic diagram of the difficulties of global optimization on rough loss landscapes. In contrast to prior work where loss surfaces are approximately convex, this paper focuses on global optimization problems where minima are numerous and there is unlikely to be a low loss path between local minima.

Global optimization problems can also be quite difficult when high loss barriers exist between local minima, as depicted in Figure 1.[1] Despite being NP-hard in the worst case, significant work has been put into developing optimization techniques for these structure prediction tasks. Nonetheless, classical approaches to this problem continue to face a number of drawbacks including requirements of: a significant number of steps (Wales & Doye, 1997; Pickard & Needs, 2011), carefully selected hand-crafted features, or sensitive dependence on learning rate schedules (Bitzek et al., 2006).

In this work, we propose adopting a new class of strategies to these global optimization problems: *learned optimization* (Bengio et al., 1992; Andrychowicz et al., 2016; Metz et al., 2018). Here, hand-designed update equations are replaced with a learned function parameterized by a neural network and trained via meta-optimization. While this strategy has shown promise for training neural networks (Metz et al., 2020) where falling into bad local minima is not a concern (Choromanska et al., 2015; Luo et al., 2018), current techniques fail to prioritize global minimum discovery or have not been tested on rough loss landscapes with many unconnected local minima.

[1]Google Research, Mountain View, California, USA [2]This work was done as part of the Google AI Residency Program (https://research.google/careers/ai-residency/). Correspondence to: Amil Merchant <amilmerchant@google.com>, Ekin D. Cubuk <cubuk@google.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

[1]See Wales & Doye (1997) for examples of difficulties in optimizing Lennard-Jones potentials.

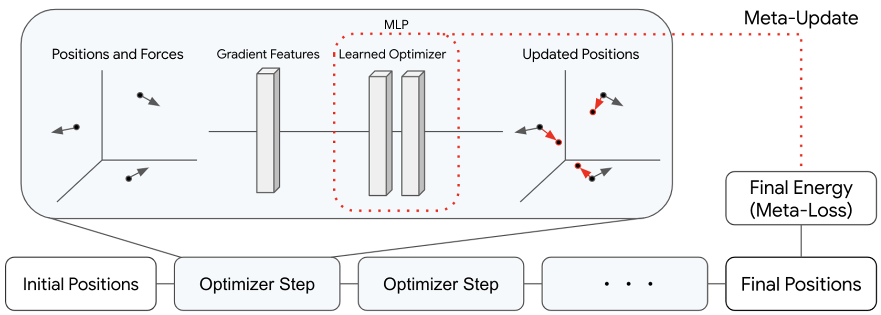

Figure 2. Schematic diagram of a learned optimizer for atomic structural optimization. Positions and forces are computed from physical simulations (i.e. empirical potentials). Gradients are accumulated, featurized, and inputted to a shallow neural network that updates positions. Although the diagram specifies a problem in $\mathcal{R}^3$, the learned optimizer framework can be applied to arbitrary global optimization problem.

In this paper, we show that learned optimizers can offer a substantial improvement over classical algorithms for these sorts of global optimization problems. To this end, we present several modifications of learned optimizers required to effectively train models which can find low energy states of many-minima loss surfaces. Using several canonical problems in atomic structural optimization, we demonstrate that the learned optimizers outperform their classical counterparts when trained and tested on similar systems and—more surprisingly—are able to generalize to unseen systems. The specific contributions of this paper are as follows:

1. Novel parameterizations of learned optimizers prioritize global minimum discovery (Section 3) and yield improvements on benchmark tasks consisting of single atom types (Section 4).
2. Analysis of learned optimizer behavior showcases an automatically-learned 'hopping' behavior which enables efficient exploration of minima (Section 5).
3. Results for bimetallic systems show that our learned optimizers can generalize beyond the examples seen during training, yielding gains in efficiency and performance over commonplace optimization techniques such as Basin Hopping (Section 6).

2. Background / Related Work

2.1. Atomic Empirical Potentials

Atomic structure optimization often requires finding the lowest energy configuration of a system (Oganov et al., 2019). However, accurate calculation of energies is expensive, often requiring quantum mechanical simulations such as DFT (Hohenberg & Kohn, 1964). In this work, we instead use approximations of the potential energy known as empirical potentials, that are not only simple and efficient to calculate but also have minima that correlate to those found by more accurate calculations (Tran & Johnston, 2011). The empirical potentials studied in this paper are as follows:

Lennard-Jones Clusters are often used in the modelling of spherically-symmetric particles in free-space such as noble gasses or methane and are the archetypal model for a simple-to-compute potential (Jones & Chapman, 1924; Doye et al., 1999). The total energy of the system is defined only by pairwise distances, denoted d_{ij}:

$$\sum_i \sum_{j>i} \epsilon \left[\left(\frac{d_0}{d_{ij}} \right)^{12} - 2 \left(\frac{d_0}{d_{ij}} \right)^{6} \right] \qquad (1)$$

where ϵ is the minimum two-particle energy and d_0 is the distance where this occurs. Despite the apparent simplicity, the minima structures of these systems are complex and vary significantly based on the number of atoms (Doye et al., 1999; Wales & Doye, 1997).[2] For example, the 13 and 19 atom systems display concentrated "funnels", where many local minima and the global minimum are connected via low energy paths. In contrast, the 38 and 75 atom counts display complex "double-funneled" landscapes, where there are two distinct sets of minima that are dynamically inaccessible due to a high energy barrier between the two.

Gupta Clusters are similar to the Lennard-Jones model in approximating the energy of sets of atoms in free-space, yet they are significantly more complex due to the inclusion of a second-moment approximation of the tight-binding Hamiltonian (Gupta, 1981; Michaelian et al., 1999; Sutton, 1993; Rosato et al., 1989). In this paper, we focus on both single element and bimetallic forms using the elements Ag, Au, Pd, and Pt. Energy equations and the constant values for all systems are provided in Appendix A.2.

[2]Diagrams depicting the local minima structures and the lowest energy paths between minima for particularly interesting cluster sizes can be viewed at `http://doye.chem.ox.ac.uk/research/forest/LJ.html`

Stillinger-Weber (SW) potentials (Stillinger & Weber, 1985) provide more accurate estimations for energies of semiconductors. This empirical potential introduces a three-body angular term between atoms, making the corresponding loss landscape significantly more difficult to optimize. In this paper, the SW potential is used to model silicon crystals. This benchmark is distinct from the others in the use of periodic boundary conditions, so that atomic structures are tiled in space. The associated energy equation and parameters are provided in Appendix A.3.

2.2. Optimization Methods from Structure Prediction

Early approaches to structure prediction problems simply initialized the particle positions at hand-crafted, physically-motivated structures, before applying gradient descent (Hoare & Pal, 1971; Farges et al., 1985; Doye et al., 1995). This technique proved effective for simple cluster systems such as Lennard-Jones but faced difficulty scaling to more complex potentials (such as those with angular dynamics). Classic optimization approaches such as Basin Hopping (Wales & Doye, 1997) and Simulated Annealing (Kirkpatrick et al., 1983; Biswas & Hamann, 1986) resulted in significant improvements and helped discover the minima for a variety of structures. However, these techniques end up requiring high step counts and may only find the global minimum in the limit of infinite optimization steps.

Modern molecular dynamics systems use a variety of techniques for optimization. Quasi-Newton techniques such as BFGS and damped Beeman dynamics (Beeman, 1976) are popular within libraries such as QuantumEspresso (Giannozzi et al., 2009). Alternate strategies include Fast Inertial Relaxation Engine–referred to as FIRE (Bitzek et al., 2006)–which adaptively modifies the velocity over the course of training and Ab Initio Random Structure Search (Pickard & Needs, 2006; 2011; Zilka et al., 2017). However, these techniques often rely on heuristics or require a large number of restarts before reaching the global minimum.

While this work only uses traditional empirical potentials, machine learning has also been used to create empirical potentials, such as those utilizing graph convolutions (Gilmer et al., 2017; Schütt et al., 2017; Cheon et al., 2020). These models are becoming a popular option for speeding up optimization. However, we note that the approach presented in this paper is complementary; the two could be combined so that both the potential and optimizer are learned.

2.3. Learned Optimization

Learned optimization (Bengio et al., 1992; Andrychowicz et al., 2016; Wichrowska et al., 2017; Lv et al., 2017; Metz et al., 2018; 2019b; Gu et al., 2019; Metz et al., 2020), has recently become a popular meta-optimization task, where updates are a function of the gradients, parameterized by a neural network. In the traditional setup depicted in Figure 2, training a learned optimizer consists of an inner-loop of optimization problems which are used to compute meta-updates to the learned optimizer parameters, referred to as the outer-loop (Wichrowska et al., 2017; Metz et al., 2018) .

In our case, the inner-loop consists of instantiations of atomic structure optimization problems, including a random initialization for atoms and a corresponding empirical potential to minimize. At each step in the inner-loop of meta-training, atomic forces are computed, featurized, and then input to the learned optimizer which computes updates to the particles. These steps are then repeated, which is often referred to as an inner-trajectory or unroll.

For each inner-loop, a meta-loss is defined based on the optimization trajectory, commonly the average loss over the trajectory in prior work. If the unrolls were short, meta-training could be performed by gradient descent (Andrychowicz et al., 2016; Wichrowska et al., 2017). However, due to memory requirements and often ill-conditioned outer-loss surfaces (Metz et al., 2019a), meta-gradients are instead approximated via Evolutionary Strategies (ES) using antithetic samples (Williams, 1992; Salimans et al., 2017; Metz et al., 2019a). A central controller collects batches of meta-gradient estimates and updates the learned optimizer parameters, typically using Adam (Kingma & Ba, 2014).

A variety of architectures have been proposed for learned optimizers. Early work utilized RNNs in order to provide the network a state that can be automatically updated throughout the course of training (Andrychowicz et al., 2016). These models were quickly developed into optimizers that scale (Wichrowska et al., 2017), and are compute-efficient (Metz et al., 2018). Most closely related to our work is that of (Metz et al., 2019a; 2018) which is novel in its parameterization of the state and input features of the learned optimizer. Instead of providing an explicit memory (e.g. in a GRU), the learned optimizer is simplified to an MLP that is applied per parameter and is provided relevant features, such as the first and second moment estimates for the gradients. See Table 1 for all features used in the MLPs.

3. Modifications for Rough Landscapes

Adapting these learned optimizers to many-minima landscapes requires modifications to both the training and model itself to improve global optimization. For example, instead of the average loss of the optimization trajectory, the learned optimizer for atomic structure optimization only uses the final step loss. This training strategy prioritizes global minimum discovery at the expense of greater variance of the gradient with respect to learned optimizer weights. Additional modifications are detailed in the following sections and training details are in Appendix B.

Table 1. Features inputted into the learned optimizers. MLPOpt refers to the model by Metz et al. (2019a).

FEATURES	DESCRIPTION	IN MLPOPT
GRADIENTS	GRADIENTS	✓
POSITIONS	PARTICLE POSITIONS	✓
DECAYS	EMA OF 1ST AND 2ND MOMENTS	✓
ADAM-LIKE	INVERSE NORM AND MOMENT CORRECTION	✓
SINGULAR	NUMBER OF PARTICLES	✓
SPECIES	SPECIES IDENTITY	✓
STEP	TRAINING STEP SINE FEATURES	✓
RADIAL	RADIAL SYMMETRY FEATURES	

3.1. Features

We follow prior work (Metz et al., 2019a) in parameterizing the learned optimizer as an MLP. The input features are often inspired by popular optimization techniques and include estimates of the first and second moments to mimic Adam (Kingma & Ba, 2014). Table 1 describes all inputted features that are adopted from "MLPOpt", the learned optimizer described in Metz et al. (2019a).

Novel to our learned optimizers is the inclusion of Behler-Parrinello radial symmetry features (Behler & Parrinello, 2007; Artrith et al., 2013; Cubuk et al., 2015). Traditional learned optimizers update each parameter independently, yet in the case of atomic structure problems, particle behavior should depend on interactions with nearest neighbors. Radial symmetry functions provide these sorts of two-body interactions for a central atom by allowing updates to be defined by local neighborhoods. Simply put, these features ϕ are computed using a Gaussian kernel and summing over all neighbors of a central atom. Smooth cutoffs Γ are applied using the formulation by Behler & Parrinello (2007):

$$\phi_i = \sum_{j \neq i} \exp\left(-\eta d_{ij}^2\right) \Gamma(d_{ij})$$

$$\Gamma(d_{ij}) = \begin{cases} 0 \text{ if } d_{ij} > c \\ 0.5\left(\cos\left(\pi \cdot d_{ij}/c\right) + 1\right) \text{otherwise} \end{cases}$$

where c is a pre-defined cutoff set to 2.5 angstroms and η is a hyperparameter controlling the scale. η is set to one of $\{0.0009, 0.01, 0.02, 0.035, 0.06, 0.1, 0.2, 0.4\}$, yielding 8 radial features per atom type.

These features are then parameterized into a log-magnitude and direction representation (Andrychowicz et al., 2016):

$$\text{features } = \begin{cases} (\log|x|/p, \text{sign}(x)) & \text{if } x > \exp(-p) \\ (-1, x\exp(p)) & \text{if } x \leq \exp(-p) \end{cases}$$

where $p = 10$ is the default hyperparameter. These features are then input to the learned optimizer, a small 2-layer dense neural network (with a hidden size of 32), that acts component-wise. The network outputs magnitude m and unnormalized direction d per component, converted to the final update via:

$$\alpha \cdot d \cdot \text{sigmoid}(\beta \cdot m + \gamma) \tag{2}$$

where α, β, γ are scalars learned via meta-optimization.[3]

3.2. Meta-Training Stability

The rough loss landscapes discussed in this paper present two significant challenges with regards to meta-optimization: high curvature and infrequent training signal.

High curvature is an intrinsic problem to atomic structures. For example, with the Lennard-Jones potentials, the energy increases at a rate of d_{ij}^{-12} as $d_{ij} \rightarrow 0$ for all i, j. When the optimizer happens to bring two particles too close together, energy (loss) spikes can yield meta-gradients that destabilize learned optimizer training. Traditional strategies such as gradient norm clipping (Pascanu et al., 2013) were found to be ineffective in preventing divergence of the meta-optimization objective.

Infrequent and noisy training signals are also problematic as learned optimizers can find simple, stable optimization strategies such as gradient descent early in training. Most perturbations to gradient-descent methods will be noise and increase the final loss. The meta-optimization model must be sensitive enough to learn from the infrequent signal occurring when few individual instantiations of a learned optimizer find better minimia structure, rather than being pushed back to descent-like methods due to noise.

As mentioned in the background, many learned optimizers are trained with antithetic ES sampling (Salimans et al., 2017; Metz et al., 2019a) where meta-gradients are estimated via perturbations of the parameters:

$$\nabla_{meta} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I\sigma^2)}\left[\frac{L(\theta+\epsilon) - L(\theta-\epsilon)}{2\sigma^2}\right]\epsilon \tag{3}$$

where L is the loss, θ the learned optimizer parameters, and σ is the perturbation scale, set to 0.1. This strategy is particularly vulnerable to the optimization difficulties, as either direction of the parameter perturbation may lead to exploding gradients. Also, the variance of the estimator makes it more difficult to learn from the sparse rewards when optimizers find better local minima. To overcome these issues, we present two modifications to the meta-update which enable stable meta-training:

[3] Note, this output parameterization contrasts from Metz et al. (2019a), but experimental evidence showed that the traditional exponential update leads to divergent optimization trajectories.

Meta-loss Clipping (ESMC)
In order to prioritize signal from perturbations that find better minima and improve the meta-loss, we propose clipping the loss functions in the meta-gradient computation at the value found by the unperturbed parameters.

$$\mathbb{E}_\epsilon \left[\frac{\min\left[L(\theta), L(\theta+\epsilon)\right] - \min\left[L(\theta), L(\theta-\epsilon)\right]}{2\sigma^2} \right] \epsilon \quad (4)$$

where again $\epsilon \sim \mathcal{N}(0, I\sigma^2)$. Intuitively, this biases against directions of high curvature in meta-optimization and empirically showed improved results. This strategy has the added benefit of heavily clipping the gradients of examples where loss spikes when atoms become too close, at the cost of an additional meta-loss calculation for $L(\theta)$.[4]

Genetic Algorithms (GA)
Instead of relying on approximate meta-gradients, a simpler strategy perhaps is to adopt the perturbed parameters when they improve the meta-loss on a batch of random examples (Holland, 1992; Goldberg & Holland, 1988). To match the number of estimators of the meta-gradient used in ESMC, we use a population of size 80. At the end of each outer loop, the best performing parameters θ are kept and used for creating the next population by drawing from $\mathcal{N}(\theta, I\sigma^2)$ where $\sigma = 0.1$. By default, θ is kept constant when all samples perform worse than the baseline.

Comparison of Methods
A comparison of these strategies on a simplified learned optimizer setup is shown in Figure 3. The genetic-algorithm approach shows improvement early in meta-training which steadily converges to an optimizer where almost all initialization find the global minimum. In contrast, both ES and ESMC show a distinct transition in behavior around steps 300-400, which demarcates a transition from simple-to-learn descent behavior to more complex global minima discovery. The ESMC method is able to retain this behavior throughout meta-optimization, whereas traditional ES appears unstable and *forgets*. Overall, both learned optimizer modifications show significant improvements in convergence speed and stability when compared to vanilla ES. Details for this experiment can be found in Appendix B.

3.3. Additional Details

In order to provide consistent scaling when averaging meta-losses across tasks, we divide energies by the best minimum found from applying Adam to 150 random initializations. This normalizes all losses to so that -1 is the best minimum found by Adam, ensuring that optimizers trained on multiple particle counts are not biased towards larger systems where the energy scales are lower. For each inner-loop, we apply 50000 optimization steps before computing meta-gradients. Batched training occurs with 80 random initializations of atomic structure problems. Once meta-gradients are averaged, a central controller meta-updates the parameters of the learned optimizer via Adam with a learning rate of 10^{-2} (which decays exponentially by 0.98 every 10 steps). This repeats for a total of 1000 meta-updates.

Finally, as the learned optimizer training does not guarantee a local minimum is found at the end of an optimization trajectory, we add 1000 steps of GD at learning rate of 0.001. We evaluate all strategies using 150 random initializations and report the mean and minimum energies found.

[4]In practice, this does not require a 50% increase in meta-training time due to parallelization. On V100 GPUs, the increase in training time was as small as 15%.

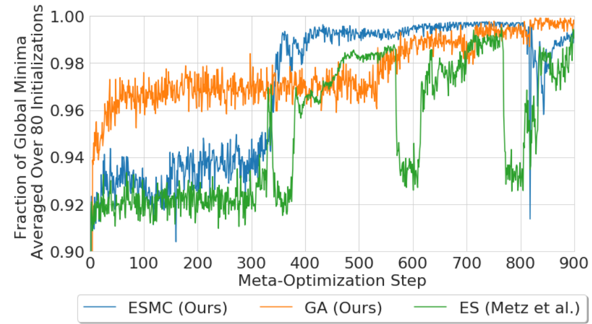

Figure 3. Comparison of training strategies for learned optimizers on Lennard-Jones clusters highlights the need for ESMC or GA. Here, we see that our learned optimizers improve performance with respect to stability and averaged meta-loss.

3.4. Implementation Details

The aforementioned potentials are coded using JAX-MD (Schoenholz & Cubuk, 2019). The learned optimizers are built in JAX (Bradbury et al., 2018) to take advantage of automatic differentiation and vectorization of the optimization simulation. The associated training and evaluation utilized V100 GPUs. For distributed training, the controller batches computation on up to 8 GPUs.

4. Experimental Results - Single Atom

We start with a simplified set of potentials comprised of a single type of atom. For Lennard-Jones, we present results when a model is trained on a diverse set of atom counts, specifically {13, 19, 31, 38, 55, 75}, which helps stabilize learned optimizer training and improve generalization beyond the training set. In the results, starting with Figure 4, the learned optimizer shows significant performance gains when compared to the benchmark optimization algorithms of Adam and FIRE. This improvement not only takes the form of better minima but also better average energy per initialization. Note, the dotted lines correspond to the atom counts used during training; that the learned optimizers per-

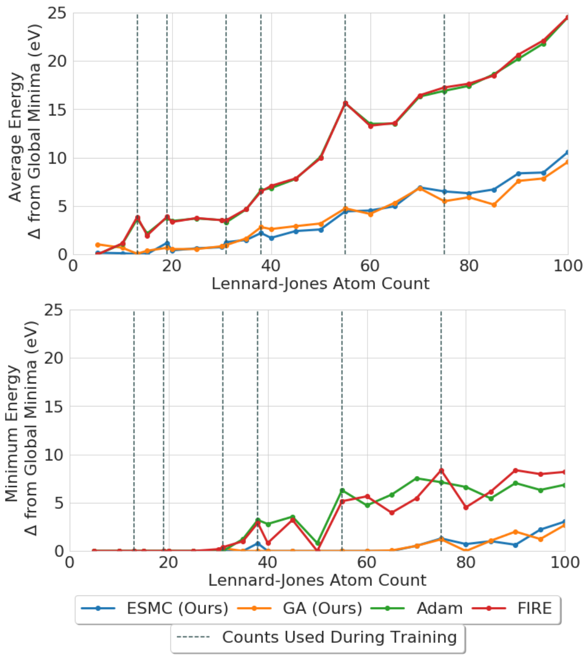

Figure 4. Comparison of the learned optimizers and baseline methods on Lennard-Jones clusters. The learned optimizers are trained on a subset of the atom counts (demarcated by dashed lines) and evaluated via 150 random initializations. When compared to Adam and FIRE, the learned optimizers generically improve both the average energy per initialization (top) and best minima found (bottom) on atom counts unseen during training.

form better between these lines shows that these learned optimizers generalize to tasks unseen during training yielding significant improvements over Adam and FIRE. Furthermore, the models generalize beyond the training distribution to tasks of up to 100 atoms.

Figure 5 analyzes the distribution of minima in greater detail for two canonical tasks: the 13 and 75 atom Lennard-Jones systems. The loss surface of the former is best described as a "funnel" and even traditional algorithms can find the global minimum in about 20/150 random initializations. On the other hand, the 75 atom Lennard-Jones system has a glassy, optimization landscape, where high energy barriers exist between local minima (Wales & Doye, 1997) and the global minimum is difficult to find.

Interestingly, we first find that the baselines of Adam and FIRE yield similar performance per task after extensive hyper-parameter tuning.[5] Nevertheless, both of our learned optimizer models show significant progress. With 13 atoms, the learned optimizers drastically increase the rate of global minimum discovery from 20/150 to above 140/150. With 75 atoms, the learned optimizers shift the distribution of minima found, finding minima within 1 eV of the global minimum compared to the 7 eV for Adam and FIRE.

[5] While this trend was common in our experiments, additional work would be necessary to thoroughly compare these optimizers.

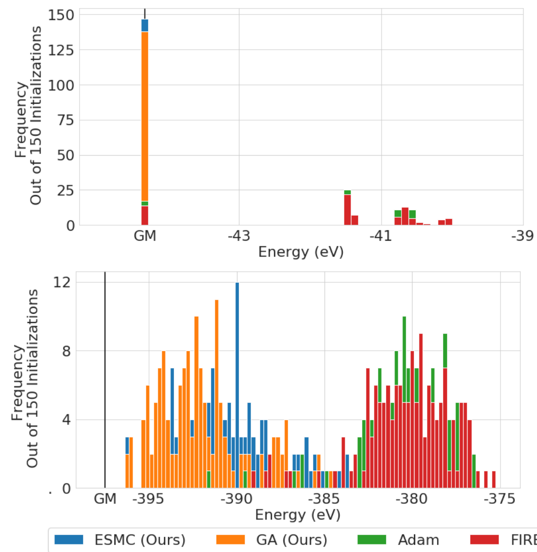

Figure 5. Distribution of minima found when baseline optimizers and the learned models are applied to 150 random initializations. For the Lennard-Jones task with 13 atoms (top), the learned optimizer find the global minimum in approximately 140 out of 150 trials, significantly better than the 20 of Adam and FIRE. Similar improvements are seen for the Lennard-Jones task with 75 atoms (bottom) where learned optimizers improve the minima found.

Similar results were obtained when learned optimizers were extended to the Gupta or SW potentials, when modeling 55-atom gold clusters and 64-atom silicon crystals respectively. Table 2 shows that the learned optimizers routinely surpass Adam and FIRE baselines and outperform Basin Hopping in a step-matched comparison.[6] For silicon crystals, we note that the large gap between the global minimum and energies found arises from the difficulty of optimizing 64 atoms; due to the size, the problem reduces to finding finding stable amorphous structures (low energy local minima) rather than the true global minimum (Barkema & Mousseau, 1996).

5. Behavioral Analysis

As the update function is parameterized by a neural network, it is unclear how learned optimizers improve atomic structural prediction. To explore the behavior, we provide three analyses showcasing an emergent 'hopping' behavior and what features are critical for learned optimizer performance.

5.1. Loss Trajectories

In Figure 6 (top), we show loss trajectories when the learned optimizer is applied to the Lennard-Jones task with 13 atoms. Interestingly, the behavior of the loss function is not monotonic. While the model does rapidly descend into individual

[6] Step-matched refers to an equivalent number of inner optimization steps. For details, please see the Appendix C.2

Table 2. Learned optimizers show improvement across all tested potential types. For each model and evaluation, the average and minimum are computed across 150 random initializations. All energies are reported in units of eV, and GM denotes the global minimum energy.

				Step-Matched Baselines			Learned Optimizer	
Potential	Evaluation # Atoms	GM	Metric	Adam	FIRE	Basin Hopping[7]	ESMC	GA
Lennard-Jones	13	-44.33	Min	**-44.33**	**-44.33**	**-44.33**	**-44.33**	**-44.33**
			Mean	-40.58	-40.45	-43.49	-44.26	**-44.31**
	75	-397.49	Min	-390.34	-389.12	-392.16	**-396.24**	**-396.28**
			Mean	-380.52	-380.23	-381.49	-390.33	**-390.92**
Gupta Gold	55	-181.89	Min	-180.94	-181.75	**-181.89**	**-181.89**	**-181.89**
			Mean	-179.94	-180.94	-181.38	-181.51	**-181.61**
SW silicon	64	-277.22	Min	-60.08	-261.44	-261.64	-262.95	**-264.17**
			Mean	-256.83	-257.01	-259.14	-260.14	**-261.81**

basins, many of these models display spikes in loss or 'hopping' behavior where the model transitions between basins of different local minima at an erratic interval. More over, the optimizers have discovered characteristics that determine when to leave their basin. Figure 6 (bottom) explores these trajectories in greater details, by filtering the 'lucky' intializations that lead to the correct global minimum via Adam only. In cases where the parameters start in the correct basin, the learned optimizers performs better, acting like traditional Adam. For worse random starts, the learned optimizer will descend and then 'hop' between basins.

5.2. Behavior at Minima

This 'hopping' behavior appears to be key to the learned optimizer performance. Inspired by Maheswaranathan et al. (2021), we fix the position at the global minimum and compute the learned optimizer update over the course entire optimization trajectory. This strategy removes the influence of gradients in the learned optimizer (as they are zero) and helps visualize the behavior, as a function of the optimization step. In Figure 6 (middle), repeatedly applying the learned optimizer and measuring the update magnitude displays these 'hops,' indicating that the model places emphasis on exploration and hopping between basins midway through these optimization trajectories (despite not receiving gradient signal to do so).

5.3. Feature Importance

Table 3 provides an ablation study to explain what input features to the learned optimizer are most important. To clarify the difference in performance, results are presented for a learned optimizer trained only on the 75 atom Lennard-Jones system (similar results were found for other clusters and crystals). To account for training instability, each result is the median over 10 shortened runs of 650 meta-updates.

We first see slight improvements coming from the addition of exponential moving averages of the gradient, similar to the benefits of momentum in stochastic optimization.

Table 3. Improvement arising from additional features shows that Sine and Radial features boost model performance, suggesting information about step count and local neighborhoods of atoms are helpful for optimization. Results are the median performance out of 10 random training seeds. Note results are worse than Table 2 due to shortened training schedules. Lower is better.

Optimizers	Minimum Energy (eV)	
Baseline		
Adam	-390.3	-390.3
Learned Optimizer	Δ from Adam ESMC (ours)	Δ from Adam GA (ours)
(1) Gradients	+0.2	-0.8
(2) Positions	+0.2	-1.2
(3) Decays	-2.0	-2.0
(4) Adam-Like	-1.0	-2.8
(5) Singular	-1.0	-2.9
(6) Species	-0.5	-2.4
(7) Sine	-2.3	-3.6
(8) Radial	-3.4	-4.4

However, what is most critical to model performance is the training step sine features and the radial symmetry functions. The sine features encode the optimization step via sine waves of various timescales ($\{1, 3, 10, 30, 100, 300, 1000, 3000, 10000\}$), and we hypothesize that these are helpful as they provides signal for when the learned optimizer should perform exploration and exploitation. Otherwise, the models tend to learn monotonic behavior that is similar in spirit to the Adam solutions. Finally, the newly introduced radial symmetry features also provide significant improvement, suggesting that per-position optimization is sub-optimal and rather information about particle neighbors (other than just gradients) is informative for optimization.

Overall, the behavioral analyses show that 'hopping' behavior is critical to the performance of learned optimizers; however, given that an equivalent number of traditional Basin Hopping steps yields worse performance suggests more complex behavior also occurs.

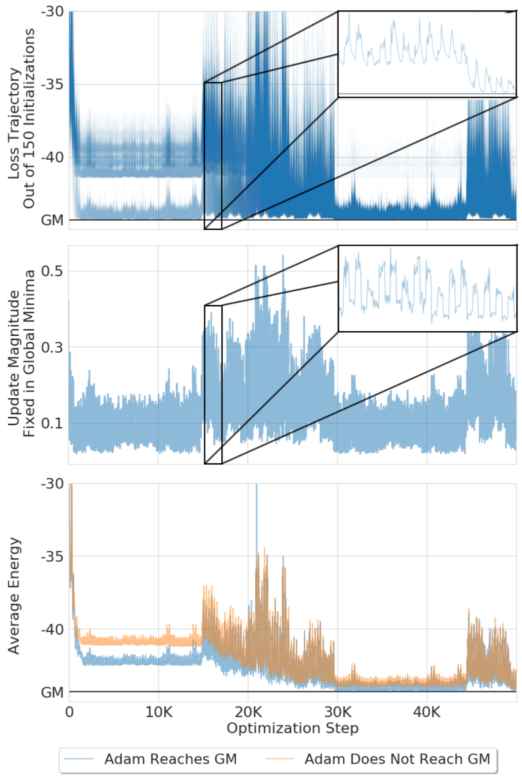

Figure 6. Behavioral analyses of learned optimizers find that examples 'hop' between basins rather than descent behavior. This is seen in the behavior of individual trajectories (top), where each trajectory has an opacity of 0.02 (so darker regions corresponds a greater number of examples). These 'hops' are further supported by the spiking update magnitudes when fixed to a local minimum, suggesting that learned optimizers prioritize exploration of various basins (middle). For both of these diagrams, we zoom-in on a single trajectory, showing how the 'hops' arise from large updates, followed by periods of descent. We also find that this hopping behavior corrects unlucky, random initializations that would not find the global minimum via Adam (bottom).

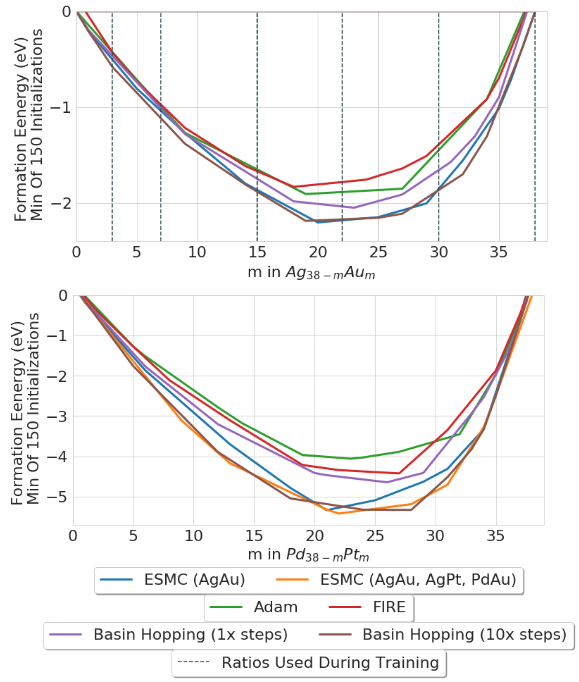

Figure 7. Results for the bimetallic AgAu (top) and PdPt (bottom) clusters. For ESMC (AgAu), the learned optimizer is trained on only a subset of the possible ratios between Ag and Au. ESMC (AgAu, AgPt, PdAu) is trained only with AgAu, AgPt, PdAu clusters. On both AgAu and PdPt systems, both of our learned optimizers significantly outperform the baselines of Adam, FIRE, and step-matched Basin Hopping, which shows that the learned optimizers can generalize to new ratios or combinations of seen elements and new elements entirely unseen during training.

6. Experimental Results - Bimetallic Clusters

Having found that learned optimizers perform well in the case of single atom systems, we introduce additional complexity and explore generalization performance of the learned optimizers using bimetallic clusters. These systems are particularly interesting as purely gradient-based optimization methods such as Adam or FIRE fail, unable to pass the large energy barriers between local minima. These potentials also allow for exploration of whether the learned behavior can transfer, a promising sign for the usage of these models in material design or crystal discovery.

For the bimetallic clusters, we focus on the Gupta potential, whose parameters are modified to correspond to specific pairwise interactions (Gupta, 1981). The constant values used can be found in Appendix A.2. First, the results of training learned optimizers on bimetallic clusters comprising of Silver (Ag) and Gold (Au) are shown in Figure 7 (top). Fixing the total number of atoms at 38, we train the learned optimizer on $Ag_{38-m}Au_m$ for $m \in \{3, 7, 15, 22, 30, 38\}$ and test on all values of m. We present the convex hull of the formation energies of the clusters, as in equilibrium when excess silver and gold particles are present, only these clusters will be stable. The graphs show the robust empirical performance of learned optimizers, significantly outperforming Adam, FIRE, and a step-matched Basin Hopping benchmark. Only after 10x evaluation steps does Basin Hopping compete with the performance of the learned optimizer. This result indicates that the learned optimizers generalize as few of the AgAu clusters were used in training.

In the context of material design and crystal discovery, another core question is whether the learned optimizers will generalize beyond the set of atoms used for training. In Figure 7 (bottom) we show the results from both the AgAu model described above and a second model trained on AgAu, AgPt, and PdAu. For both, we test on clusters of PdPt, which is not in the training set of either model. The learned optimizers show successful transfer performance, exceeding the step-matched Basin Hopping results. Only after 10x the number of evaluation steps can Basin Hopping compete with the learned optimizers (even after tuning, see Appendix C.2. While increasing the diversity of training tasks does improve generalization performance, both optimizers show an ability to transfer to unseen elements or combinations, a promising sign for this strategy of learning to optimize.[8]

7. Conclusion

With current optimization techniques in material design and chemical physics requiring hand-crafted features or significant evaluation time, this paper explores the idea of global minimum discovery using learned optimizers. Although novel adaptations are required from the current state-of-the-art learned optimizers, we show that the resulting models can beat current baseline optimization techniques such as Adam and FIRE, not only in terms of minima discovery but also in terms of average energy per initialization. Better yet, these learned optimizers show signs of transference between potentials of similar varieties (ex: clusters parameterized by the Lennard-Jones and Gupta potentials), even on never before seen elements or combinations. Although a single optimizer for all task remains a goal for future work, learned optimizers show promise in automatically finding minima in complex optimization landscapes. We hope that the resulting models can aid in the design of new materials (e.g. for addressing energy challenges).

Acknowledgements

We thank Jascha Sohl-dickstein and the Google Brain team for their help with the project.

References

Abebe, A. and Solomatine, D. Application of global optimization to the design of pipe networks. In *Proc. 3rd International Conference on Hydroinformatics, Copenhagen*, pp. 989–996, 1998.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pp. 3981–3989, 2016.

Artrith, N., Hiller, B., and Behler, J. Neural network potentials for metals and oxides–first applications to copper clusters at zinc oxide. *physica status solidi (b)*, 250(6): 1191–1203, 2013.

Barkema, G. and Mousseau, N. Event-based relaxation of continuous disordered systems. *Physical review letters*, 77(21):4358, 1996.

Beeman, D. Some multistep methods for use in molecular dynamics calculations. *Journal of Computational Physics*, 20(2):130 – 139, 1976. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(76)90059-0. URL http://www.sciencedirect.com/science/article/pii/0021999176900590.

Behler, J. and Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

Bengio, S., Bengio, Y., Cloutier, J., and Gecsei, J. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pp. 6–8. Univ. of Texas, 1992.

Biswas, R. and Hamann, D. R. Simulated annealing of silicon atom clusters in langevin molecular dynamics. *Phys. Rev. B*, 34:895–901, Jul 1986. doi: 10.1103/PhysRevB.34.895. URL https://link.aps.org/doi/10.1103/PhysRevB.34.895.

Bitzek, E., Koskinen, P., Gähler, F., Moseler, M., and Gumbsch, P. Structural relaxation made simple. *Phys. Rev. Lett.*, 97:170201, Oct 2006. doi: 10.1103/PhysRevLett.97.170201. URL https://link.aps.org/doi/10.1103/PhysRevLett.97.170201.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. JAX: composable transformations of Python+NumPy programs. 2018. URL http://github.com/google/jax.

Cheon, G., Yang, L., McCloskey, K., Reed, E. J., and Cubuk, E. D. Crystal structure search with random relaxations using graph networks. *arXiv preprint arXiv:2012.02920*, 2020.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204, 2015.

Cubuk, E. D., Schoenholz, S. S., Rieser, J. M., Malone, B. D., Rottler, J., Durian, D. J., Kaxiras, E., and Liu, A. J. Identifying structural flow defects in disordered solids using machine-learning methods. *Physical review letters*, 114(10):108001, 2015.

Doye, J. P., Wales, D. J., and Berry, R. S. The effect of the range of the potential on the structures of clusters. *The Journal of chemical physics*, 103(10):4234–4249, 1995.

[8]The improvement in step count only occurs after training the learned optimizer, which is expensive. The transference results, however, are promising, as this cost could be offset by the optimizer being used to efficiently evaluate a variety of element pairs.

Doye, J. P. K., Miller, M. A., and Wales, D. J. Evolution of the potential energy surface with size for lennard-jones clusters. *The Journal of Chemical Physics*, 111(18):8417–8428, 1999. doi: 10.1063/1.480217. URL https://doi.org/10.1063/1.480217.

Farges, J., De Feraudy, M., Raoult, B., and Torchet, G. Cluster models made of double icosahedron units. *Surface Science*, 156:370 – 378, 1985. ISSN 0039-6028. doi: https://doi.org/10.1016/0039-6028(85)90596-5. URL http://www.sciencedirect.com/science/article/pii/0039602885905965.

Flikkema, E. and Bromley, S. T. Dedicated global optimization search for ground state silica nanoclusters. *The Journal of Physical Chemistry B*, 108(28):9638–9645, 2004. doi: 10.1021/jp049783r. URL https://doi.org/10.1021/jp049783r.

Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Chiarotti, G. L., Cococcioni, M., Dabo, I., et al. Quantum espresso: a modular and open-source software project for quantum simulations of materials. *Journal of physics: Condensed matter*, 21(39): 395502, 2009.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Goldberg, D. E. and Holland, J. H. Genetic algorithms and machine learning. 1988.

Gu, K., Greydanus, S., Metz, L., Maheswaranathan, N., and Sohl-Dickstein, J. Meta-learning biologically plausible semi-supervised update rules. *bioRxiv*, 2019.

Gupta, R. P. Lattice relaxation at a metal surface. *Phys. Rev. B*, 23:6265–6270, Jun 1981. doi: 10.1103/PhysRevB.23.6265. URL https://link.aps.org/doi/10.1103/PhysRevB.23.6265.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2020. URL http://github.com/google/flax.

Hoare, M. and Pal, P. Physical cluster mechanics: Statics and energy surfaces for monatomic systems. *Advances in Physics*, 20(84):161–196, 1971.

Hohenberg, P. and Kohn, W. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

Holland, J. H. Genetic algorithms. *Scientific american*, 267 (1):66–73, 1992.

Jones, J. E. and Chapman, S. On the determination of molecular fields.—i. from the variation of the viscosity of a gas with temperature. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106(738): 441–462, 1924. doi: 10.1098/rspa.1924.0081. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1924.0081.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Liwo, A., Lee, J., Ripoll, D. R., Pillardy, J., and Scheraga, H. A. Protein structure prediction by global optimization of a potential energy function. *Proceedings of the National Academy of Sciences*, 96(10):5482–5485, 1999. ISSN 0027-8424. doi: 10.1073/pnas.96.10.5482. URL https://www.pnas.org/content/96/10/5482.

Luo, J., Wu, C., and Lee, J. No spurious local minima in a two hidden unit relu network. January 2018. 6th International Conference on Learning Representations, ICLR 2018 ; Conference date: 30-04-2018 Through 03-05-2018.

Lv, K., Jiang, S., and Li, J. Learning gradient descent: Better generalization and longer horizons. *arXiv preprint arXiv:1703.03633*, 2017.

Maheswaranathan, N., Sussillo, D., Metz, L., Sun, R., and Sohl-Dickstein, J. Reverse engineering learned optimizers reveals known and novel mechanisms, 2021. URL https://openreview.net/forum?id=y_pDlU_FLS.

Metz, L., Maheswaranathan, N., Nixon, J., Freeman, D., and Sohl-Dickstein, J. Learned optimizers that outperform sgd on wall-clock and test loss. In *Proceedings of the 2nd Workshop on Meta-Learning. MetaLearn*, 2018.

Metz, L., Maheswaranathan, N., Nixon, J., Freeman, D., and Sohl-Dickstein, J. Understanding and correcting pathologies in the training of learned optimizers. In *International Conference on Machine Learning*, pp. 4556–4565, 2019a.

Metz, L., Maheswaranathan, N., Shlens, J., Sohl-Dickstein, J., and Cubuk, E. D. Using learned optimizers to make models robust to input noise. *arXiv preprint arXiv:1906.03367*, 2019b.

Metz, L., Maheswaranathan, N., Freeman, C. D., Poole, B., and Sohl-Dickstein, J. Tasks, stability, architecture, and compute: Training more effective learned optimizers, and using them to train themselves. *arXiv preprint arXiv:2009.11243*, 2020.

Michaelian, K., Rendón, N., and Garzón, I. Structure and energetics of ni, ag, and au nanoclusters. *Physical Review B*, 60, 07 1999. doi: 10.1103/PhysRevB.60.2000.

Oganov, A. R., Pickard, C. J., Zhu, Q., and Needs, R. J. Structure prediction drives materials discovery. *Nature Reviews Materials*, 4(5):331–348, 2019.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. PMLR, 2013.

Paz-Borbón, L. O., Johnston, R., Barcaro, G., and Fortunelli, A. Structural motifs, mixing, and segregation effects in 38-atom binary clusters. *The Journal of chemical physics*, 128 13:134517, 2008.

Pickard, C. J. and Needs, R. J. High-pressure phases of silane. *Phys. Rev. Lett.*, 97:045504, Jul 2006. doi: 10.1103/PhysRevLett.97.045504. URL https://link.aps.org/doi/10.1103/PhysRevLett.97.045504.

Pickard, C. J. and Needs, R. J. Ab initiorandom structure searching. *Journal of Physics: Condensed Matter*, 23(5):053201, jan 2011. doi: 10.1088/0953-8984/23/5/053201. URL https://doi.org/10.1088%2F0953-8984%2F23%2F5%2F053201.

Rosato, V., Guillope, M., and Legrand, B. Thermodynamical and structural properties of f.c.c. transition metals using a simple tight-binding model. *Philosophical Magazine A*, 59(2):321–336, 1989. doi: 10.1080/01418618908205062. URL https://www.tandfonline.com/doi/abs/10.1080/01418618908205062.

Ryoo, H. S. and Sahinidis, N. V. Global optimization of nonconvex nlps and minlps with applications in process design. *Computers & Chemical Engineering*, 19(5):551–566, 1995.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Schoenholz, S. S. and Cubuk, E. D. Jax, md: End-to-end differentiable, hardware accelerated, molecular dynamics in pure python. *arXiv preprint arXiv:1912.04232*, 2019.

Schütt, K., Kindermans, P.-J., Sauceda, H., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 992–1002, 2017.

Stillinger, F. H. and Weber, T. A. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.

Sutton, A. P. *Electronic structure of materials*. Clarendon Press, 1993.

Tran, D. and Johnston, R. Study of 40-atom pt-au clusters using a combined empirical potential-density functional approach. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 467:2004–2019, 05 2011. doi: 10.1098/rspa.2010.0562.

Tsai, C. J. and Jordan, K. D. Use of an eigenmode method to locate the stationary points on the potential energy surfaces of selected argon and water clusters. *The Journal of Physical Chemistry*, 97(43):11227–11237, 1993. doi: 10.1021/j100145a019. URL https://doi.org/10.1021/j100145a019.

Wales, D. J. and Doye, J. P. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., and Sohl-Dickstein, J. Learned optimizers that scale and generalize. *arXiv preprint arXiv:1703.04813*, 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Zilka, M., Dudenko, D. V., Hughes, C. E., Williams, P. A., Sturniolo, S., Franks, W. T., Pickard, C. J., Yates, J. R., Harris, K. D. M., and Brown, S. P. Ab initio random structure searching of organic molecular solids: assessment and validation against experimental data. *Phys. Chem. Chem. Phys.*, 19:25949–25960, 2017. doi: 10.1039/C7CP04186A. URL http://dx.doi.org/10.1039/C7CP04186A.