MACHINE LEARNING METHODS FOR THE ACCELERATED GLOBAL STRUCTURAL OPTIMIZATION OF THIOLATE-PROTECTED GOLD NANOCLUSTERS

by

Sam Walton Norwood

A thesis submitted to Johns Hopkins University in conformity with the requirements for

the degree of Master of Science in Engineering

Baltimore, Maryland
October 2020

Abstract

Modern computational chemistry techniques allow for the calculation of a wide set of material properties at the level of quantum physics, but such calculations require as input the atomic structure of the material in question. The first-principles prediction of a substance's atomic structure from knowledge of its composition is a standing challenge in chemistry and materials science, and this thesis documents efforts to surmount this challenge for a model system of thiolated gold nanoclusters. We employ a pool-based genetic algorithm to efficiently search configuration space for global optima, learning the most likely structures for a given ligated cluster composition by iteratively selecting and recombining elements from the stablest-yet-discovered examples. In previous work, density functional theory calculations were used to determine the stability of each new structure discovered by the genetic algorithm, but this approach scales poorly for ligand-terminated systems, which have more atoms and more geometric and electronic degrees of freedom. To extend the capabilities of our genetic algorithm and bring ligated systems within reach, we accelerate energetic evaluation by implementing a class of machine-learned interatomic potentials known as moment tensor potentials. After being initialized on a small set of *ab initio* structure-energy data, these potentials can be used to calculate energies in good agreement with DFT and to directly optimize newly generated structures via gradient descent. We make use of an active learning approach to select optimal subsets of candidate structures for the training of moment tensor potentials, to quantify the reliability of energetic evaluations by these potentials, and to prevent unrealistic structures from being propagated in the course of the genetic algorithm. By tailoring the training set to emphasize low-energy candidates, we help our potentials to learn with high accuracy the evolving hull of lowest-energy structures observed so far. Applying these methods, we study the impact of ligand substitution on the ground state structure of $Au_{18}(SR)_{14}$ and report new ground states for R = CH_3.

Advisor: Dr. Tim Mueller

Acknowledgements

First and foremost, the full measure of my gratitude is due to Dr. Mueller, whose wise and encouraging guidance was essential to this project. As an educator and mentor, he first sparked my interest in computational chemistry and materials science, and, in short, I was lucky to have known and worked for him in my time at Hopkins.

I'm equally indebted to my collaborators on this project. Yunzhe Wang was the architect of the active learning approach so fundamental to the work herein, not to mention the primary developer and maintainer of the Cluster-GA code used in this project. Shanping Liu's pioneering work developing moment tensor potentials for single-element clusters was deeply informative; in my study of clusters with ligands, I followed the path she set down. I relied on Yunzhe and Shanping's guidance more times than I can recollect, and they helped me find the answers to questions big and small.

The same goes for the rest of the Mueller Research Group, and in particular I'd like to extend thanks to Chuhong Wang for her advice on weighted training strategies for moment tensor potentials, and to Alberto Hernandez for his help in understanding the similarity evaluation algorithm. Thanks to all in the group for being brilliant, interesting, and fun people to be around, even over Zoom.

Thanks in parting to the Department of Materials Science and Engineering for a fine education, and to Jeanine Majewski, our program coordinator, for being truly generous with her time.

Thanks to my father for keeping the coffee pot going, to Katie and Joe for their company, and to my grandmother, Mary "Mimi" Walton, for sending me frequent text messages of encouragement while I wrote this thesis.

Dedication

This thesis is dedicated to my late mother, Nancy Walton Norwood. I wouldn't be writing it without her. Above all, she taught me to value truth.

Contents

List of Figures

Chapter 1. Introduction

1.1 Nanoclusters

Nanoclusters are aggregates of atoms, typically metals, ranging in dimension from a few Angstroms to a few nanometers. It can be said that clusters in this size range bridge the bulk and quantum regimes: structure, properties, and stability have a strong dependence on size and composition and can all be starkly different from that observed on the macro-scale, with novel qualities resulting from electronic confinement and finite size effects [1]. Emergent from their unique electronic structure, nanoclusters exhibit many properties of scientific interest, including high surface reactivity, magnetic anisotropy, magnetoresistance, quantum confinement, variable metallicity, and plasmonic absorption [2].

With the advent of new techniques for atomically precise synthesis [3], these properties are made practically tunable through control of size and composition, leading to potential engineering applications in biomedical sensing and imaging [4], electrocatalysis [5], optoelectronics [6], light capture [7], and magnetic storage [8], among others [9]. Computational chemistry methods that treat electronic structure explicitly are capable of calculating many of the unique properties of nanoclusters, raising the possibility of using predictive computational techniques to guide the inverse design of clusters with desired features.

Before the properties of a cluster of a given size and composition can be computed, however, its structure must be determined: namely, we must identify the stablest possible configuration of the set of constituent atoms. This stablest configuration will, more rigorously, be the one that globally minimizes the free energy of the cluster with respect to the variation of its atomic coordinates. The so-called global minimum (GM) or ground state configuration is of interest

because it is most likely to correspond to the experimentally observed structure, although metastable structures residing in low-energy local minima may be kinetically favored depending on the synthetic conditions, and a distribution of higher-energy conformations will be thermally accessible and statistically populated at nonzero temperatures [10].

The formation of nanosized aggregates is a consequence of the stability of the bulk condensed phase, and the corresponding stability of small condensates against loss of atoms or fission [11]. Certainly, however, these small condensates are not stable against the addition of atoms; rather, they will spontaneously aggregate to form larger clusters, eventually growing to a bulk mass, unless low densities and temperatures are maintained. This means that pure elemental clusters are only stable under very narrow conditions, and so are primarily synthesized and studied in the gas phase [2].

For application in other conditions, nanoclusters must be protected from aggregation and reaction, and this is usually accomplished by surface termination with a ligating species. These ligands play a determining role in the ground state structure and properties of the cluster [12]. **Figure 1.1** compares the minimum-energy structures of bare Au_{10} and of its methanethiolate-protected form, $Au_{10}(SCH_3)_{10}$; the structures are markedly different. Moreover, the particular ligand chemistry (i.e., the active site and R-group) chosen for stabilization can modulate the properties of clusters of a given core composition [13]. Experimental work on protected nanoclusters has demonstrated ligation-driven control of both properties and structure [14, 15].

Among the best-studied ligated nanoclusters are those belonging to the gold-thiolate system. Over the past 25 years, dozens of varieties of thiolate-protected gold nanoclusters have been synthesized and characterized [16]. Currently, the smallest gold-thiolate nanocluster whose structure has been experimentally determined is $Au_{18}(SR)_{14}$, **Figure 1.2**, which was characterized

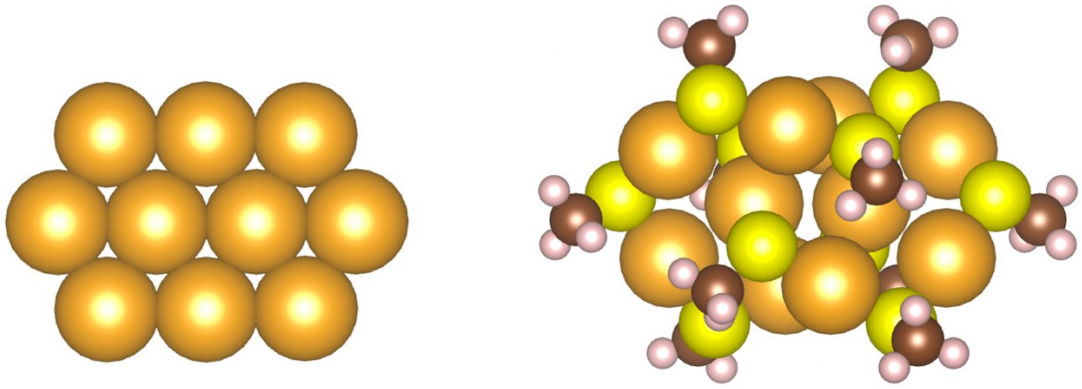

Figure 1.1: The minimum-energy structures of Au_{10} and $Au_{10}(SCH_3)_{10}$. The bare Au_{10} cluster adopts a planar close-packed configuration, while thiolate-protected $Au_{10}(SCH_3)_{10}$ is a catenane composed of two interlocking $Au_5(SCH_3)_5$ rings. These structures have been identified [19, 20] with genetic algorithms using density functional theory calculations for energy comparison.

by X-ray diffraction by both Jin [17] and Zhu [18] separately in 2015. Since the structure of this nanocluster is known, we take $Au_{18}(SR)_{14}$ as a model stoichiometry to test the capabilities of our group's genetic algorithm for cluster structure discovery.

This thesis will proceed as follows. Chapter 2 provides an overview of our group's genetic algorithm (GA) as implemented for bare clusters. Chapter 3 details the modifications made to the cluster genetic algorithm to accommodate ligated systems. Chapter 4 explains the importance of energy evaluation to the genetic algorithm and discusses the use of fast machine-learned interatomic potentials (MLIP), trained on high-quality *ab initio* structure-energy data, to replace the majority of first-principles calculations required by GA. Chapter 5 covers the active learning (AL) approach developed to stabilize GA against incorrect predictions made by the machine-learned interatomic potentials, and to allow for the continual improvement of these potentials in operation. Chapter 6 presents results pertaining to the optimization of the GA+MLIP+AL system, and to its application to search for the ground state of $Au_{18}(SR)_{14}$. Chapter 7 concludes with a discussion of these results and possibilities to be explored in future work.

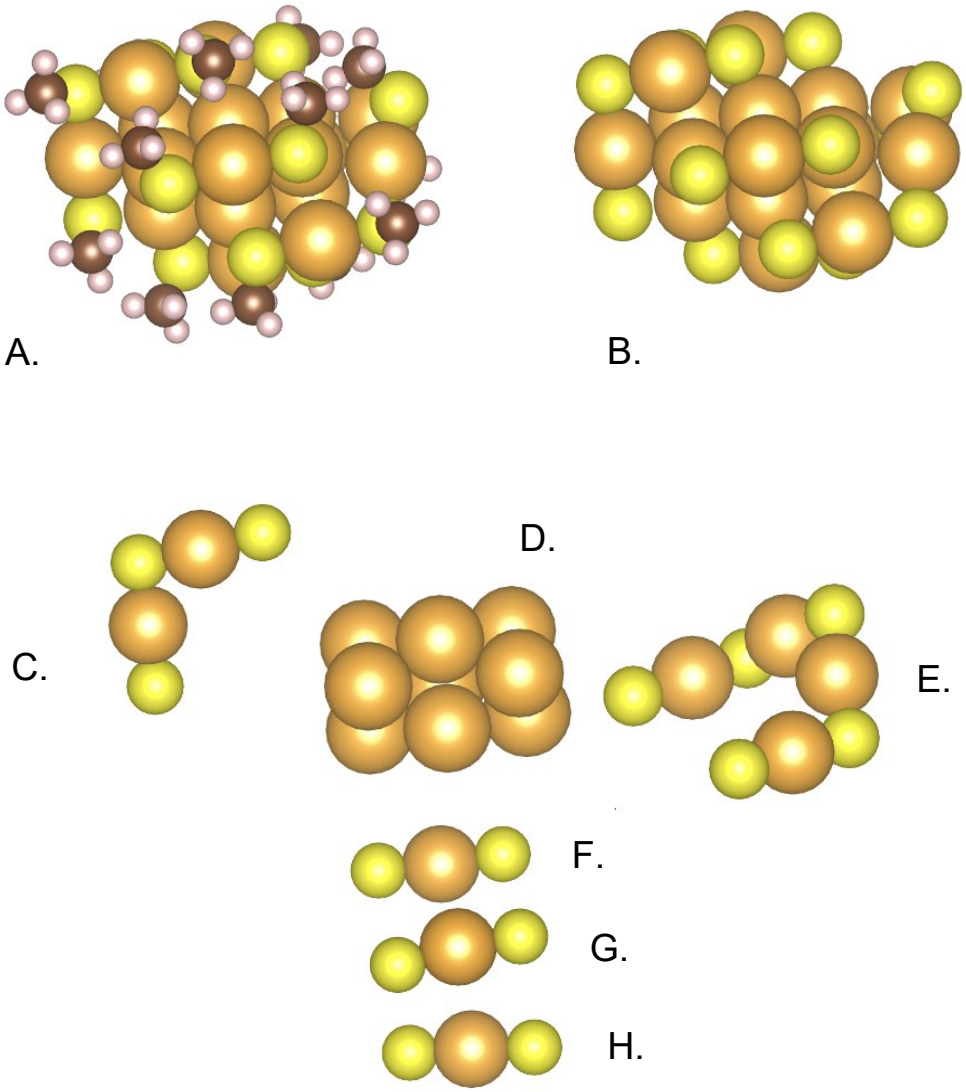

Figure 1.2: The ground state configuration of $Au_{18}(SR)_{14}$ and its component motifs.
A: the ground state structure, taken from the experimental characterization of [17] and relaxed using spin-polarized density functional theory as implemented in the Vienna *Ab initio* Simulation Package (VASP). The ground state energy of this structure was calculated to be -396.977 eV. **B:** the ground state structure shown without ligand side chains for clarity. **C:** $Au_2(SR)_3$ bridge. **D:** Au_9 *hcp*-stacked core [21]. **E:** $Au_4(SR)_5$ tetramer staple motif [22]. **F-G:** $Au(SR)_2$ staple motifs surrounding the Au_9 core.

Chapter 2. Genetic algorithms

2.1 The problem of global structural optimization

The lowest-energy atomic structure for a given nanocluster inevitably represents a subtle balance between competing electronic effects, and cannot therefore be determined a priori. First-principles quantum chemical methods are best suited to the task of determining the stability of a given configuration, and can be used for the local optimization of randomly sampled candidate structures to yield atomic configurations that locally minimize the potential energy. Naïvely, the absolute lowest-energy structure could be found by exhaustive generation and comparison of such locally optimized configurations. However, the number of minima in the potential energy landscape corresponding to locally optimal configurations grows exponentially with the number of atoms in the system, making this approach impracticable [23]. Efficient and intelligent methods for sampling configuration space are therefore needed to address the problem of global structural optimization. Different approaches to the problem of global optimization in systems exhibiting many local optima include simulated annealing [24], basin-hopping [25], particle swarm optimization [26], the artificial bee colony algorithm [27] and the focus of this work, genetic algorithms.

2.2 Genetic algorithms for structure discovery

Genetic algorithms attempt to find global solutions to high-dimensional optimization problems by iteratively recombining the features of distinct local solutions in a process that is intended to emulate evolution and natural selection [28]. The hypothesis underlying this approach is that the globally optimal solution should be comprised of portions (by analogy, the “good

genes") that are themselves optimal when considered independently, and that these optimal features will also manifest in some of the local minima. The extent to which this hypothesis applies will depend on the problem being considered. We would expect, for example, that a genetic algorithm will only be efficient in comparison to uniform sampling to the extent that the "goodness" of the solution being sought is reducible to the contributions of its parts.

Genetic algorithms are particularly applicable to the problem of identifying low-energy material structures because the stable arrangements of atoms in ordered materials are often divisible into recurring motifs. For instance, in the case of carbon nanostructures, pentagons and hexagons of sp_2-conjugated carbon are prominent in fullerenes across the size range [29]. Similarly, it is known that many gold nanoclusters adopt configurations featuring one or more Au_4 tetrahedra, and the independent stability of these tetrahedra has been used heuristically to help propose plausible structures for nanoclusters in the absence of more exact means of structural determination [30]. Genetic algorithms have been successfully applied to efficiently identify low-energy configurations of a wide range of nanoscale systems, including metallic [19] and nonmetal [31] clusters, amorphous materials [32], catalytic binding sites [33], biomolecules [34, 35], and metal-organic frameworks [36].

2.3 Overview of the pool-based genetic algorithm

In ongoing and as-yet unpublished work, our group has used a genetic algorithm to identify new structural candidates based on a running memory of the best configurations found so far. Our implementation is based on the pooled Birmingham cluster genetic algorithm (Pool-BCGA) [37], which improves upon generation-based genetic algorithms by allowing for the construction and evaluation of structural candidates to take place continuously and in parallel.

The algorithm begins by generating a number of random arrangements of atoms of the desired composition (referred to as **candidate** structures); subsequently, each random structure is **relaxed** by varying its atomic coordinates to bring its energy to a local minimum, and the final energies of all of the relaxed structures are ranked. The lowest-energy (i.e., most stable) candidates are stored in a database, called the **pool**, and a parameter called the **selectability** is calculated for each structure. Importantly, the pool is structured so that it can be accessed and updated by multiple instantiations of the genetic algorithm simultaneously, providing good speedup and scaling with parallelization.

After this random initialization, new candidate structures are produced by recombining fragments of two or more previous candidates in a process termed "**crossover**," or by randomly modifying a single previous candidate in a process called "**mutation**." After each new candidate is generated and relaxed, the ranking of energies is updated, and only the stablest candidates—specifically, those in the pool—are mutated or crossed over to produce new candidates. By this approach, the genetic algorithm is able to generate progressively better candidates until eventually, typically after some thousands of iterations, it may **converge** upon the global minimum structure.

This basic workflow is represented in **Figure 2.1** and is covered in more detail in the sections that follow.

2.4 Initialization

The first step in the genetic algorithm is to generate an initial population of candidate cluster structures. This can be done one of two ways: either by providing the algorithm with structure files to start with, or by using the algorithm's internal routines to randomly generate clusters to comprise the initial population. Our group's genetic algorithm has two distinct

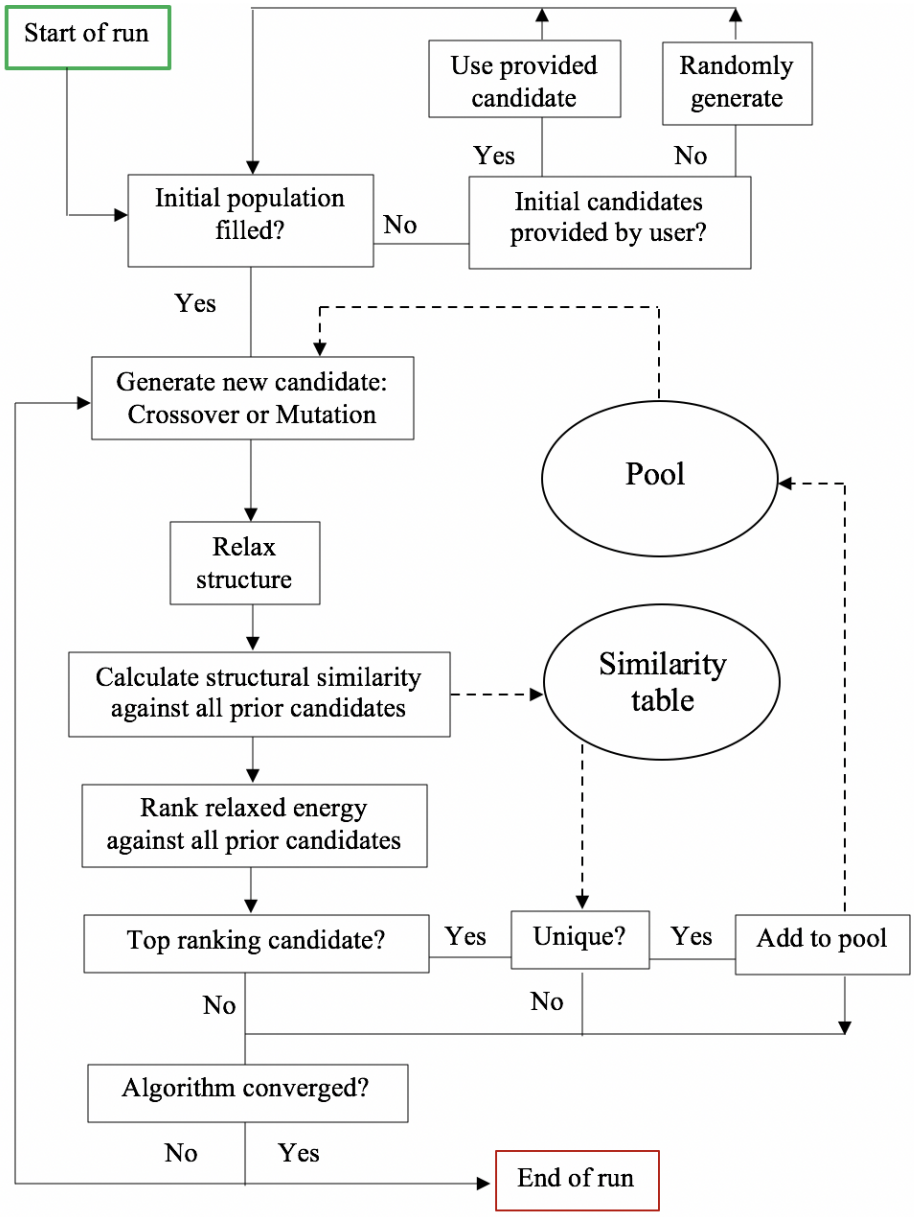

Figure 2.1: Basic workflow of the pool-based genetic algorithm.

methods for generating random clusters, described below.

Random generation by scattering: A bare cluster of the required number of atoms is generated by randomly placing atoms in a cubic box of side length $L = r_{ij}\sqrt[3]{N}$, where N is the number of atoms and r_{ij} is the nearest-neighbor distance. If there are multiple atomic species, the larger

nearest-neighbor distance is used to set the box length. Overlap is avoided by rejection sampling: that is, as each atom is randomly placed, it is checked for overlap with any other atoms and if there is overlap, a new placement is randomly chosen.

Random generation by concatenation: A single atom is placed at the origin, to which an atom is attached at the nearest-neighbor distance in a randomly chosen orientation. Atoms are placed in this manner, with every atom being attached to the one placed before it, until all atoms in the cluster have been placed. Overlap is avoided by rejection, as in the scattering method. This method has the advantage of ensuring that all atoms are in contact with at least two other atoms, whereas generation by scattering can lead to diffuse or disconnected clusters that require more ionic steps during relaxation to become close-packed.

2.5 Selection and recombination of candidates

At each iteration of the genetic algorithm, a new candidate structure is produced from the low-energy clusters in the pool. This takes place through either a mutation operation, whereby a single cluster in the pool is modified to produce a new candidate, or via crossover, where two pool clusters are chosen as "**parents**" and recombined to produce a candidate "**child**" cluster. For a mutation operation, the cluster to be mutated is chosen randomly and uniformly from the pool. By contrast, for crossover operations, the parents are selected via a roulette algorithm that favors clusters with higher assigned selectability (see Section 2.7 for details of the selectability calculation). The roulette algorithm compares the selectability of a randomly chosen pool cluster to a randomly generated number between zero and one, and the cluster is selected if its selectability is larger than the random number. This process is repeated until two parent clusters have been selected. Although in our current implementation only two-parent crossover is supported, in

principle, this selection algorithm can be used as written to identify multiple parents for multi-way crossover operations.

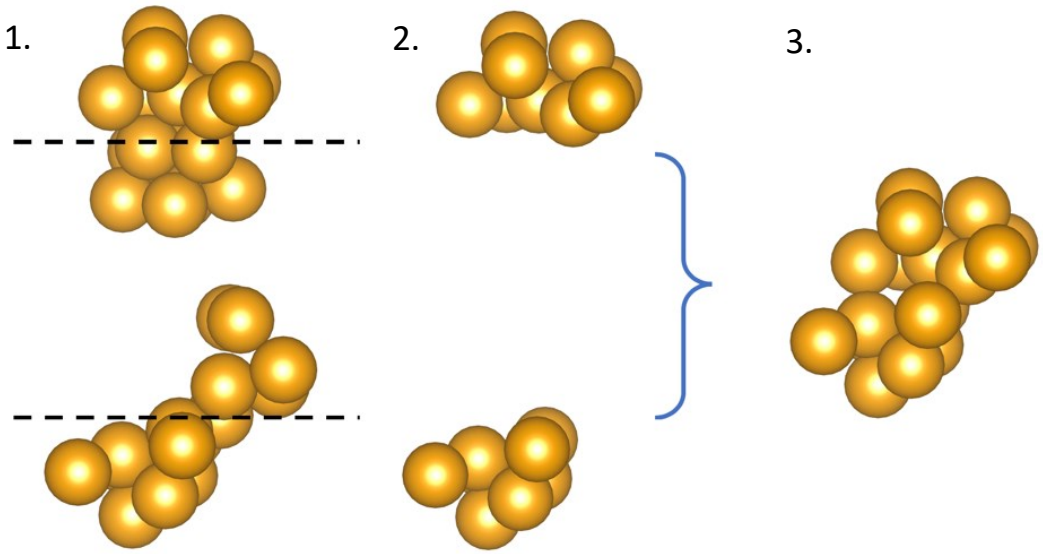

Figure 2.2: Cut-and-splice crossover operation. Two parent clusters are chosen, and a dividing plane is drawn through each structure (1) to produce complementary segments (2). These segments are merged to produce a new, unique child cluster (3).

2.6 Genetic operations: crossover and mutation

The primary determinant of a genetic algorithm's success resides in its method for recombining previous solutions to produce new candidate solutions—that is, how the algorithm generates children from parents. In particular, the algorithm must ensure that significant properties of the parents are passed down to the children [38]. In the first demonstrations of GA for the discovery of low-energy cluster structures by Xiao and Williams [39] and Hartke [40], clusters were encoded as binary strings, and new clusters were generated by concatenating substrings of

low-energy parent clusters. Later, Deaven and Ho [38] proposed a more physically meaningful crossover method known as the *cut-and-splice* crossover operation, which we apply in this work. The cut-and-splice operation involves merging segments (often halves) of parent structures in real space to create a new structure, as illustrated for a pair of bare clusters in **Figure 2.2**. Cut-and-splice crossover implementations may differ with respect to how the sections of each parent are chosen and merged, e.g. in the choice and consistency of the dividing plane, in whether the orientation of the parent segments is preserved through the operation, or in whether the parent segments are of equal size.

Since crossover operations attempt to convey the structural properties of the parent structures to the child structure, they have only a limited ability to generate new structural features. This can result in a lack of diversity in the population of candidates, impeding the efficiency of the genetic algorithm. Accordingly, methods for introducing randomness into the structure generation process are often employed to improve the genetic algorithm's efficiency [10]. Such methods are known as *mutation* operations. Mutation operations involve modifying a randomly selected subset of a cluster's structure to produce a new candidate. Following Johnston [10], we use the term "static mutation" to refer to operations where the subset being modified is assigned a random value (i.e., randomized), and "dynamic mutation" for operations that change the subset to a value dependent on its initial state.

Two mutation methods are used in our genetic algorithm for bare clusters. The first, **Rotate**, is a dynamic mutation that rotates a portion of the cluster by a random angle from its initial position. The second, **Move**, is a static mutation that randomly selects atoms from the cluster and shifts each atom's position by a random vector of a small magnitude (up to the nearest-neighbor distance).

2.7 Convergence

While genetic algorithms have been successfully applied to the problem of ground-state structure identification by our group and many others, we must mind the fact that identification of the global minimum is not mathematically guaranteed. Rather, genetic algorithms belong to a class of optimization strategies known as heuristics: methods that "steer towards" iteratively better solutions to a problem in a non-exhaustive, but expedient manner [41]. Accordingly, a genetic algorithm must be equipped with a reasonable criterion by which to gauge convergence and to call off the search. In the context of structural search, this criterion is sometimes that a number of iterations have gone by without the discovery of any new low-energy configurations [37], or simply that a pre-set number of candidates have been evaluated [36]. In our group's implementation, convergence is declared when all of the top-ranked structures have been extensively operated upon (i.e., crossed over or mutated) and no new top-ranking candidates have emerged. Mathematically, this condition is expressed as follows. For each candidate in the pool, we calculate a selectability S which monotonically decreases with respect to both the candidate's energy E and the number of times N_C that it has previously been chosen for crossover:

$$S_i = \left[\frac{1 - \tanh(2R_i - 1)}{2}\right]\left[\frac{1}{1 + \sqrt{N_{C,i}}}\right]$$

The term R_i above expresses the normalized relative energy of configuration i with respect to the rest of the pool, varying from 0 when i has the lowest energy in the pool to 1 when it has the highest:

$$R_i = \frac{E_i - E_{min,pool}}{E_{max,pool} - E_{min,pool}}$$

The first term in brackets in the equation for S_i is called the *fitness* for candidate i, F_i, and the second bracketed term is known as the *regulated frequency* of selection, ω_i. We aim to evolve stable structures, so candidates with lower energy are assigned higher fitness. Meanwhile, the frequency of selection is decreased (or "regulated down") for candidates that have been selected many times in the past. In total, the selectability for each candidate in the pool is the product of the candidate's fitness and its regulated frequency:

$$S_i = \omega_i F_i$$

$$\omega_i = \frac{1}{1 + \sqrt{N_{C,i}}}$$

$$F_i = \frac{1 - \tanh(2R_i - 1)}{2}, \quad R_i = \frac{E_i - E_{min}}{E_{max} - E_{min}}$$

Iteration of the genetic algorithm continues until the maximum difference in selectability among any two candidates in the pool is less than a user-set threshold, δS. If a candidate is very stable relative to the rest of the pool, yet also very frequently selected, then its selectability will be low. Conversely, if it is less stable but has rarely been selected, then its selectability is increased. This arrangement ensures that the possible permutations of the pool are widely explored: when the pool is initiated, all candidates are new, and so more stable candidates are recombined more frequently, but as the algorithm iterates, less-fit candidates in the pool will be chosen for reproduction with increasing frequency. When a new candidate enters a pool that has stayed stable for several hundred iterations, it will be chosen frequently regardless of its stability. A convergence profile of the variation of δS in a typical GA run is shown in **Figure 2.3** below.

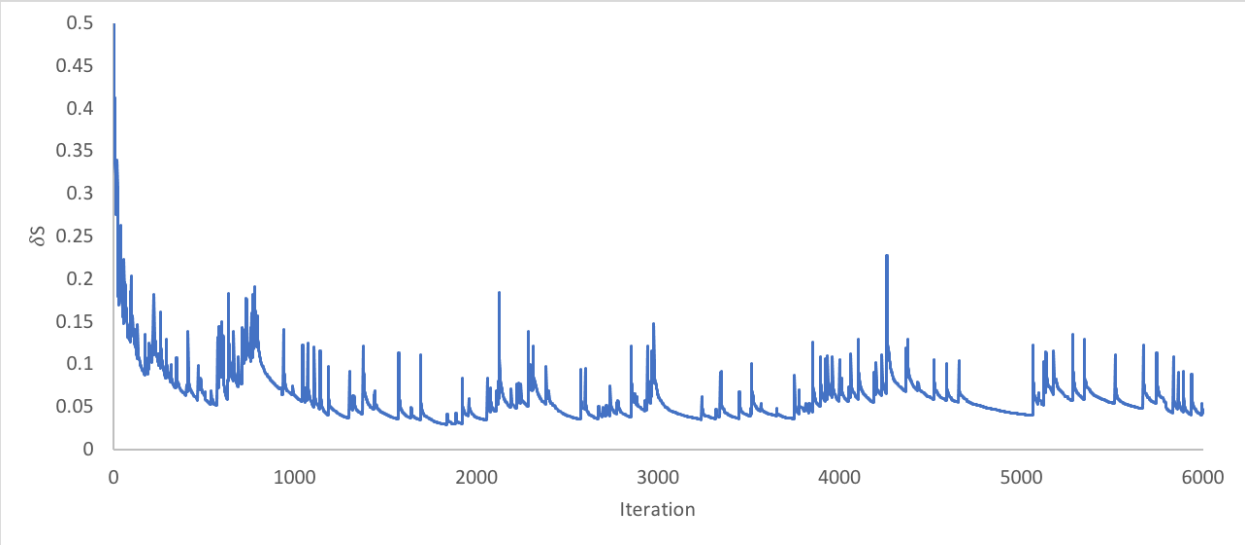

Figure 2.3: Convergence profile for a typical GA run. The difference in selectability across the pool candidates, δS, is plotted as a function of the iteration. The value of δS declines while the pool is unchanged, spiking when new candidates enter the pool.

2.8 Identifying similar candidate structures

As the genetic algorithm explores configuration space, it is possible that it will repeatedly visit the same local minima, discovering near-identical structures multiple times. In this circumstance, the pool could become populated with copies of the same cluster, thereby losing its structural diversity. To prevent this from happening, the genetic algorithm needs a method to recognize similar candidates.

Our approach is as follows. For each new candidate, a difference score is calculated against all previous candidates using the eigen-subspace representation of Li, Yang, and Zhao [42]. In overview, six basic steps are involved in the construction of this representation:

1. First, a distance matrix $\boldsymbol{D}$ is constructed for each structure. In the diagonal elements of the matrix, we enter the atomic number (or another element-identifying value) of each atom in the structure. Off-diagonal elements correspond to Cartesian distances between the atoms.
2. The distance matrix is decomposed into its eigenvalues λ_k and eigenvectors $\boldsymbol{u}_k$:

$$\boldsymbol{D} = \sum_{k=1}^{n} \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T$$

3. Next, the complete set of eigenvectors for each eigenvalue, i.e. the eigen-subspace for each eigenvalue, is used to construct an eigen-subspace projection array (EPA) $\boldsymbol{s}_i$ for each atom i in the structure. Each entry in this array, $s_i^{\lambda_k}$, is the norm of orthogonal projection of atom i over the complete set of eigenvectors m associated with the eigenvalue λ_k. By using the complete set of eigenvectors for each eigenvalue, we ensure that we uniquely specify the atomic coordinates.

$$\boldsymbol{EPA(i)} \equiv \boldsymbol{s}_i = \{ s_i^{\lambda_1}, s_i^{\lambda_2}, \dots , s_i^{\lambda_n} \}$$

$$s^{\lambda_k} = \sqrt{\sum_m \left(r_m^{\lambda_k} \right)^2}$$

4. These EPAs are rendered as eigen-subspace projection functions (EPFs) Λ_i, in which each atom is represented as a unit vector in the eigen-space $S \in [0, 1]$ and its components associated with each eigenvalue λ_k are grouped together piecewise, in order of increasing λ_k, as shown in **Figure 2.4**.
5. Next, the "EPF distance" between two atoms is calculated as the absolute difference between their EPFs integrated across the eigen-space:

$$d_{ij}^{EPF} = \int_0^1 \left| \Lambda_i - \Lambda_j \right| dS$$

6. Finally, the difference score between two structures is given as the sum of EPF distances between their atoms, minimized over possible one-to-one correspondences of atoms. This minimization between pairs of atoms can be thought of as a problem of optimal assignment and is carried out by the Hungarian algorithm [43].

If two structures have a difference score lower than a threshold value (internally set in our implementation to 0.5), they are deemed to be similar. If a newly generated candidate is similar to a candidate structure already in the pool, the energy of the new candidate is compared with the pool candidate. The new candidate takes the place of the candidate it resembles in the pool if it has a lower energy.

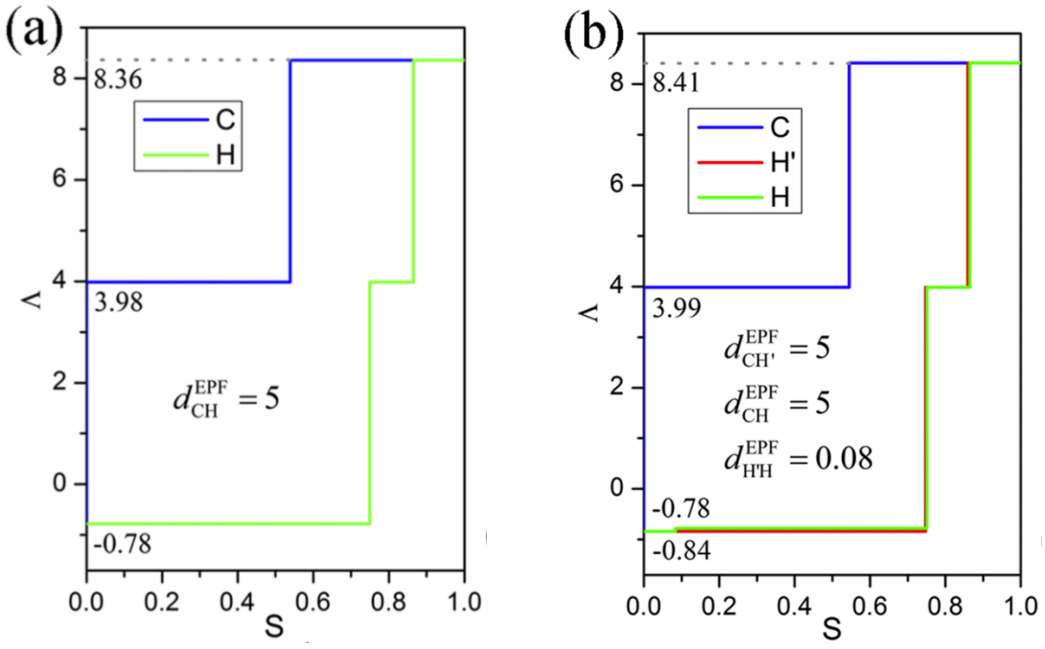

Figure 2.4: Eigenvalue projection functions. Reproduced from [42], with the permission of AIP Publishing. This plot shows the eigenvalue projection functions for the carbon and hydrogen atoms in a standard methane molecule (a) and methane with one C-H bond lengthened by 0.05 Å, (b). The eigenvalues of the distance matrix are at 8.36, 3.98 and -0.78 in (a) and 8.41, 3.99, -0.78 and -0.84 in (b). Though new eigenvalues and eigenvectors emerge as a result of the stretched bond, this minute structural change is represented as a correspondingly small change in the EPF, demonstrating the utility of the EPF as a means of structure comparison.

Chapter 3. Adapting GA for ligated clusters

3.1 Structural considerations

As discussed in Chapter 1, for most practical applications, nanoclusters must be protected from aggregation by the attachment of surface ligands. These ligands do not simply passivate the cluster's surface, but themselves actively influence the structure and properties of the ligated cluster. In order to make predictions for most nanocluster systems of practical interest, therefore, we are tasked with identifying the ground state structure of the core cluster together with all terminating ligands: the cluster and ligands cannot be considered separately. This is a more challenging problem than the bare cluster case for several reasons.

First, the number of atoms that must be considered when calculating the energy and when performing local optimizations is substantially increased by the addition of ligands, which makes individual iterations of the genetic algorithm take longer for a given number of core cluster atoms. For energy evaluation, if density functional theory is used—representing the least expensive *ab initio* method of appropriate accuracy—the computing time scales with between the square and the cube of the number of symmetrically distinct electrons in the system [44].

Secondly, the geometry of ligated systems is more complex than the bare cluster case. Each ligand introduces multiple geometric degrees of freedom beyond the usual single positional degree of freedom of the core atoms. This adds another layer of difficulty to local structural optimizations, but more importantly, it means that the configurational space that the genetic algorithm can sample is much larger, and identification of the global minimum becomes a correspondingly more daunting exercise.

Finally, ligated clusters impose special constraints on the genetic algorithm's structure generation methods. Genetic operations used for bare clusters are generally not applicable for ligated clusters without modification. Ligands must be treated as distinct entities from the core atoms to maintain stoichiometry and to avoid sampling unlikely configurations where, for example, side chains are separated from their binding moieties (see **Figure 3.1**). At the same time, we should seek to constrain the genetic operations as little as possible, since there is a risk of unintentionally biasing the genetic algorithm away from conformations that may be worth exploring.

The sections in this chapter primarily deal with this last challenge. Herein, we describe the modifications made to our genetic algorithm's structure-handling methods to enable the study of ligated clusters.

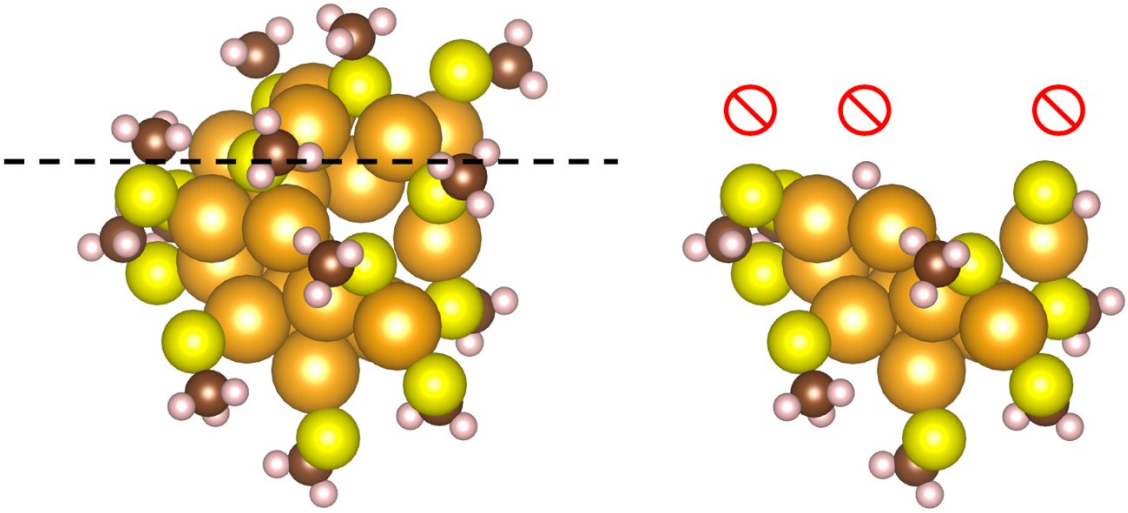

Figure 3.1: Failure of cut-and-splice crossover for ligated clusters. The cut-and-splice method must be modified for ligated clusters, as randomly-chosen cutting planes will often pass through ligands, resulting in segments with cleaved molecular bonds.

3.2 Initialization

To start the genetic algorithm, an initial population of structural candidates is required. In our implementation, this initial population can be randomly generated or, alternatively, taken from files provided by the user. For the case of ligated clusters, we construct the initial population by attaching ligands to bare clusters, which can similarly either be randomly generated or user-provided. If the user provides an initial population of bare clusters, these will subsequently be randomly ligated with the necessary number of ligands. If the initial pool is to be randomly generated, the same procedure applies: bare clusters are randomly generated (see Section 2.4) to which ligands are randomly attached.

Our process for ligand attachment is designed to assure that ligands are placed on the outside of the cluster without overlapping. This is accomplished by the following procedure, illustrated in **Figure 3.2**. Starting from a bare cluster that has been randomly generated or supplied by the user, we choose a random point on a sphere centered at the center of gravity of the bare cluster and position a ligand so that its center of gravity is on this chosen point. Next, we find the atom in the cluster that is closest to the positioned ligand's center of gravity, and the atom in the positioned ligand that is closest to this nearest cluster atom. The ligand is moved towards the cluster along the vector between these two atoms until the ligand just contacts the cluster. This "orbiting-and-landing" approach is repeated until all requested ligands are placed, with two conditions checked at each iteration: first, if the cluster atom nearest the ligand's original position already has a ligand attached to it, the next-nearest unoccupied cluster atom is chosen as the ligand's destination; second, ligands are checked for overlap in their final position. The placement is rejected if overlap is detected and a new trial placement is begun. Optionally, we can require

that ligand placements are only valid if a particular atom in the ligand is in contact with the cluster; this allows for the orientation of ligands to be specified if the active site of the ligand is known.

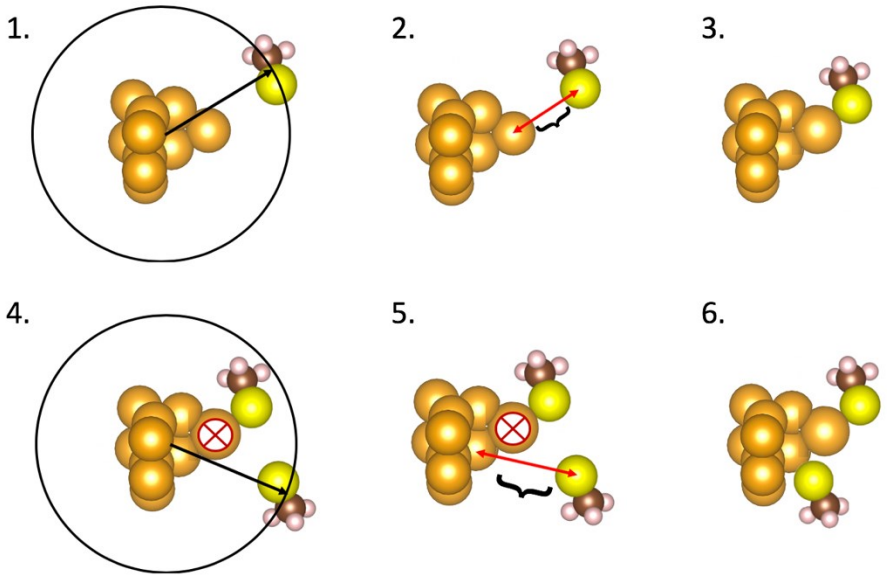

Figure 3.2: Random ligation of bare nanoclusters. Starting from a bare cluster structure, a random point on a sphere centered at the cluster's center of mass is chosen as the location of a randomly-oriented ligand (1). The closest pair of ligand and cluster atoms is identified, and the clearance between the atoms calculated (2). The ligand is then moved into contact with surface of the cluster (3). Next (4), a new ligand is positioned as in (1), but only unoccupied cluster atoms are valid for placement. The clearance between the ligand and the nearest unoccupied cluster atom is determined (5), and the ligand is put into contact with this unoccupied atom.

3.3 Crossover

For the cut-and-splice crossover of ligated clusters, two parent clusters are selected from the pool using the roulette method (Section 2.5) and a ratio is chosen for their mating. The **ratio of parenthood** depends on the mode of crossover; three modes are available to be activated by the user. **Even** crossover takes 50% of the core atoms and ligands from each parent cluster to produce the child cluster. **Random** crossover chooses the percentage of each parent randomly and

uniformly. **Weighted** crossover determines the number of atoms (and ligands) n_i contributed by each parent cluster i based on the ratio of fitnesses F_i of the parents:

$$n_1 = n_{total}\left(\frac{F_1}{F_1 + F_2}\right), n_2 = n_{total}\left(\frac{F_2}{F_1 + F_2}\right)$$

Where n_{total} is the number of core atoms and/or ligands specified by the stoichiometry of the system being studied. The equation for the fitness F_i is given in Section 2.7.

Once two parent clusters are selected from the pool and the ratio of their parentage is established, each parent is subdivided into its ligands and its core atoms. The core atoms are then indexed by their ascending Cartesian z-coordinates (or, in the case of the ligands, the z-coordinates of their centers of mass) to provide a standard ordering. Next, core atoms are taken one by one in order of increasing index from the first parent, up to a total number of atoms n_1, i.e. from index 0 to index $(n_1 - 1)$; the remaining n_2 atoms are taken from the second parent from index n_1 to index $(n_1 - 1 + n_2)$. The segments of the core atoms are merged, and overlapping atoms are moved away from each other until overlap is corrected. Concurrently, the same process is conducted for the ligands. This process is illustrated in **Figure 3.3**. Overlap between ligands and core atoms, as well as between ligands and other ligands, is corrected by moving overlapping ligands and atoms away from each other until overlap is no longer observed. Throughout the overlap correction process, ligands are moved as groups of atoms; care is taken to never alter the relative position of atoms within individual ligands.

3.4 Mutation

The Rotate method described in Section 2.6 was found to be inapplicable to ligated clusters because a random rotation applied to a segment of the target cluster could often result in

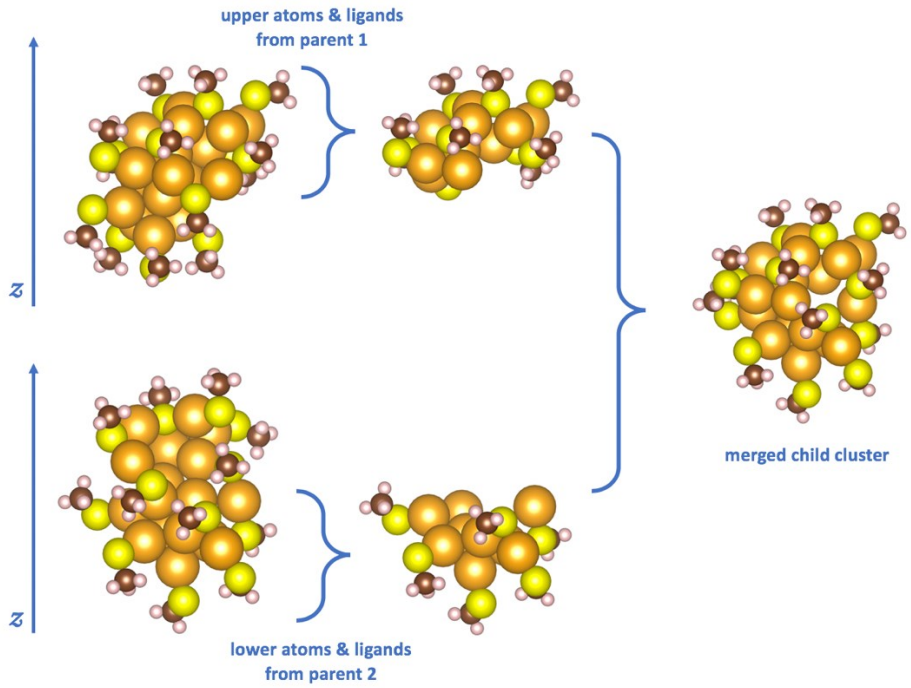

Figure 3.3: Crossover of ligated clusters. Parent clusters are chosen, and their atoms and ligands are sorted along a single Cartesian direction. Atoms and ligands are chosen in order of this ranking to build up segments, which are then merged to form the child cluster.

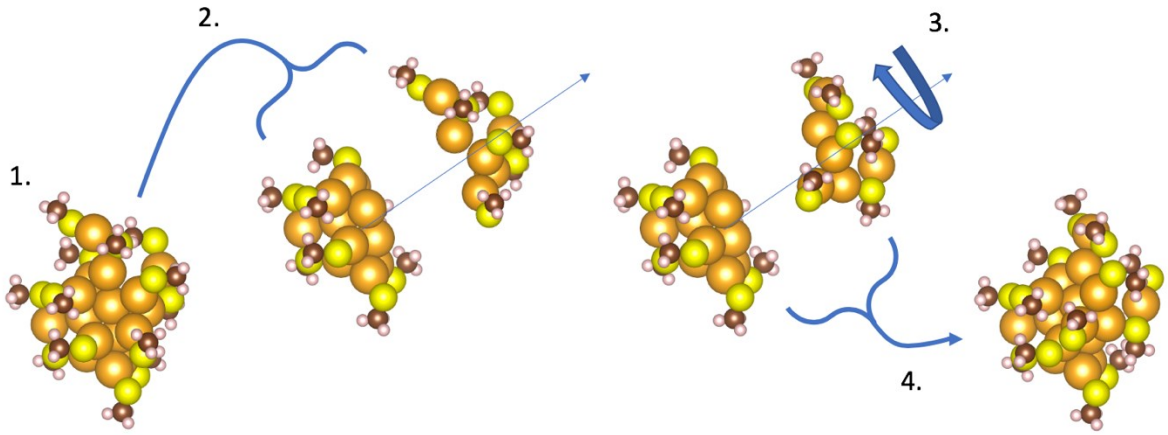

Figure 3.4: Mutation of ligated clusters. First, (1) a ligated cluster is chosen from the pool for mutation. A random axis passing through the cluster's center of mass is chosen and the cluster is subdivided along this axis (2). The smaller segment of the cluster is rotated around the chosen axis by a random angle (3), and the segments are merged (4) to form the mutated cluster.

surface-attached ligands being rotated inwards into the body of the cluster. In response, a new Rotate method was developed for ligated clusters that preserves the outward orientation of ligands, illustrated in **Figure 3.4**. In this method, we choose a random axis passing through the center of gravity of the ligated cluster and rotate a segment of the cluster around this axis by a random angle.

3.5 Similarity evaluation

Due to the greater structural complexity of ligated systems, more iterations of the genetic algorithm are required to discover low-energy configurations than for bare clusters. Our similarity evaluation method, which calculates the similarity of each new candidate to every previous candidate, can become a considerable bottleneck in the algorithm when handling systems with many atoms over many iterations. To reduce the time spent on similarity evaluation, we pass a reduced representation of each candidate to the similarity calculator that only includes the core atoms and the active sites of the ligands, ignoring the ligand side chains. The speedup compared to similarity evaluations using the entire ligated structure is shown for $Au_{18}(SCH_3)_{14}$ running on 24 CPUs in **Figure 3.5**. At 500 iterations, the similarity evaluation of each new candidate takes nearly 100 seconds if the side chains are included, whereas the time per evaluation is below 2 seconds at the same iteration when considering only the core atoms and active sites.

In addition to affording an improvement in throughput, this way of evaluating similarity between ligated structures is arguably more appropriate in principle, since for a particular ligated cluster the ligand side chains may be able to take on many degenerate, energetically equivalent sets of conformations, especially in the case of small and/or flexible ligands. If identical clusters differing only in the position of the ligand side chains are recognized as different structures, the structural diversity in the pool could suffer as a result.

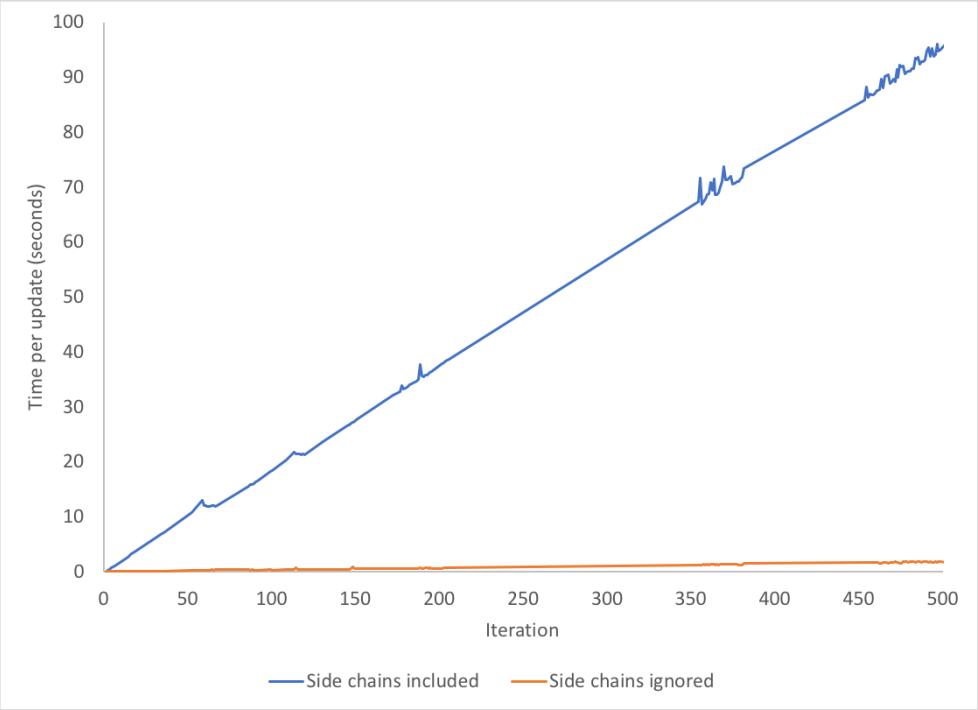

Figure 3.5: Time required for similarity evaluation of new candidates.

Chapter 4. Energy evaluation by machine learning

4.1 The importance of energy evaluation in GA

Energy evaluation is the critical step in a structural search, whether the search is conducted by a genetic algorithm or an alternative global optimization method, and it is also generally the rate-limiting step. The primary task of searching for low-energy cluster structures comes down to the accurate comparison of the energy of different conformations. This task is doubly complicated in the case of ligated systems, as with the addition of ligands we must account for more atoms and element types that increase the computational cost, while the potential energy surface can be quite flat due to the flexibility of the ligands, requiring us to resolve small differences in energy between conformers. The energy of a cluster conformation can be most accurately evaluated with first principles quantum mechanical techniques, though we are limited to the simpler methods among these due to the large number of atoms being considered; density functional theory is applicable whereas, for example, wavefunction methods are not [11].

4.2 Density functional theory

Density functional theory (DFT) is the name given to a family of computational methods that efficiently solve for the electronic ground state of a system of atoms from first principles. DFT is a formally exact means to solve the Schrödinger equation for the lowest-energy state of a system with n electrons by taking advantage of the one-to-one mapping of electron densities to ground states [45]. In a DFT calculation, the Hamiltonian of the system being evaluated is formulated in terms of the 3-dimensional average electron density rather than the $3n$-dimensional many-body

wavefunction, which much simplifies computation. The ground state is discovered by minimizing the energy of the system with respect to the variation of the electron density.

4.3 Machine-learned interatomic potentials

While first-principles quantum mechanical calculations provide both reliability and accuracy, their computational cost becomes prohibitive for applications that require large systems (>100s of atoms) to be evaluated many times (>1000s of iterations). If we are willing to sacrifice the general applicability afforded by methods derived from physical principles, comparable accuracy can be obtained with greatly reduced computational expense by fitting parameterizable interatomic potentials to quantum mechanical data [46]. This approach falls within the paradigm of supervised machine learning [47]: we aim to learn a good approximation to the energy-structure hypersurface from a training set of structures and their DFT-calculated energies, forces, and stresses.

Once trained, the quality of a machine-learned potential can be judged in a variety of ways. Three figures of merit of particular relevance to GA are the **root-mean-square error** (RMSE) of the energy, the **average of errors** (or the **bias**), and the Spearman **rank-order correlation**. All of these values should be assessed as prediction errors, not as training errors; that is, the trained model should be tested with configurations it has never encountered before. The RMSE simply indicates how far on average the energy predicted by the trained potential will deviate from the DFT-calculated energy. The average of errors is ideally zero, and its value reveals any tendency the trained potential might have to systematically over- or underestimate the energy compared to *ab initio* methods. **Figure 4.1** shows parity plots of potentials exhibiting large magnitudes of RMSE and average error, respectively.

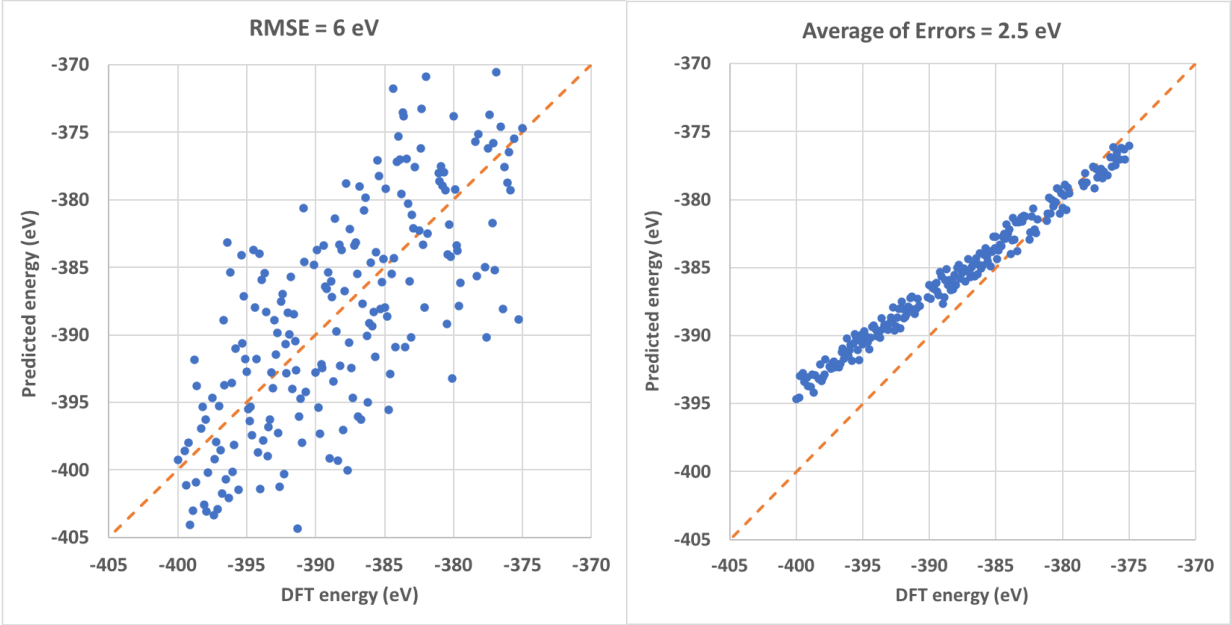

Figure 4.1: Different kinds of prediction error for machine-learned potentials. Left: a representative parity plot for a potential exhibiting high root-mean-square energy error across a test set. Right: a representative parity plot for a potential exhibiting a high average of errors on the test set, systematically overestimating the energy of low-energy configurations. In both plots, the orange line represents ideal agreement (perfect parity) between the model and direct quantum mechanical calculations.

The Spearman rank-order correlation, **Figure 4.2**, compares the ordering of configurations according to their DFT energies with their ordering according to their energies as predicted by the machine-learned potential. This correlation is of interest because the performance of a potential model for GA hinges on its ability to place structures in the pool in agreement with DFT; its accuracy in energy prediction mainly matters as a proxy for its competence in this ranking task. The formula for the Spearman rank-order correlation is:

$$r_S = 1 - \frac{6\sum_i^N \left(R_{ML,i} - R_{DFT,i}\right)^2}{N(N^2-1)}$$

Where N is the number of configurations being ordered, $R_{ML,i}$ is the rank of the i'th configuration according to the learned model, and $R_{DFT,i}$ is the rank of the i'th configuration according to DFT.

The value of r_S ranges from +1 in the case of perfect ranking to -1 in the case of exactly inverse ranking.

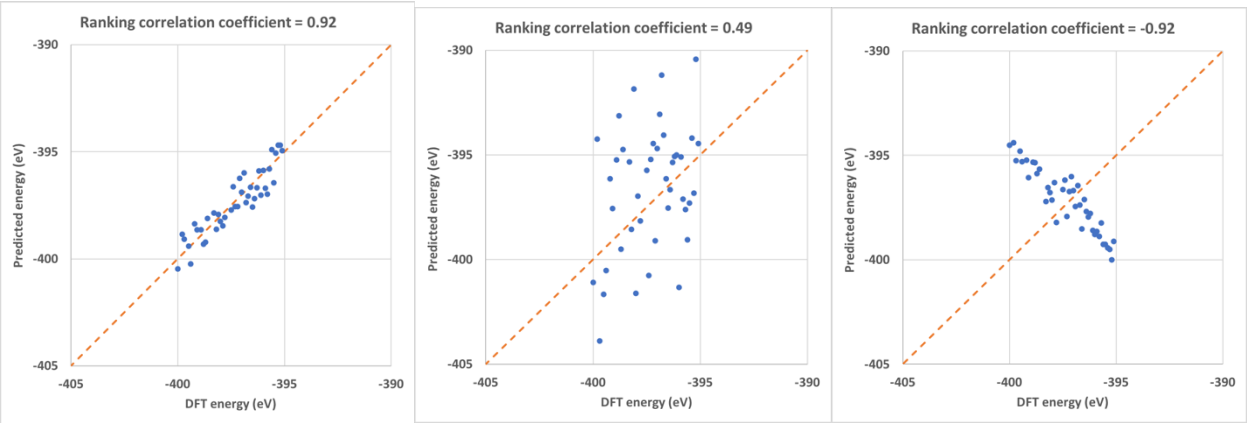

Figure 4.2: The Spearman rank correlation coefficient. Left: a potential that ranks a test set in good agreement with DFT, yielding a high positive rank-order correlation of 0.92. Center: a potential with greater variance cannot order the test set as accurately, resulting in a lower correlation of 0.49. Right: a potential that gets the trend in energy wrong, yielding a negative ranking correlation on the test set of -0.92, corresponding to a negative monotonic relationship between the predicted energy and the DFT energy.

4.4 Moment tensor potentials

Moment tensor potentials are a class of machine-learned interatomic potentials introduced by the Shapeev group in 2016 [48]. The content of the following mathematical description is abridged from their recent paper on the MLIP package that implements these potentials [49].

Moment tensor potentials learn to predict the energy associated with an atom i and its local environment, or **neighborhood**, N_i, and give the energy of a configuration cfg of multiple atoms as the sum of the energies of all neighborhoods:

$$E^{MTP}(cfg) = \sum_{i=1}^{n} V(N_i)$$

The energetic contributions V are defined by a linear combination of basis functions, B_α, weighted by learned parameters ε_α:

$$V(N_i) = \sum_\alpha \varepsilon_\alpha B_\alpha(N_i)$$

The basis functions B_α are a set of **moment tensor descriptors** (or **moments**) $M_{\mu,\upsilon}$ and tensor contractions thereof (i.e. scalar products, vectorial dot products, and Frobenius matrix inner products between $M_{\mu,\upsilon}$). These descriptors are comprised of a radial component, which is a linear combination of polynomials defined within a cutoff radius R_{max}, multiplied by an angular component, which is a repeated outer product between interatomic distances, summed over all atoms within the neighborhood N_i:

$$M_{\mu,\upsilon}(N_i) = \sum_j \left[\sum_{\beta=1}^{N_Q} C^\beta_{\mu,t_i,t_j} Q^\beta\left(\left|r_{ij}\right|\right) \right] \underbrace{r_{ij} \otimes \ldots \otimes r_{ij}}_{\upsilon \text{ times}}$$

Where $Q^\beta\left(\left|r_{ij}\right|\right)$ are the radial basis functions, which take the form of Chebyshev polynomials of order β multiplied by a smoothing term so as to approach zero at the cutoff radius $\left|r_{ij}\right| = R_{max}$. The radial weights C^β_{μ,t_i,t_j} are learned parameters of the model, analogous to ε_α. The repeated outer product between interatomic distances, $r_{ij} \otimes \ldots \otimes r_{ij}$, is a tensor of rank υ introduced to encode angular information about the atomic neighborhood. **Figure 4.3** depicts the transformation of local atomic environments to moment tensor descriptors.

The size of the basis set B_α is defined by the **level** of the potential. The level for a single moment tensor descriptor is defined as:

$$\text{level}(M_{\mu,\upsilon}) = 2 + 4\mu + \upsilon$$

The constant coefficients 2, 4 and 1 were empirically tested and chosen as optimal by Shapeev et al. The level of a contraction of multiple moment tensor descriptors is defined as the sum of the moments being contracted, e.g.:

$$\text{level}\left(M_{1,0} \cdot M_{0,1}\right) = \text{level}\left(M_{1,0}\right) + \text{level}\left(M_{0,1}\right) = 9$$

$$\text{level}\left(M_{2,3}^{4}\right) = 4 * \text{level}\left(M_{2,3}\right) = 52$$

Each function B_α in the basis set is a moment or a contraction of moments. The size of the basis set of the potential is set by specifying a maximum level, where the basis set will be comprised of all moments and contractions of moments with $\text{level}(B_\alpha) < \text{level}_{\max}$. We also specify N_Q to define the size of the radial polynomial basis set for each moment. These two values, N_Q and $\text{level}_{\max}$, define the functional form of a particular moment tensor potential, and also determine the number of coefficients ε_α and C^β that must be set during training.

This apparently elaborate construction carries with it important advantages. The moment tensor descriptors are, by design, invariant to the permutation of atoms of the same species, as well as to transformations such as rotation and reflection. Each moment $M_{\mu,\upsilon}(N_i)$ is a two-body descriptor, having the notable quality of encoding angular information for rank $\upsilon \geq 1$ without incorporating three-body descriptions of the environment. On the other hand, moment tensor potentials are capable of representing arbitrarily many-body interactions through contractions of moments; by increasing $\text{level}_{\max}$ and thereby the varieties of moment contractions included in the basis set, progressively higher-order interactions can be captured. A moment tensor potential of a given level thus exists within a well-defined hierarchy of quality, where greater accuracy can be achieved at the tradeoff of slower computation and the requirement of larger amounts of training data. In this sense, moment tensor potentials are a systematically improvable class of models [48].

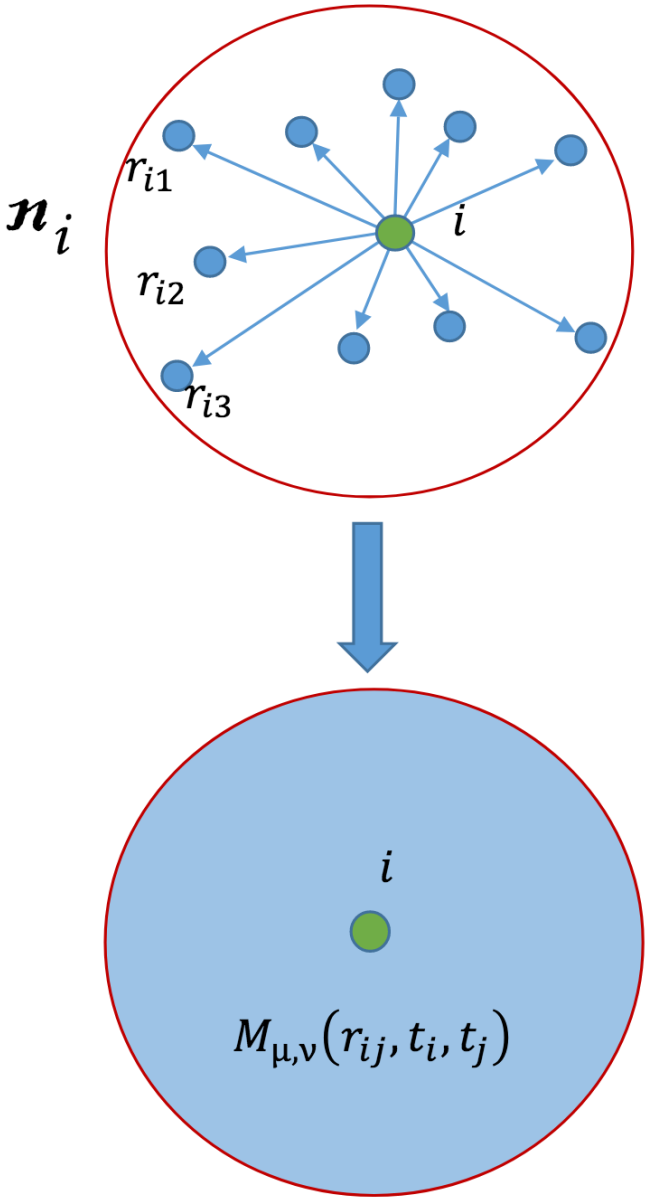

Figure 4.3: Moment tensor representation of local atomic environments. Reproduced from [50]. Moment tensors are descriptors of the neighborhood N_i of atom i, encoding information on the relative distances and angles between atom i and all neighboring atoms within the cutoff radius, as well as the atomic types t_i.

Despite these useful qualities, moment tensor potentials remain subject to some of the usual limitations of machine-learned models. Namely, their accuracy cannot exceed that of the method used to produce the training data; they typically struggle to represent systems unlike those that they were trained on; and they can be overfit to the training data, learning as signal what is actually noise. Also, although the polynomial basis chosen for the radial components helps ensure that the parameterized potential energy surface varies smoothly, it provides no guarantee on the model's performance outside the parameterized region. In practice, this means that a moment tensor potential can produce qualitatively incorrect results when tasked with evaluating or relaxing a configuration that is, in some sense, "too far beyond" what it was trained on. Here it is useful to introduce the notion of **interpolation** and **extrapolation**: moment tensor potentials and other machine-learned local potential models tend to have acceptable accuracy only within the region of configuration space spanned by their training set (i.e., when interpolating), and are unreliable when asked to make predictions outside of (i.e., extrapolating from) this region, as illustrated in **Figure 4.4**. This concept is formalized in MLIP, the package that implements moment tensor potentials, by way of the D-optimality criterion.

4.5 The D-optimality criterion

The following summary is adapted from [50]. A trained potential function P learns to approximate the energies, forces, stresses or other features y from associated configuration data x. The predicted features y' depend on the configurations x as well as the variable learned parameters of the potential, θ:

$$y' = P(\theta, x)$$

When training on a set of paired feature-configuration data $\{y_j, x_j\}$, we aim to minimize a loss functional L by varying the parameters θ:

$$L(\theta) = \sum_{j=1}^{n} \big(y_j - P(\theta, x_j)\big)^2$$

The minimization of this loss functional for a particular training set results in a set of trained parameters, $\bar{\theta} = \arg\min L(\theta)$.

For ligated nanoclusters, we must employ multi-component moment tensor potentials. Multi-component moment tensor potentials have a nonlinear dependence on the parameters θ. As long as the trained parameters of the potential are near their optimal values, however, we can approximate the potential as varying linearly with respect to its parameters θ:

$$P\big(\theta, x_j\big) \approx \sum_i (\theta_i - \bar{\theta}_i) \frac{\partial P(\bar{\theta}, x_j)}{\partial \theta_i}$$

Then, the terms within the summation of the loss functional are:

$$y_j - P\big(\theta, x_j\big)$$

$$\approx y_j - \sum_i (\theta_i - \bar{\theta}_i) \frac{\partial P(\bar{\theta}, x_j)}{\partial \theta_i}$$

Once linearized in this way, training of the potential can be expressed as the problem of finding error-minimizing solutions to an overdetermined set of n linear equations, where n is the number of configuration-energy pairs in the training set, with respect to the $m < n$ parameters θ_i. We can express this system of equations as a tall $n \times m$ matrix, B:

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial P(\bar{\theta}, x_1)}{\partial \theta_1} & \cdots & \frac{\partial P(\bar{\theta}, x_1)}{\partial \theta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial P(\bar{\theta}, x_n)}{\partial \theta_1} & \cdots & \frac{\partial P(\bar{\theta}, x_n)}{\partial \theta_m} \end{bmatrix}$$

From this matrix, we can select the m most linearly independent configurations in the training set, representing the maximally diverse subset of m configuration-energy pairs in the training set. Such a selection is called **D-optimal**, meaning that the information content of the data subset is maximized relative to the available data and the size of the subset. The process of choosing this D-optimal subset is known as **active selection**, and can be equivalently understood as finding the $m \times m$ submatrix of B with maximal volume. In other words, we aim to populate a square submatrix A with row entries from B so as to maximize the value of $|\det(A)|$. This is accomplished by use of the MaxVol algorithm [51].

Finally, the **extrapolation grade** γ of a configuration x_i is defined as the maximum factor by which the value of $|\det(A)|$ could change by the addition of a row corresponding to x_i. If the extrapolation grade of x_i is greater than 1, the volume of configuration-energy space spanned by the training set will be increased by the inclusion of x_i. Stated differently, if $\gamma(x_i) > 1$, then x_i resides outside the volume of configuration-energy space currently spanned by the training set.

The extrapolation grade can be used as an indicator of the likely error on a configuration by the following reasoning. We can think of the actively selected set of m configurations as points in configuration space defining the boundaries of a region within which the energy predicted by the fitted potential should vary smoothly, due to the polynomial nature of the fit, from one boundary value to the other. In this sense, the energy error for a configuration in this region should be "limited." Outside of this region, however, the polynomial fit is unbounded, so the errors can be extreme.

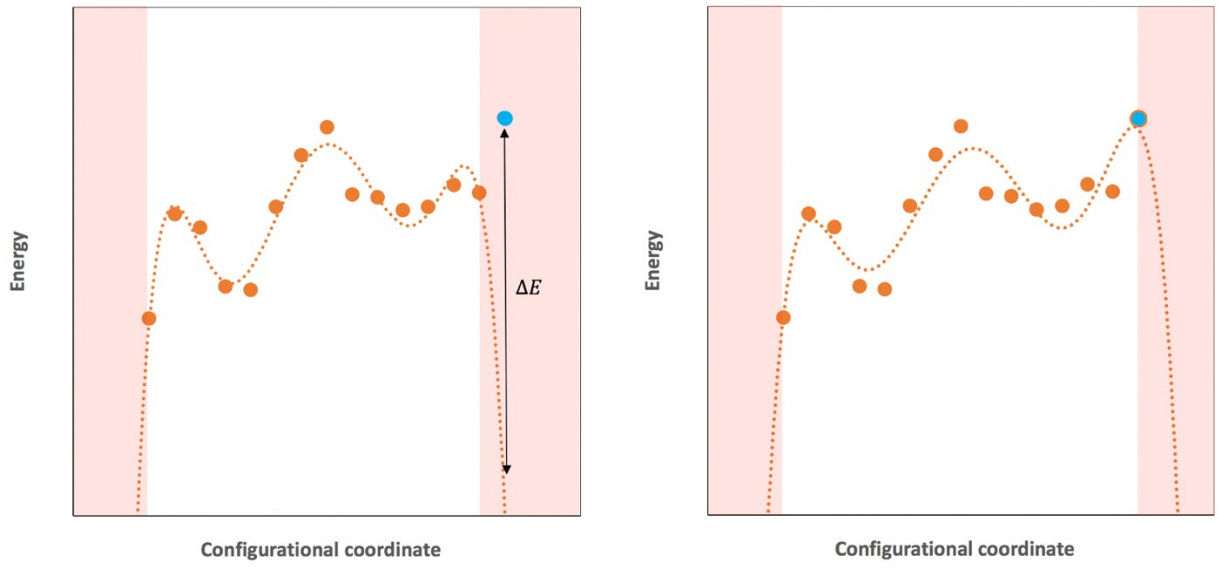

Figure 4.4: A simplified example of interpolation and extrapolation. Left: a 6th-order polynomial "potential" fitted to a set of "training data" with only one configurational dimension. Outside of the region of configuration space spanned by the training data, prediction errors can be large, as shown for the blue structure which lies in the extrapolating region. Right: the extrapolating structure is added to the training data and the polynomial is re-fit, avoiding the prediction error.

Chapter 5. Active learning

5.1 The goals of active learning

As described in the last chapter, moment tensor potentials can learn to approximate a potential energy surface by interpolating quantum-mechanical data, providing accurate energy evaluations at very low cost. We would like to be able to use moment tensor potentials for energy evaluation and structural relaxation in GA to accelerate the structural discovery of ligated clusters, which are relatively large and unwieldy to calculate with DFT.

However, moment tensor potentials produce unreliable results when used for the relaxation or energy evaluation of configurations that extrapolate from the region of configuration space spanned by the data on which the potential was trained. Poor results from such evaluations can disrupt the operation of the genetic algorithm. In particular, experience has shown that the potential can falsely predict that certain extrapolating (and often unphysical) configurations are much lower in energy than any interpolating configurations, and the pool of the genetic algorithm can become "poisoned" as a result, with these misevaluated configurations being reproduced at the expense of all other pool candidates and coming to dominate the pool. Even if such pathological results do not arise from relaxations of extrapolating clusters, a genetic algorithm driven by moment tensor potentials is only capable of identifying locally optimal structures within the interpolating region, and then only those that are comprised of atomic environments represented in the training data, which may or may not be sufficient to represent the globally optimal structure. In short, when machine-learned interatomic potentials are used for energy evaluation, GA's progress towards the global minimum is hampered by the limited data available to the potential at any given point.

Our group has developed an active learning approach to address this issue, robustly integrating moment tensor potentials for structure optimization with a genetic algorithm for global structural search. A trained moment tensor potential is used to relax and evaluate all GA-generated candidate clusters which are sufficiently similar to the data the potential was trained on, i.e. whose extrapolation grade is below a user-set threshold. Clusters with extrapolation grades above the threshold are deemed unviable for moment tensor potential calculation and are evaluated instead with DFT for a limited number of ionic steps. The data produced by these DFT calculations are used to retrain the potential, improving its accuracy on-the-fly throughout the operation of the genetic algorithm. DFT relaxations are also performed on the pool clusters at every retrain, and the data from these relaxations are added to the training set to specifically improve the ability of the moment tensor potential to describe the lowest-energy regions of the potential energy surface.

5.2 Active learning workflow

The workflow for active learning is shown in **Figure 5.1**, and involves three basic phases: selection, reevaluation, and retraining. The genetic algorithm is initialized with a trained moment tensor potential and a file containing its training set included in the calculation directory. For each new structure generated by GA, an extrapolation grade is calculated with respect to the potential and its training set using the MLIP package's internal "calc-grade" routine. If the extrapolation grade for the initial structure is larger than the user-set value "BREAKTHRESHOLD," the initial structure is copied to a database, the **training pool**, for later DFT evaluation, and a new structure is generated. Otherwise, the moment tensor potential is used to relax the initial structure.

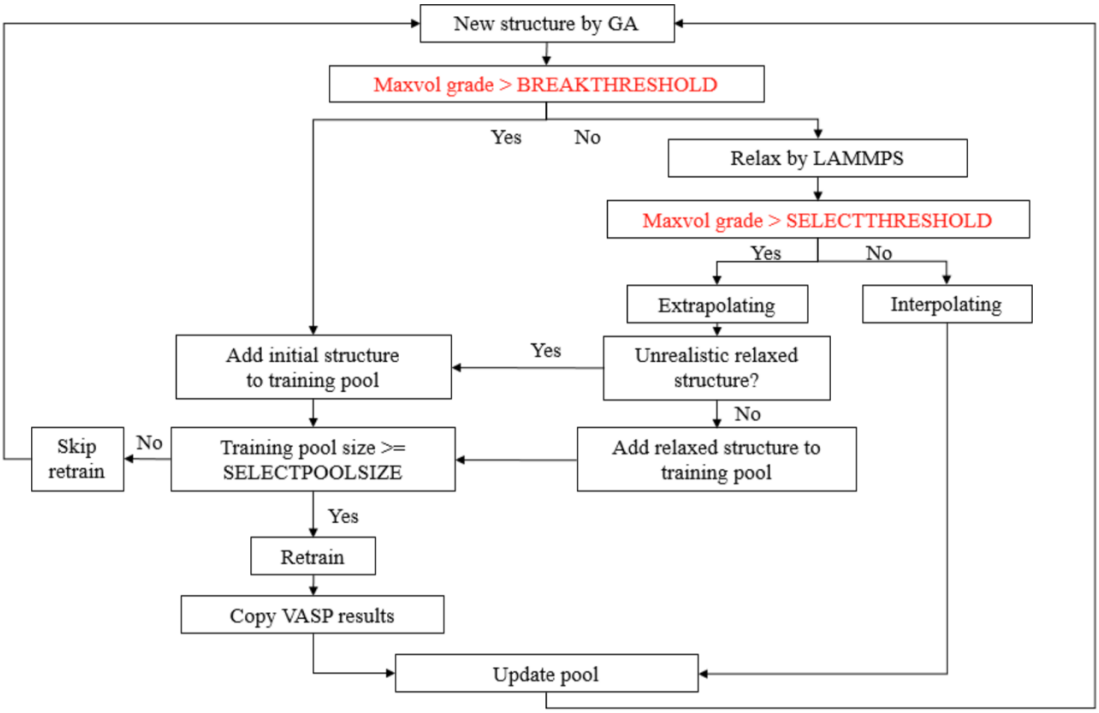

Figure 5.1: Active learning workflow. Image author: Yunzhe Wang.

Next, the extrapolation grade of the relaxed structure is calculated, and if this grade exceeds the user-set "SELECTTHRESHOLD"—indicating MTP was unable to relax the candidate to a region of configuration space acceptably near its training set—the candidate is **selected** for DFT evaluation. In this case, all interatomic distances are checked to establish whether the relaxed structure is physically realistic before passing the structure to DFT, since MTP can produce unphysical configurations when tasked with optimizing structures beyond its training region, and DFT may be unable to handle such configurations. In particular, if the relaxed structure has atoms that overlap or are detached from one another, we add the initial structure to the training pool instead. When the training pool is filled, i.e. when the number of selected candidates is greater than the user-set "SELECTPOOLSIZE", we **reevaluate** all of the candidates in the training pool with DFT and **retrain** MTP with the new data from these *ab initio* evaluations incorporated into the training set.

To ensure correct ranking of the low-energy structures and to improve the moment tensor potential's ability to describe the lowest-energy region of configuration space yet explored by the genetic algorithm, DFT calculations are also performed on the structures in the pool. After the pool calculations are evaluated by DFT, the pool is updated with the DFT energies and the candidates are reranked accordingly. Along with ensuring that the pool energies are accurate, ionic relaxation of the pool clusters by DFT can aid in the discovery of new lower-energy configurations.

In addition to the values BREAKTHRESHOLD and SELECTTHRESHOLD, which establish how the conservative the active learning routine should be in checking MTP's results by *ab initio* reevaluation, and SELECTPOOLSIZE, which establishes how frequently the reevaluation and retraining loops should occur, there are a number of user-settable parameters which affect the retraining process for MTP, especially by affecting the range of data added to the training set. First, a different number of ionic steps can be specified for DFT calculations depending on whether the cluster being evaluated is extrapolating or a member of the pool. For example, at each reevaluation, a full relaxation could be conducted for pool clusters, and only a single-point calculation made for extrapolating clusters. In this way, the potential may be supplied with more samples of the low-energy region of interest for a given amount of computational expenditure. Alternatively, more ionic steps could be conducted for extrapolating clusters than for the pool clusters, allowing DFT relaxation to reveal new local minima in regions of configuration space to which the potential is relatively unexposed.

Additionally, the values "HIGHENERGYTHRESHOLD" and "LOWENERGYTHRESHOLD" can be used to select the range of ionic steps from each DFT relaxation that should be included in the training set. Ionic steps with energies greater than HIGHENERGYTHRESHOLD (eV) above the energy of the final ionic step in a relaxation are excluded from the training set, whereas all ionic

steps with energies within LOWENERGYTHRESHOLD (eV) of the final ionic step will be included. Ionic steps with energies between these two thresholds will be added by default, but this can be further parameterized with the "INCLUDEINTERPOLATING" option, which when disabled will calculate the extrapolation grade of all ionic steps from each DFT optimization and select only those that are extrapolating to be added to the training set.

5.3 The influence of the moment tensor potential

Since the extrapolation grade of each new candidate structure depends on the training state of the moment tensor potential and the configurations included in its training set, these conditions will substantially affect the behavior and efficacy of the active learning algorithm. A key point of the active learning approach is that we use the extrapolation grade, a measure of the proximity of a configuration to the training data, as a predictor for the reliability of a MTP calculation (i.e., of the error with respect to DFT). In fact, this relationship predicts well in only one direction: candidates that extrapolate can be expected to have high error, but interpolating candidates may or may not have low error. This is because the interpolating region is defined by a volume of configuration space but provides no condition on the resolution of features within this space; as a result, relevant energy-structure relationships may not be captured. The worst-case scenario is a potential that "doesn't know what it doesn't know," rarely triggering DFT reevaluation, yet providing large errors on interpolating configurations of interest (particularly low-energy configurations). This can be the case with a training set that spans a large range of configuration space with insufficient detail, as when a moment tensor potential with too small of a basis for the system being studied is trained with an actively selected dataset.

5.4 Construction of the training set

Our active learning scheme aims at the selection of salient training information to maximally improve the performance of MTP with the minimum computation of first-principles data, and at the preemptive bypassing of MTP calculations that are expected to be inaccurate. As a further measure to augment the performance of MTP for structural search, we can choose to weight the lowest-energy configurations in the training set. With the "SCALETOPCANDIDATES" option, after reevaluation of all extrapolating and pool candidates, MLIP's "select-add" routine will be used to choose a maximally diverse subset of all DFT-evaluated configurations. To this actively selected subset, we add a user-settable number of copies ("TOPMULTIPLE") of the lowest-energy DFT-evaluated configurations ("TOPSELECTION") to construct the training set. For example, the training set may comprise 10 copies each of the 200 lowest-energy-by-DFT configurations, plus an actively selected, optimally diverse subset of all configurations. This method helps ensure that MTP learns the atomic environments comprising low-energy configurations with good accuracy, because by adding multiple copies of these configurations to the training set, we proportionally multiply the loss assigned for error on these configurations during retraining. This approach was found to improve the ability of MTP to characterize and rank low-energy configurations compared to training with an unweighted dataset, as discussed in Section 6.1 and 6.2.

It should be noted that when this method is used for retraining, a new training set is constructed at each retraining, and the size of this training set is fixed (since MLIP's select-add feature selects a fixed number of configurations that depends on the number of functions in the potential's basis set - see Section 4.5). This has the important advantage of limiting the time spent in retraining. However, it also means that some configurations are "forgotten" with each successive

retrain. In particular, these forgotten configurations will be those that do not maintain their position in the lowest-energy TOPSELECTION candidates, and that also do not belong to the optimally diverse subset of structures in the dataset.

Chapter 6. Results and discussion

6.1 Training methods for moment tensor potentials for GA

A moment tensor potential intended to substitute for first-principles calculations in a genetic algorithm must meet specialized requirements:

1. Prediction accuracy on low-energy configurations matters much more than prediction accuracy on less stable configurations, since the genetic algorithm will use the low-energy configurations predicted by MTP to produce new candidates.
2. The potential needs to be able to rank low-energy configurations by their energy in good agreement with DFT, so as to be able to accurately construct the pool. Performance at ranking is more important than absolute energy error, as the pool will be reevaluated by DFT at every retraining cycle; it is acceptable if the energies assigned by MTP to the pool candidates change once checked by DFT, as long as the pool candidates are still the stablest structures known. Finally:
3. Systematic overestimation or underestimation of energy should be avoided. A tendency to underestimate energy is more tolerable than a tendency to overestimate energy, as there is a practical risk in the latter case that, following DFT evaluation of the pool, the DFT-calculated energies will be lower than the values that MTP can possibly predict for any configurations, thereby preventing the pool from changing.

MLIP's integrated training procedure for moment tensor potentials will find a parameterization that minimizes the energy error across the training set, which should result in a potential that performs well at predicting energies of configurations that are similar to those in its training set. Therefore, our aim is to construct training sets that will produce a potential that complies with

requirements 1-3 above. To test different approaches to this problem, training data on the $Au_{18}(SCH_3)_{14}$ system was generated in two stages.

First, the genetic algorithm's ligated cluster initialization routine (see Section 3.2) was used to randomly ligate Au_{18} structures sourced from our group's Quantum Cluster Database. These randomly generated $Au_{18}(SCH_3)_{14}$ structures were then relaxed using spin-polarized density functional theory as implemented in the Vienna *Ab initio* Simulation Package (VASP) using PAW-PBE pseudopotentials, a plane-wave cutoff of 500 eV, a single gamma-centered k-point, an energy convergence criteria of 10^{-4} eV, and a force convergence criteria of 0.1 eV/Å. A total of 200 structures were generated and relaxed, producing 4461 ionic steps that were used as training configurations. This dataset (**Training Set 1**) was used for the initial training of moment tensor potentials with basis sets of $\text{level}_{\max} = 10, 12$, and 14, all of which used radial basis sets of $N_Q = 8$, inner cutoff radii of 0.8 Å, and outer cutoff radii of 8.0 Å.

Next, the aforementioned potential of $\text{level}_{\max} = 10$ was used for 10 independent genetic algorithm runs to produce and relax 200 candidates each, resulting in 4000 configurations total, corresponding to the initial and relaxed states of 10 sets of 200 candidates. Each of these 4000 configurations was evaluated by a single-point DFT calculation using the same settings as above. This second dataset of 4000 DFT configurations was split into halves, with half reserved for validation and half used for the retraining of moment tensor potentials (**Training Set 2**) by a variety of different methods.

The different training methods investigated are described in **Figure 6.1**. Here, the performance of each different training strategy is compared in terms of the RMSE, average error, and Spearman rank correlation coefficient against a test set of 100 conformers of $Au_{18}(SCH_3)_{14}$

with DFT energies ranging from to -390.583 to -394.195 eV. This test set represents the 100 lowest-energy structures of the 2000 reserved for validation.

Figure 6.1 demonstrates that by numerically weighting low-energy configurations in the training set, the performance of MTP in the low-energy region can be improved. Every weighted training strategy investigated decreased the root-mean-square energy error and the magnitude of the average of errors on low-energy configurations relative to the unweighted control strategy.

Interestingly, however, only the weighted strategies that actively selected from the total set of training configurations improved the ranking correlation on low energy structures. All of the weighted strategies that included all training configurations resulted in lower ranking correlation coefficients than the unweighted control strategy. In addition, most of the strategies which included all training configurations exhibited a negative average of errors, indicating a systematic overestimation of energy (or underestimation of stability) compared to DFT.

These trends could simply arise because we apply a greater effective weight to the low-energy clusters when we include only an actively-selected subset of the remainder of the training set. An equivalent weight could be achieved for the strategies that incorporate the entire set of training data by applying a larger multiple to the population of low-energy candidates, but the consequence would be much more time spent in training.

The two best-performing strategies in terms of ranking correlation used potentials of $\text{level}_{\max} = 14$ and $N_Q = 8$ and actively-selected training sets augmented with 5 copies of the 100 or 200 stablest candidates ("14g top 100/200 5x + active"). Nearly equivalent results were obtained with a potential of $\text{level}_{\max} = 12$ and $N_Q = 8$ and an actively-selected training set with 5 added copies of the 100 stablest candidates ("12g top 100 5x + active"). The strategy using a potential of $\text{level}_{\max} = 10$, $N_Q = 8$, with 20 copies of the 200 stablest candidates added to an

Strategy	RMSE (eV)	Bias (eV)	Ranking correlation
14g top 200 5x + active	0.49342	0.15352	0.77678
14g top 100 5x + active	0.52165	0.18067	0.76580
12g top 100 5x + active	0.51535	0.16036	0.76055
10g top 200 20x + active	0.49245	0.07459	0.75920
12g top 200 5x + active	0.51453	0.13467	0.74842
10g top 200 5x + active	0.51304	0.05306	0.74267
12g top 50 5x + active	0.53038	0.20566	0.74087
10g top 50 5x + active	0.55227	0.20240	0.73535
10g top 100 5x + active	0.55015	0.11356	0.73440
10g top 100 20x + active	0.52458	0.10462	0.72996
14g top 50 5x + active	0.53787	0.19555	0.72869
10g all (control)	**0.91401**	**-0.72324**	**0.71742**
10g top 200 20x + all	0.53490	-0.03179	0.71695
10g top 50 20x + active	0.56097	0.21167	0.71106
10g top 200 + active	0.55999	0.01421	0.70681
10g top 50 5x + all	0.72165	-0.47484	0.70643
10g top 100 5x + all	0.67272	-0.37390	0.69474
10g top 100 20x + all	0.55084	-0.04820	0.69258
10g top 50 20x + all	0.55955	-0.09768	0.68061
10g top 200 5x + all	0.63969	-0.29478	0.66334
10g top 100 + active	0.59218	0.00275	0.64516
10g top 50 + active	0.56917	-0.04798	0.63670

Figure 6.1: Different training strategies and their effects on MTP performance. Moment tensor potentials of three different basis sizes were prepared: $\text{level}_{\max} = 10$, $N_Q = 8$ ("10g", blue rows); $\text{level}_{\max} = 12$, $N_Q = 8$ ("12g", green rows); and $\text{level}_{\max} = 14$, $N_Q = 8$ ("14g", orange rows). To construct the training set, training configurations were sorted by energy and the lowest-energy 50, 100, and 200 candidates were extracted as subsets ("top 50," "top 100," "top 200"). These subsets were added in multiples (one copy, 5 copies, "5x", or 20 copies, "20x") to either the entire set of training configurations ("+ all"), or to an actively-selected subset of the training configurations obtained via MLIP's "select-add" functionality ("+ active"). As a control, a 10g potential was trained on the entire set of training configurations without modifications. In all cases, training was limited to a maximum of 1000 iterations, and energy, force, and stress weights applied during training were 1, 0.01, and 0, respectively. Strategies are listed in descending order of their ranking correlation coefficients.

actively-selected training set ("10g top 200 20x + active") also tested well, but the size of the training set in this case implies that retraining will be much slower than with the other competitive strategies.

This comparison of training strategies does not aim to be exhaustive, but to give an indication of what might constitute a good approach for our purposes. Potentials of $\mathrm{level}_{\max} = 12$ and 14 tended to outperform potentials of $\mathrm{level}_{\max} = 10$, and methods that actively selected from the training configurations were broadly better than those that used all training configurations. Scaling the lowest-energy configurations by a multiple of 20 did not give any consistent advantage over a multiple of 5, while scaling a larger selection of low-energy configurations—using the stablest 100 or 200 vs. the stablest 50 structures—tended to provide better ranking correlation on the validation set.

6.2 Weighting of low-energy configurations during active learning

Beyond the initial training of potentials, the weighting methods evaluated in Section 6.1 can be applied for the retraining of potentials during active learning, as a means to sustain the potential's accuracy as the genetic algorithm explores progressively lower-energy regions of configuration space. This is the intended use of the SCALETOPCANDIDATES feature described in Section 5.4. The efficacy of this approach is demonstrated in **Figure 6.2,** which plots the energies of the clusters in the pool as calculated by DFT against their MTP-predicted energies at each active learning reevaluation stage. A genetic algorithm run on the $Au_{18}(SCH_3)_{14}$ system was started with a moment tensor potential of $\mathrm{level}_{\max} = 12$ and $N_Q = 8$ trained on Training Set 1. Active learning was enabled to retrain this potential on-the-fly, using a SELECTTHRESHOLD of

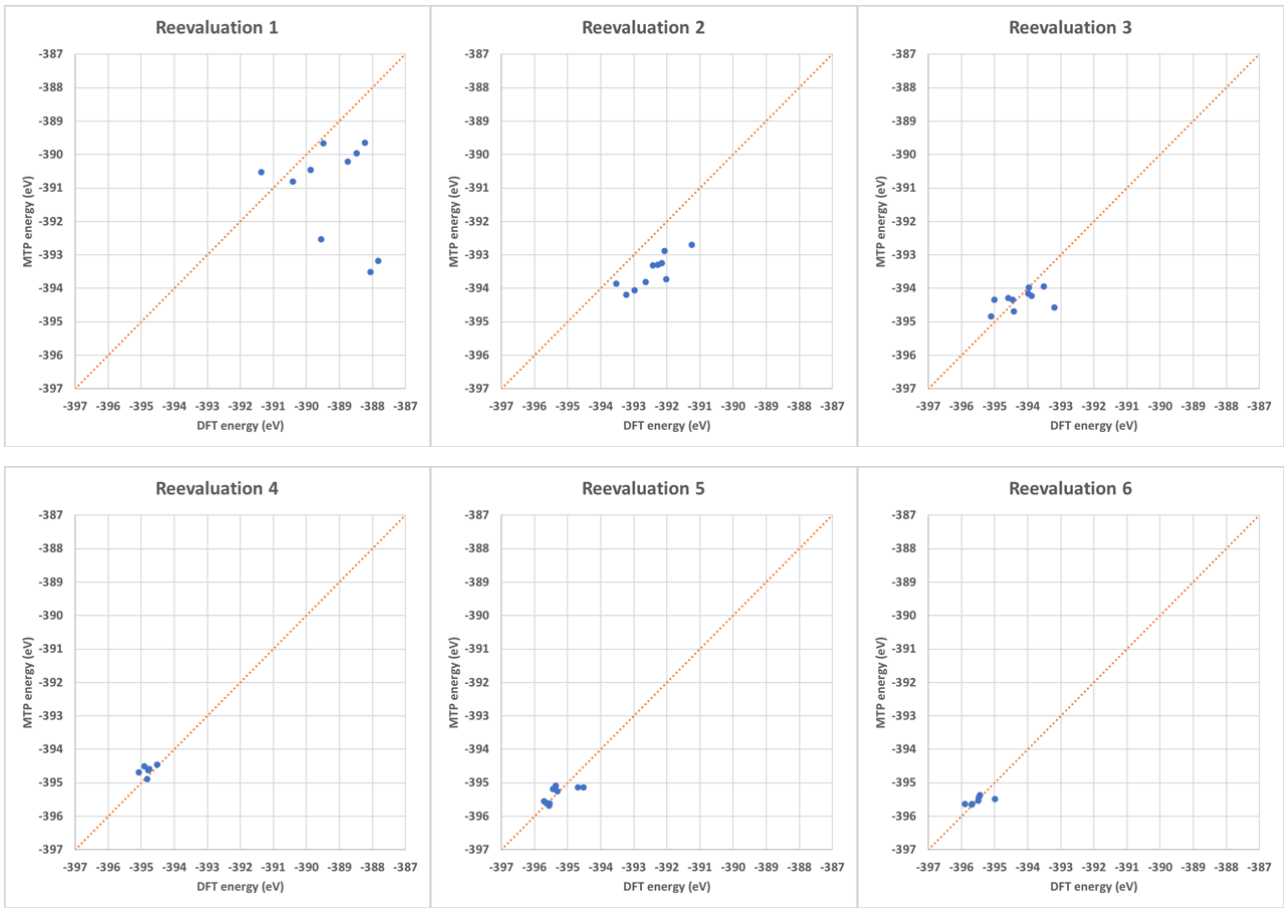

Figure 6.2: Progression of pool parity with successive weighted retrains.

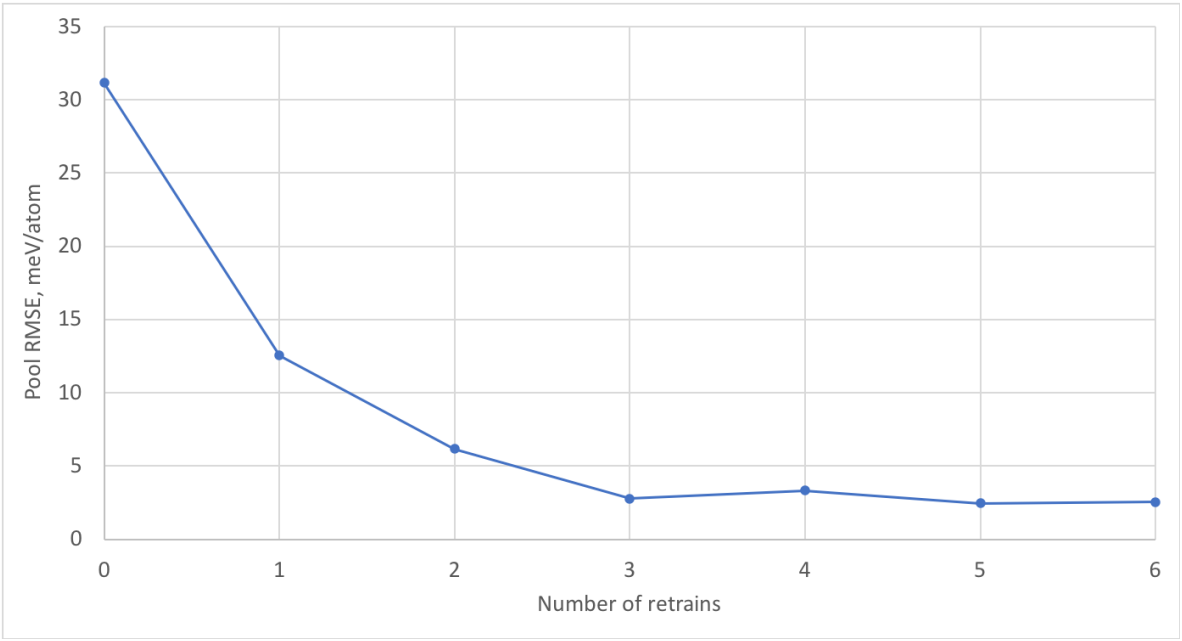

Figure 6.3: Root-mean-square error on clusters in the pool with successive weighted retrains.

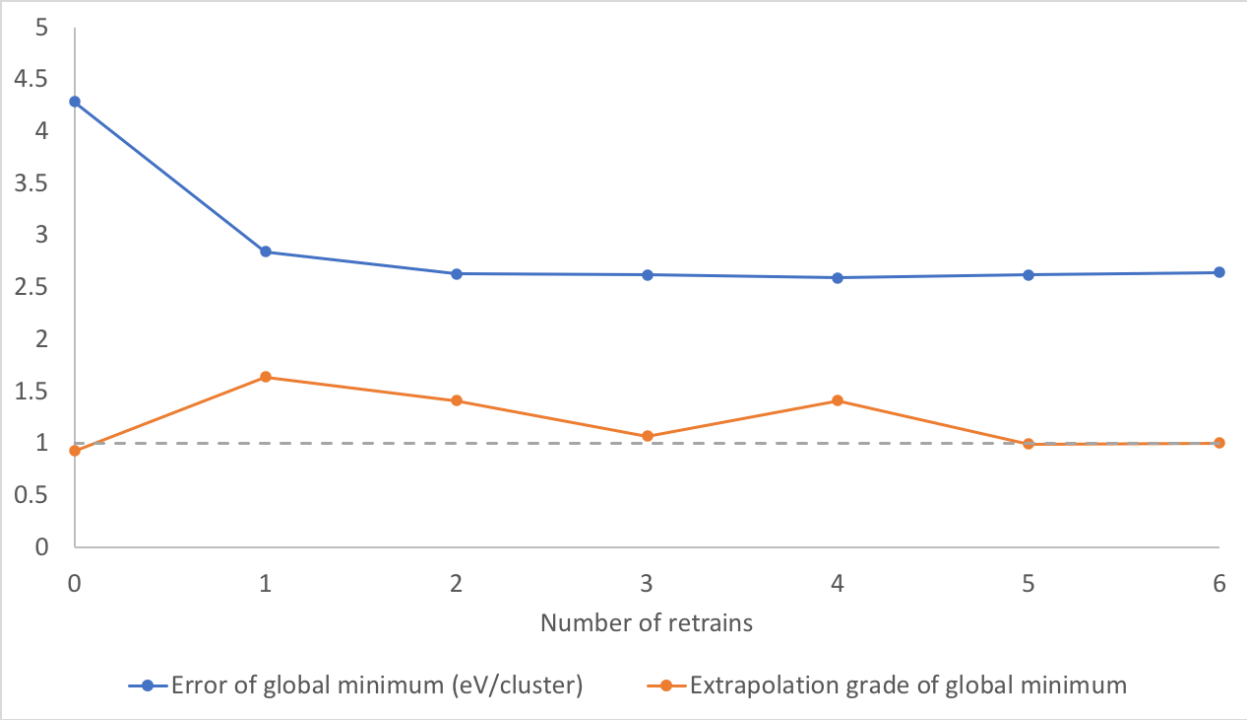

Figure 6.4: Energy error and extrapolation of the global minimum with successive retrains.

1.01, a BREAKTHRESHOLD of 10, and a SELECTPOOLSIZE of 10. For retraining, SCALETOPCANDIDATES was enabled with a TOPSELECTION of 100 and a TOPMULTIPLE of 5. During DFT reevaluation, static calculations were run for extrapolating clusters, whereas pool clusters were relaxed for a maximum of 5 ionic steps.

Figure 6.2 demonstrates how the moment tensor potential's accuracy on low-energy clusters is improved by on-the-fly active learning and weighted retraining in the context of an actual genetic algorithm run; MTP gets clearly better at predicting the energy of pool clusters with each weighted retrain, and the energy range of the pool steadily decreases as we would hope, indicating that the MTP-driven genetic algorithm is successfully discovering low-energy structures. The evolution of the RMSE on new clusters added to the pool with successive weighted retraining is shown in **Figure 6.3**. After 6 retrains, MTP is able to predict the energy of new clusters entering the pool with RMSE of approximately 2.5 meV/atom, or around 230 meV per cluster.

These evident improvements in the prediction of pool clusters are tempered by the uncertain benefit of active learning and weighted retraining with respect to the global minimum structure, shown in **Figure 6.4**. MTP's prediction error on the global minimum structure does decline with retrain, but only modestly, from an initial overestimation of 4.29 eV (i.e., a predicted cluster energy of -392.69 eV compared with the actual energy of -396.98 eV) to a final overestimation of 2.66 eV. Additionally, no improvement to this energy error is realized beyond the second retrain. The extrapolation grade of the global minimum structure fluctuates without a clear trend; however, for the last two retrains, the global minimum is just below the threshold of interpolation (extrapolation grade of <1), meaning that if the global minimum was discovered at this point by the genetic algorithm, it would be evaluated by MTP and judged to be ~2.6 eV higher in energy than it actually is. In this circumstance, the global minimum would likely not enter the pool.

Figure 6.3 and 6.4 may seem to tell different stories; in fact, they capture different angles of the challenge of using machine-learned potential models to identify novel low-energy structures. The potential's performance can be excellent when used to evaluate configurations comprised of atomic environments similar to those it has been trained on, and relatively poor when evaluating configurations that are dissimilar. The genetic algorithm must play a complementary role by generating diverse candidates that sample configuration space widely, providing the moment tensor potential with novel training configurations that can expand its capacity.

6.3 Benchmark of genetic operations for ligated clusters

The set of genetic operations employed by GA will impact its efficiency at discovering new low-energy candidates. To assess this effect, genetic algorithm runs on $Au_{18}(SCH_3)_{14}$ were

conducted with varying sets of crossover and mutation operations. The "12g top 100 5x + active" potential described in Section 6.1 was used for energy evaluation and relaxation, with active learning used only to identify and pass over extrapolating candidates; retraining of the potential was not enabled. For each different set of genetic operations, the MTP energy of the stablest observed candidate was plotted as the algorithm proceeded for 1000 iterations; 10 runs were made for each set of operations and averaged. The results are shown in **Figure 6.5**.

All modes of operation improve over the baseline approach of random structure sampling, confirming the basic efficacy of the genetic algorithm. The simplest mode of operation, using only even crossover for structure generation ("Even"), was also the worst-performing, with the stablest structure evolved at 1000 iterations having an average energy of -395.2 ± 0.2 eV. The addition of mutation at a rate of 30% ("Even + Mutation") was an improvement, bringing the average lowest energy at 1000 iterations down by 400 meV to -395.6 ± 0.2 eV. Compared to even crossover, fitness-weighted crossover ("Weighted") and random crossover ("Random") reduced the average minimum energy at 1000 iterations by 300 meV (-395.5 ± 0.2 eV) and 500 meV (-395.7 ± 0.2 eV), respectively. An ensemble of all three crossover operations ("Even + Weighted + Random") performed essentially identically to random-only crossover. Adding mutation to this ensemble ("Even + Weighted + Random + Mutation") did not improve performance, and resulted in a statistically insignificant increase in the average lowest energy at 1000 iterations from -395.7 ± 0.2 eV to -395.6 ± 0.1 eV.

Overall, these results indicate that random-ratio crossover without mutation provides the most efficient search of configuration space among the genetic operations considered. To gain an understanding of why random-ratio crossover performs better than the other methods, and why also it behaves nearly identically to the ensemble of even, weighted and random crossover

operations, we compare the distribution of parent ratios for random-ratio crossover, fitness-weighted crossover, and the even/weighted/random ensemble of operations in **Figure 6.6**. This comparison of distributions suggests that the factor that contributes the efficiency of the random-ratio and even/random/weighted crossover modes are their inclusion of high-ratio crossover operations, where one parent cluster contributes 90 percent or more to the child cluster. Such operations are also included in the even/weighted/random ensemble, which performed equally to random-ratio crossover in identifying lowest-energy candidates, but are absent from fitness-weighted crossover, which was outperformed by random-ratio crossover by an average of 200 meV at 1000 iterations. This comparison indicates the importance of small structural changes to the performance of the genetic algorithm. "Fine-tuning" of candidates thus appears to be a useful strategy for efficiently generating lower-energy configurations, and in future work new genetic operations could be developed to more specifically exploit this effect.

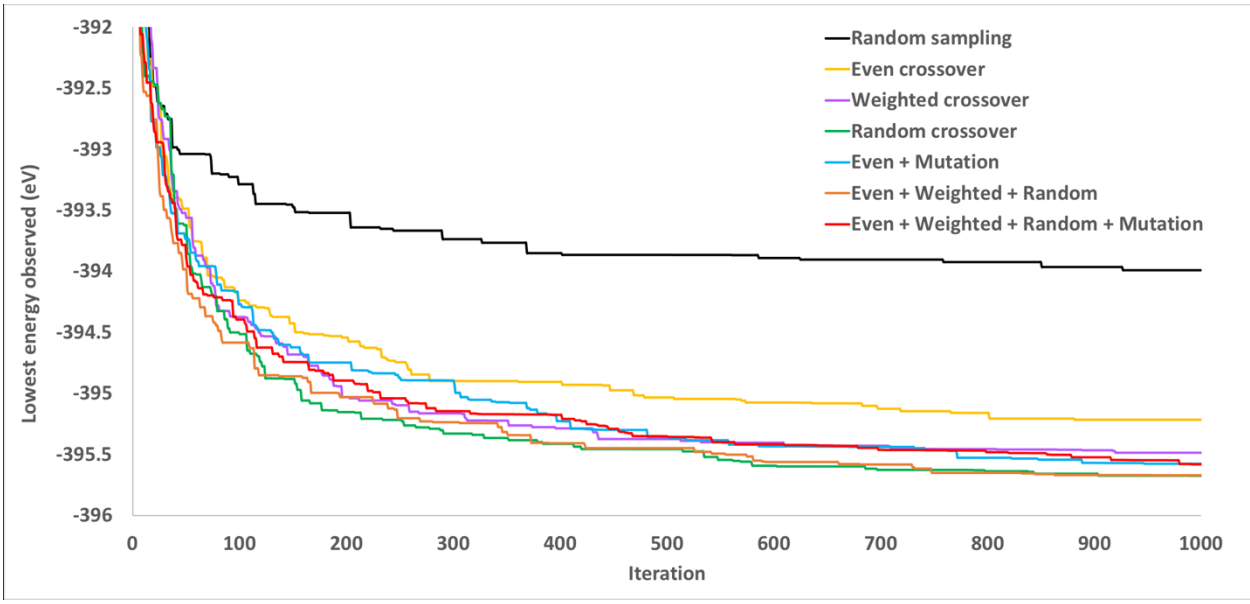

Figure 6.5: Averaged minimum energy hulls with different sets of genetic operations, n=10.

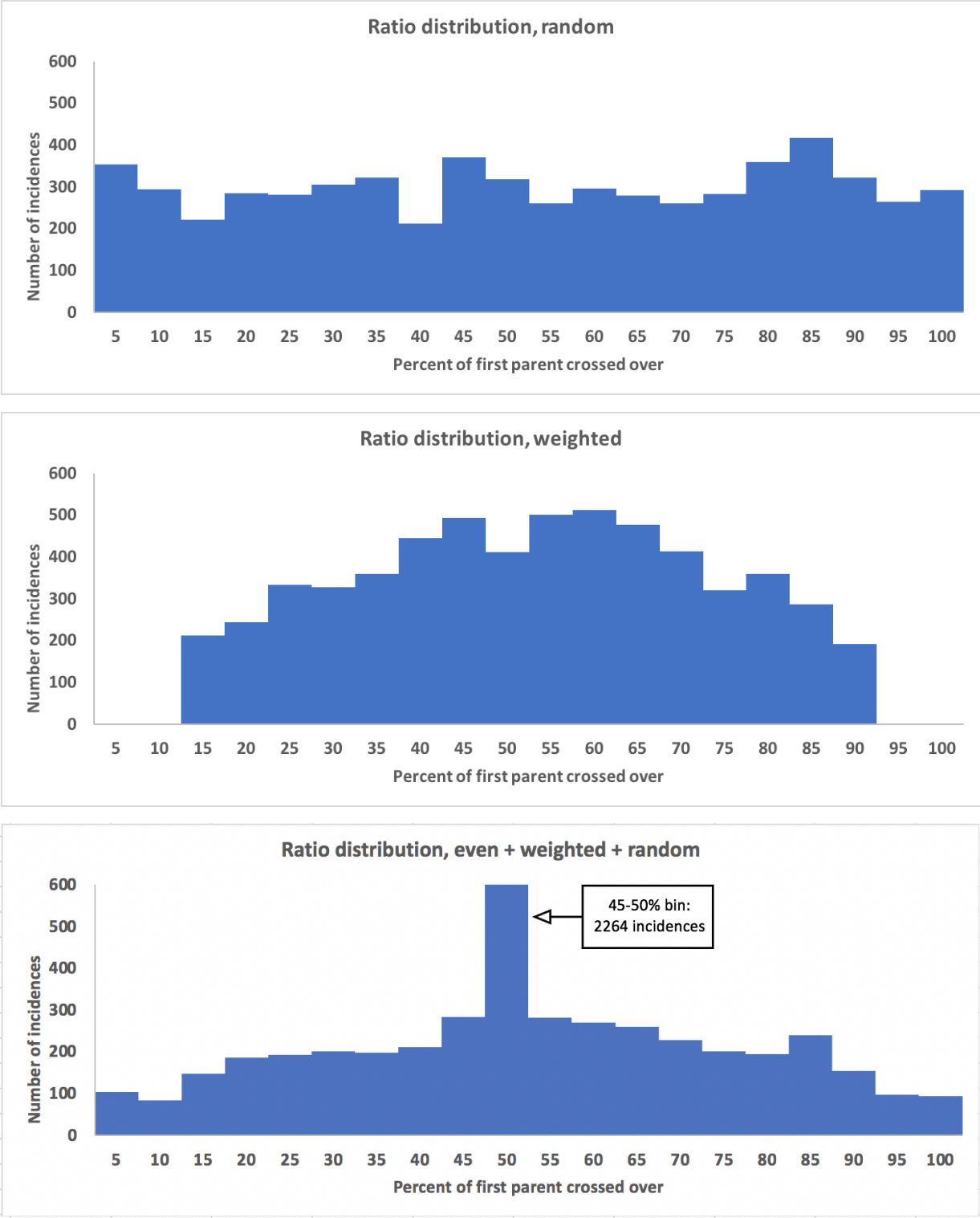

Figure 6.6: Distribution of parent ratios in different crossover modes. Histograms plotted for 6000 iterations in crossover modes "Random," "Weighted," and "Even + Weighted + Random." For the "Even + Weighted + Random" mode ensemble, at each crossover operation, a mode was randomly chosen with equal probability, i.e. 1/3 each for Even, Weighted and Random modes.

6.4 Search for the DFT ground state of $Au_4(SCH_3)_4$

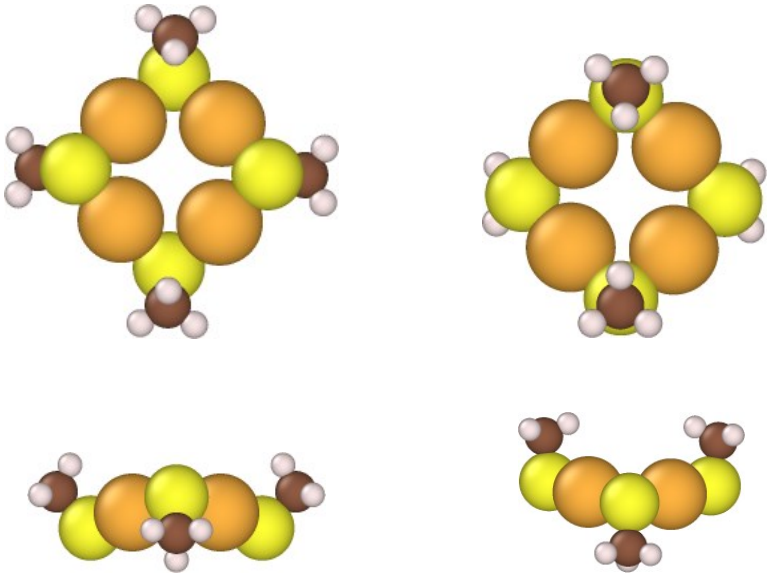

Figure 6.7: Stablest configurations of $Au_4(SCH_3)_4$ found by GA using DFT and MTP. On the left, the lowest-energy structure found by DFT-driven GA. The "12g top 100 5x + active" potential described in Section 6.1, trained only on $Au_{18}(SCH_3)_{14}$ configurations, was used with GA to discover the structure on the right.

As a preliminary test of moment tensor potentials for energy evaluation in GA, we attempted to rediscover with GA/MTP the minimum energy configuration of $Au_4(SCH_3)_4$ that had previously been identified using GA/DFT. We did not specially train a moment tensor potential on the $Au_4(SCH_3)_4$ system, instead opting to use the potential "12g top 100 5x + active" trained on $Au_{18}(SCH_3)_{14}$ configurations, described in Section 6.1. This choice was made on the hypothesis that the greater number of atoms and ligands in $Au_{18}(SCH_3)_{14}$ would equate to a much larger number of atomic environments being described than a dataset of comparable size generated for $Au_4(SCH_3)_4$, and that the physics captured in these atomic environments should largely be transferrable between the two systems, notwithstanding possible differences due to quantum finite size effects.

The results of this test are shown in **Figure 6.7** above. The lowest-energy configuration discovered by MTP-driven GA broadly agrees with the minimum discovered via DFT, being square in shape with ligand pairs across the square's diagonals pointing in opposing directions. The most notable difference between the DFT and MTP structures is in the ligand orientation; the MTP-minimum configuration has the ligands rotated relative to the DFT-minimum configuration so that the diagonal ligand pairs are closer together. The DFT-minimum configuration is more stable than the MTP-minimum configuration by 78.5 meV; their energies are -109.98264 eV and -109.90407 eV, respectively, when evaluated using the same DFT parameters. However, when the MTP used for this GA run is employed to relax the DFT-minimum configuration, it relaxes it to the MTP-minimum configuration, suggesting that the MTP-minimum configuration discovered by GA is indeed the global minimum with respect to the potential's parameterized representation of the energy landscape. Notably, the MTP-driven GA discovered this minimum-energy structure in 11 minutes while running on 6 CPUs, which is roughly equivalent to the amount of time required for a single DFT relaxation of a given $Au_4(SCH_3)_4$ configuration.

6.5 Search for the experimentally verified ground state of $Au_{18}(SR)_{14}$

Encouraged by the performance of MTP-driven GA for small ligated systems, we attempted to rediscover the structure of $Au_{18}(SR)_{14}$, the smallest thiolated gold cluster whose structure has been experimentally determined [17, 18]. In this section, we describe the results of a genetic algorithm run on the $Au_{18}(SCH_3)_{14}$ system employing active learning and weighted retraining, using a moment tensor potential of $\text{level}_{\max} = 12$ and $N_Q = 8$ initially trained on the "12g top 100 5x + active" protocol detailed in Section 6.1.

For this run, all three crossover modes (Even, Weighted and Random) were enabled, along with a mutation ratio of 10%. For retraining, SCALETOPCANDIDATES was enabled with a TOPSELECTION of 100 and a TOPMULTIPLE of 5. For active selection, we used a SELECTTHRESHOLD of 1.01 and SELECTPOOLSIZE of 10. Static DFT calculations were conducted for extrapolating clusters, while pool clusters were relaxed for up to 5 ionic steps. The initial population was taken from user-provided files of low-energy bare Au_{18} clusters from the Quantum Cluster Database, which were then randomly ligated (Section 3.2). The algorithm was allowed to run for 72 hours on 24 CPUs.

With these settings, GA/MTP/AL discovered a configuration with an energy within 0.518 eV of the experimentally derived structure. The energy hull for this run is shown in **Figure 6.8**.

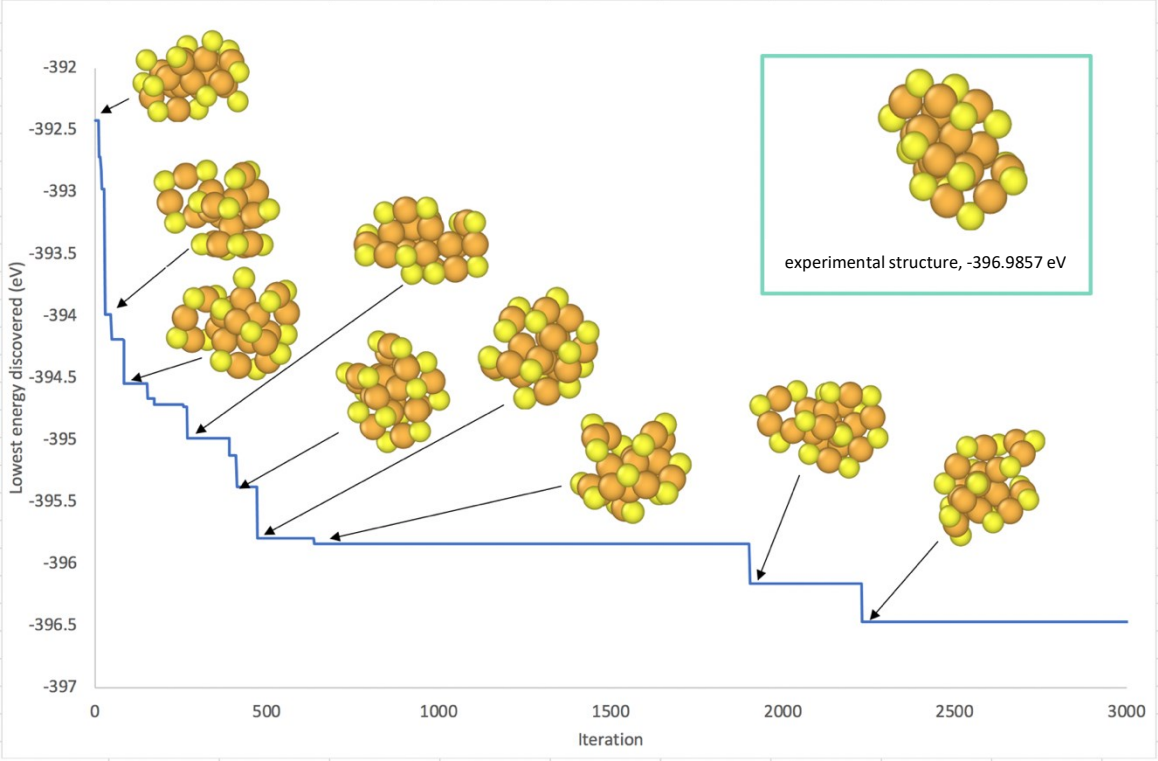

Figure 6.8: Minimum energy hull of a GA/MTP/AL run on $Au_{18}(SCH_3)_{14}$. Structures representing major transitions in the minimum energy are shown, along with the literature-reported empirical structure for $Au_{18}(SR)_{14}$ inset at top right, rendered without ligand side chains for clarity. This run proceeded for 6452 iterations, but no lower energy structure was discovered after candidate 2231.

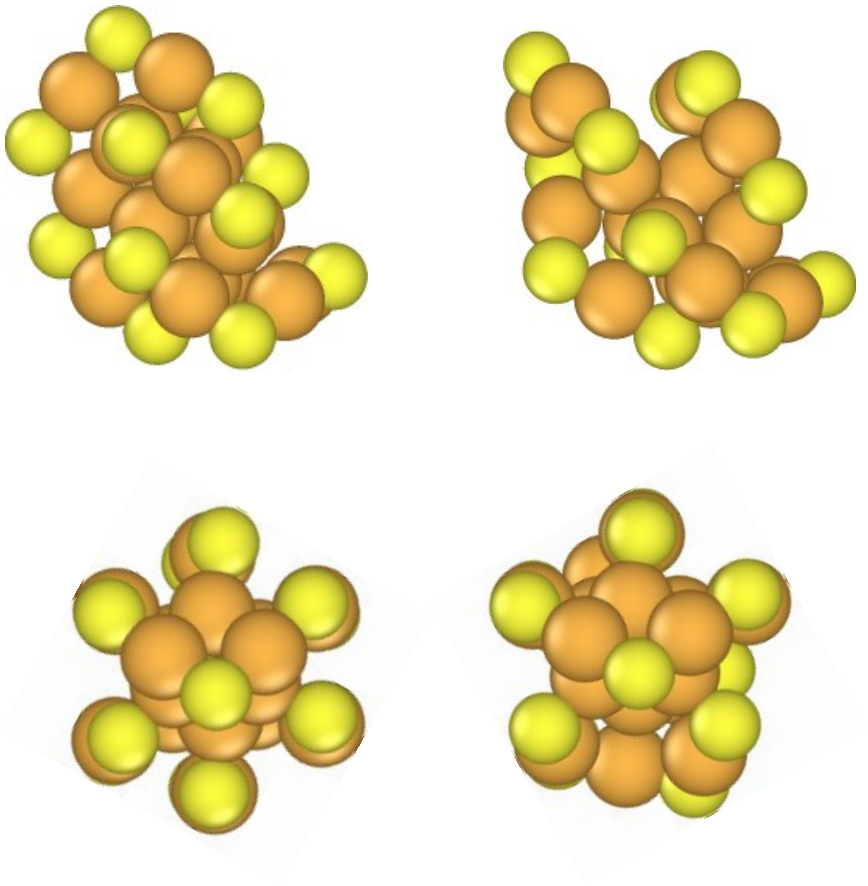

Figure 6.9: Comparison of stablest GA structure with experimental structure of $Au_{18}(SR)_{14}$. Top and bottom left: the experimentally derived structure of $Au_{18}(SR)_{14}$, shown without side chains for clarity, from two different angles. Top and bottom right: candidate 2231, the stablest structure discovered by GA/MTP/AL in the run described in this section.

The lowest-energy structure discovered by GA/MTP/AL in this run, candidate 2231, is compared to the experimental structure in **Figure 6.9**. The calculated similarity of this structure to the empirical structure is 1.35694, indicating the structures do not meet our criterion for similarity. Nonetheless, candidate 2231 has evolved important features of the experimentally known structure, particularly the $Au_4(SR)_5$ staple motif [22] and a close-packed Au_6 core of the same geometry as two layers of the three-layer Au_9 core reported in the literature [21]. The arrangement of these features is consistent with their position in the experimentally determined structure, as well.

6.6 New minimum-energy structures for $Au_{18}(SCH_3)_{14}$

Following its discovery by GA/MTP/AL, candidate 2231 was refined by a more stringent ionic relaxation by DFT to within a tolerance of 10^{-6} eV. This relaxation yielded a structure with an energy of -396.9829 eV, only 28 meV (or 0.3 meV/atom) higher than the experimental structure's energy of -396.9857 eV.

With the aim of verifying the reproducibility of this result, we conducted 12 independent repetitions of the method used to discover this low-energy candidate (i.e., GA/MTP/AL with the settings listed above, followed by tight-tolerance DFT relaxation of the stablest evolved candidate). Remarkably, two of these 12 runs (**Figure 6.10**) yielded structures with lower energies than the proposed experimental structure for $Au_{18}(SR)_{14}$.

These two structures, hereafter referred to as GA4 and GA11 and shown in **Figure 6.11**, were predicted by DFT to be more stable than the experimentally reported structure by 79.5 and 71.6 meV per cluster, respectively. While this result demonstrates the efficacy of GA/MTP/AL for structure discovery, it was initially the cause of some concern, as our hope was to validate the approach by rediscovering structures in agreement with experiment.

As mentioned, all computational work documented in this thesis to this point has been done with the gold-methanethiolate system, since these ligands (SCH_3) are the smallest thiolate and therefore the least computationally demanding to model. However, the experimental characterizations of the Jin [17] and Zhu [18] groups were both carried out on clusters ligated with cyclohexanethiolate, SC_6H_{11}, though the structure they commonly derived was reported as generally valid for $Au_{18}(SR)_{14}$. Therefore, we turn to investigate the impact of ligand substitution on the relative stability of the three lowest-energy $Au_{18}(SR)_{14}$ configurations known: the experimental structure, GA4, and GA11.

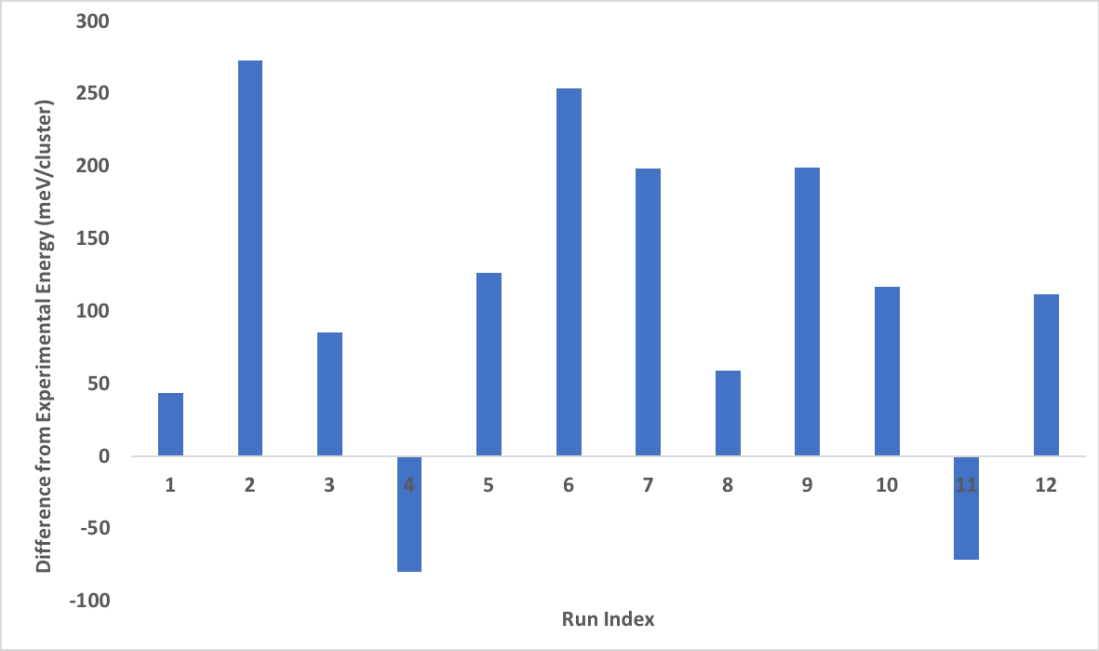

Figure 6.10: Energies of top-of-pool GA/MTP/AL candidates after fine DFT optimization. For 12 independent trials, GA/MTP/AL was used to optimize the structure of $Au_{18}(SCH_3)_{14}$ for 72 hours on 24 cores with the settings listed in Section 6.5. The lowest-energy structural candidate discovered at the end of each run was then ionically relaxed with DFT to within 10^{-6} eV per cluster. The energies are compared to the energy of the experimentally reported structure as relaxed with the same DFT settings. Runs 4 and 11 produced candidates that were stabler than the experimental structure.

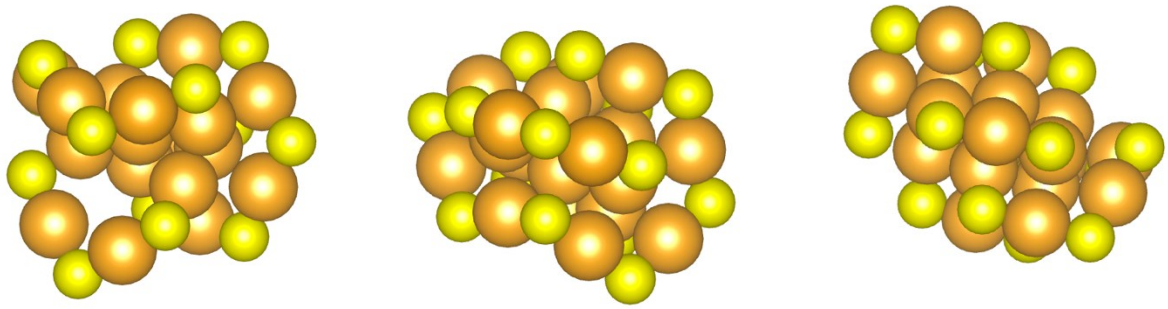

Figure 6.11: Two new minimum-energy structures for $Au_{18}(SCH_3)_{14}$. At right, the experimentally derived structure of [17] and [18]. The ligated clusters at left and in the center, GA4 and GA11, exhibit lower energies than the experimental structure when the ligating species is SCH_3. All clusters are shown without ligands for clarity. GA4 has an Au_7 core surrounded by one $Au_4(SR)_5$ motif, one $Au_5(SR)_6$ motif, and one $Au_2(SR)_3$ motif. GA11 has an Au_7 core surrounded by one $Au_8(SR)_9$ motif, one $Au_2(SR)_3$ motif, and one $Au(SR)_2$ motif.

6.7 The effect of ligand type on the ground state of $Au_{18}(SR)_{14}$

To this end, all of the -CH_3 side chains on GA4, GA11, and the experimental structure were replaced with -C_6H_{11} groups in random, nonoverlapping orientations. Since the cyclic SC_6H_{11} ligands are bulkier than the SCH_3 ligands considered previously, DFT molecular dynamics calculations were used to preliminarily optimize the positions of the substituted side chains by annealing from 300K to 0K in 450 timesteps of 1.0 femtosecond each. In this procedure, the Au and S atoms were frozen in place, while the side chain atoms were unconstrained. Three separate anneals were conducted for each configuration in order to sample a range of possible low-energy ligand arrangements. After annealing, the side chain atoms were relaxed to within a tolerance of 10^{-6} eV. This process yielded nine structures in total; their relative energies are compared in **Figure 6.12**. At this stage, five of the nine structures—all three replicates of GA11, and two of the replicates of the experimental configuration—were substantially lower in energy than the others, and the energies of these five were all within 0.4 meV/atom of each other. As a final measure, these five clusters were more finely optimized by relaxing all atoms to within 10^{-6} eV.

By this process, the experimental structure was found to be the stablest configuration of $Au_{18}(SR)_{14}$ for R = C_6H_{11}, with a calculated energy of -1449.0005 eV per cluster. GA11 was less stable by 51.6 meV for R = C_6H_{11}, with an energy of -1448.9489 eV per cluster. GA4 was the least stable configuration of the three. It should be noted that this is a reversal of the order observed for the R = CH_3 case, indicating ligand-driven control of structure for this cluster.

These results should be interpreted with some caution. The energies of the SC_6H_{11}-ligated systems depend strongly on the positioning of the ligands, as shown in Figure 6.12 by the difference in energy among ligand anneals of the same structure. With only three anneals attempted for each, it is possible that lower-energy configurations of ligands could be found for any of the

structures, and that this could upset the order we have proposed here; indeed, the different anneals of the experimental $Au_{18}(SC_6H_{11})_{14}$ structure were further apart in energy than the lowest-energy anneals of GA11 and the experimental structure were from each other. The difficulty in converging to globally optimal ligand arrangements when performing this ligand substitution is an example of the difficulty of global optimization in general, and in future work, the use of GA/MTP/AL to directly optimize systems with larger ligands may help to avoid these ambiguities.

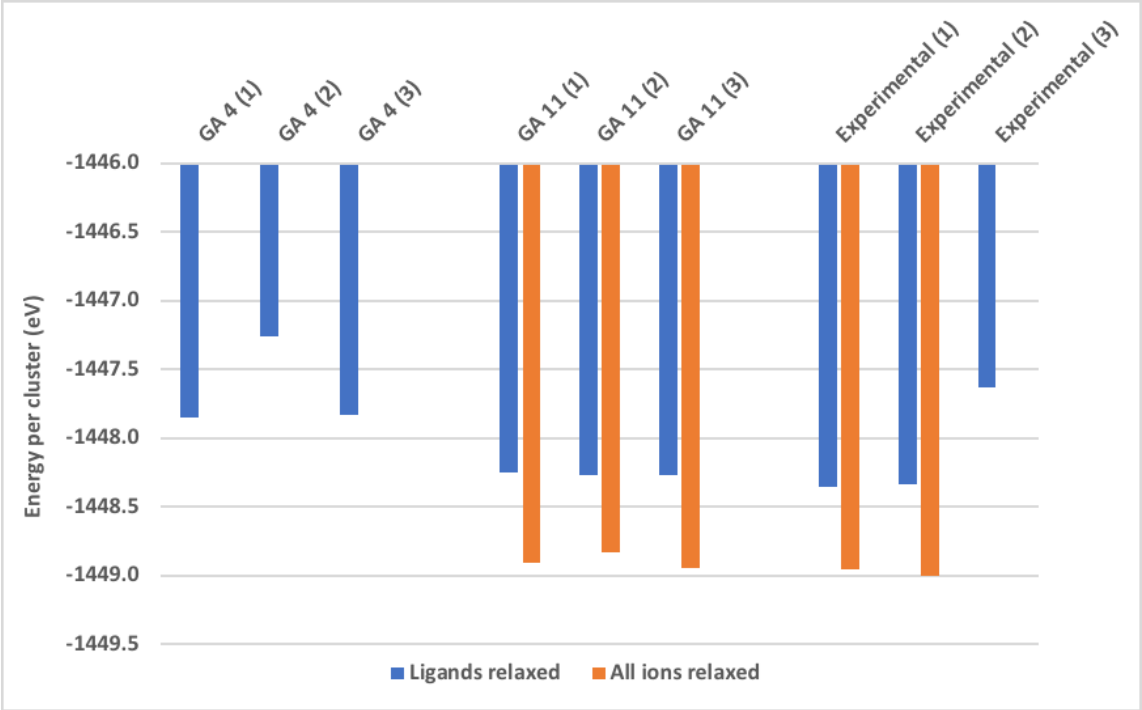

Figure 6.12: Stability of candidates with substituted SC_6H_{11} ligands. Cyclohexanethiolate-protected models of GA4, GA11, and the experimental structure for $Au_{18}(SR)_{14}$ were prepared by randomly attaching $-C_6H_{11}$ R-groups to the S active sites in nonoverlapping orientations. Three $Au_{18}(SC_6H_{11})_{14}$ replicates were prepared by this random attachment method for each cluster structure, and the ligands were annealed with DFT-MD as described in the Section 6.7. The ligands were then relaxed with DFT to within 10^{-6} eV. The resultant energies are shown in blue. The five replicates with energy below -1448 eV/cluster at this stage (GA11 1-3, Experimental 1 & 2) were then further optimized by relaxing all ions to within 10^{-6} eV. The energies of these relaxed structures are shown in orange. After relaxation of all atoms, the energies of the GA11 replicates ranged from -1448.8349 eV to -1448.9489 eV, and the Experimental replicates ranged from -1448.9557 eV to -1449.0005 eV.

Chapter 7. Conclusions and future outlook

7.1 Standing challenges

In this thesis, we have presented preliminary results on the development, testing, and application of a system that uses machine learning methods for the local and global structural optimization of ligated nanoclusters. At the time of writing, the best system presented here has been applied to discover a new global minimum for $Au_{18}(SCH_3)_{14}$ and can reliably evolve near-minimum structures for this chemistry in a few thousand iterations, starting from data from *ab initio* calculations on several hundred randomly generated structures. Our approach substantially reduces the number of first-principles calculations required compared to traditional genetic algorithms, so it has a strong advantage in the number of configurations that can be evaluated in a given time. Moving forward, more work will need to be done to improve the GA/MTP/AL system to the point where it can reliably discover the ground states of large, multi-component clusters.

With the issues of structure handling for ligated systems and robustness of the algorithm against extrapolation largely resolved, the greatest remaining challenge is to develop better ways of dynamically improving the performance of moment tensor potentials. A design goal of this system is to minimize the number of *ab initio* calculations required for operation of the genetic algorithm, but a consequence of this is that training data for the moment tensor potentials is few and far between. To the point, the GA/MTP/AL run discussed in Section 6.5 entered retraining twice, once at iteration 2387 and again at iteration 4412: at each retrain, 10 extrapolating structures and 10 pool structures were reevaluated by DFT. In other words, in a typical run, we ask MTP to make predictions on more than 2000 structures—which, when the genetic algorithm is functioning

properly, are becoming less and less like the relatively high energy structures represented in the training set—before providing it with 20 new structures to learn from. We must get the most out of the data that we gain from these rare reevaluations.

This is not a matter of optimal performance, but of function: if MTP doesn't learn the configurations being produced by GA which are lower in energy than its training data suggests, the algorithm can lose its ability to place new low-energy candidates in the pool. The price paid for the substitution of DFT with a faster machine-learned model is the additional complexity required to render these separate modes of energy evaluation compatible.

Finally, in its current state, GA/MTP/AL's ability to handle ligated systems for the thousands of necessary iterations hinges on the neglect of the ligand side chains in similarity evaluation. It is conceivable that this could become a problem for the optimization of ligated systems where the orientation of the ligand side chains themselves play a critical role in the stability of the ground state. Thus, faster methods of similarity evaluation may be worthwhile; for example, it may be a good tradeoff to evaluate the similarity of each new candidate with respect to only a limited number of the lowest-energy candidates.

7.2 Avenues to explore

Active learning has been essential to allowing MTP to intelligently improve throughout the GA run, and multiplying the population of low-energy candidates in MTP's dataset prior to retraining has helped to maintain agreement with DFT where MTP needs it the most. A logical next step may be to introduce a reinforcement learning framework, where (for example) the most weight in training is placed on structures for which MTP's predictions are the farthest off from the DFT results.

Additionally, the experiments of Section 6.1 gave some indication that larger basis sets improve the performance of MTP for the gold thiolate system considered here. Further study is warranted to see if this bears out in practice, since potentials with larger basis sets will require more training data and take longer to relax and evaluate structures, which could doubly count against their efficiency for GA/MTP/AL.

As a longer-term project, it may be worthwhile to use more than one moment tensor potential for the purpose of GA/MTP/AL, which could be arranged in a number of different ways. Interesting options include employing separate potentials for coarse and fine evaluation, parallel potentials that could evaluate candidates in ensemble to average out errors, and the use of multiple low-basis potentials trained on dissimilar subsets of candidates to "tile" the potential energy surface for a better balance of accuracy and parallelization than a single potential of larger basis. These options suggest themselves as natural extensions of the parallel pool-based genetic algorithm, though the conceivable improvements in performance must be balanced against the certainly greater complexity.

Lastly, the basic hypothesis of GA—that good wholes are made of good parts—has a pleasing compatibility with MTP's approach of evaluating energy by summing local environments. Conventionally, the "good parts" here are only implicit, since GA relies on energy evaluation methods that treat each configuration holistically. The process of training a moment tensor potential, however, relies exactly on the digestion of training configurations into local environments and the estimation of the average contributions they make to stability. By making use of this information, we might develop more informed and efficient ways of modifying and recombining candidates, going beyond the simple stochastic processes of crossover and mutation.

References

1. Castleman AW, Bowen KH. Clusters: Structure, Energetics and Dynamics of Intermediate States of Matter. *J. Phys. Chem.* (1996) 100, 12911-12944.
2. Ferrando R, Jellinek J, Johnston RL. Nanoalloys: From Theory to Applications of Alloy Clusters and Nanoparticles. *Chem. Rev.* (2006) 108, 3, 845-911.
3. Jin R, Zeng C, Zhou M, Chen Y. 2016. Atomically Precise Colloidal Metal Nanoclusters and Nanoparticles: Fundamentals and Opportunities. *Chem. Rev*. (2016) 116, 18, 10346-10413.
4. Tao Y, Li M, Ren J, Qu X. Metal nanoclusters: novel probes for diagnostic and therapeutic applications. *Chem. Soc. Rev.* (2015) 44, 8636-8663.
5. Tyo EC, Vajda S. Catalysis by clusters with precise numbers of atoms. *Nature Nanotechnology* (2015) 10, 577-588.
6. Lee TH, Gonzalez JI, Zheng J, Dickson RM. Single-Molecule Optoelectronics. *Acc. Chem. Res.* (2005) 38, 7, 534-541.
7. Abbas MA, Kamat PV, Bang JH. Thiolated Gold Nanoclusters for Light Energy Conversion. *ACS Energy Lett.* (2018) 3, 4, 840-854.
8. Xu YF, Yan ML, Sellmyer DJ. FePt Nanocluster Films for High-Density Magnetic Recording. *J. Nanosci. Nanotechnol.* (2007) 7, 1, 206-224.
9. Schmid G, Bäumle M, Geerkens M, Heim I, Osemann C, Sawitowski T. Current and future applications of nanoclusters. *Chem. Soc. Rev*. (1999) 28, 179-185.
10. Johnston RL. Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries. *Dalton Trans.* (2003) 4193-4207.
11. Weigend F, Ahlrichs R. Quantum chemical treatments of metal clusters. *Phil. Trans. R. Soc. A* (2010) 368, 1245-1263.
12. Tsukuda T. Toward an Atomic-Level Understanding of Size-Specific Properties of Protected and Stabilized Gold Clusters. *Bull. Chem. Soc. Jpn.* (2012) 85, 2, 151-168.
13. Kilina S, Ivanov S, Tretiak S. Effect of Surface Ligands on Optical and Electronic Spectra of Semiconductor Nanoclusters. *J. Am. Chem. Soc.* (2019) 131, 7717-7726.
14. Higaki T, Li Y, Zhao S, Li Q, Li S, Du X-S, Yang S, Chai J, Jin R. Atomically Tailored Gold Nanoclusters for Catalytic Application. *Angew. Chem. Int. Ed.* (2019) 58, 8291-8302.
15. Bao Y, Yeh HC, Zhong C, Ivanov SA, Sharma JK, Neidig ML, Vu DM, Shreve AP, Dyer RB, Werner JH, Martinez JS. Formation and Stabilization of Fluorescent Gold Nanoclusters Using Small Molecules. *J. Phys. Chem. C* (2010) 114, 15879-15882.
16. Kang X, Chong H, Zhu M. $Au_{25}(SR)_{18}$: the captain of the great nanocluster ship. *Nanoscale* (2018) 10, 10758-10834.
17. Das A, Liu C, Byun HY, Nobusada K, Zhao S, Rosi N, Jin R. Structure Determination of $[Au_{18}(SR)_{14}]$. *Angew. Chem. Int. Ed.* (2015) 54, 3140-3144.
18. Chen S, Wang S, Zhong J, Song Y, Zhang J, Sheng H, Pei Y, Zhu M. The Structure and Optical Properties of the $[Au_{18}(SR)_{14}]$ Nanocluster. *Angew. Chem. Int. Ed.* (2015) 54, 10, 3145-3149.

19. Assadollahzadeh B, Schwerdtfeger P. A systematic search for minimum structures of small gold clusters Au_n (n = 2-20) and their electronic properties. *J. Chem. Phys.* (2009), 131, 064306.
20. Bertorelle F et al. $Au_{10}(SG)_{10}$: A Chiral Gold Catenane Nanocluster with Zero Confined Electrons. Optical Properties and First-Principles Theoretical Analysis. *J. Phys. Chem. Lett.* (2017) 8, 1979-1985.
21. Yu Y, Yao Q, Chen T, Lim GX, Xie J. The Innermost Three Gold Atoms Are Indispensible To Maintain the Structure of the $Au_{18}(SR)_{14}$ Cluster. *J. Phys. Chem. C* (2016) 120, 38, 22096-22102.
22. Jiang D, Tiago ML, Luo W, Dai S. The "Staple" Motif: A Key to Stability of Thiolate-Protected Gold Nanoclusters. *J. Am. Chem. Soc.* (2008) 130, 9, 2777–2779.
23. Wille LT, Vennik J. Computational complexity of the ground-state determination of atomic clusters. *J. Phys. A* (1985), 18, 8, L419-L422.
24. Kirkpatrick S, Gellat CD, Vecchi MP. Optimization by Simulated Annealing. *Science* (1983), 220, 4598, 671-680.
25. Wales D, Doye J. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J. Phys. Chem. A* (1997) 101, 5111-5116.
26. Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks* (1995) 4, 1942-1948. IEEE.
27. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.* (2007) 39, 459-471.
28. Morris JR, Deaven DM, Ho KM, Wang CZ, Pan BC, Wacker JG, Turner DE. Genetic algorithm optimization of atomic clusters. *Evolutionary Algorithms* (1999) 167-175. Springer, New York, NY.
29. Yadav BC, Kumar R. Structure, properties and applications of fullerenes. *Int. J. Nanotechnol. Appl.* (2008) 2, 1, 15-24.
30. Tlahuice-Flores, A. New Polyhedra Approach To Explain the Structure and Evolution on Size of Thiolated Gold Clusters. *J. Phys. Chem. C* (2019) 123, 17, 10831-10841.
31. Heydariyan S, Nouri MR, Alaei M, Allahyari Z, Niehaus TA. New candidates for the global minimum of medium-sized silicon clusters: A hybrid DFTB/DFT genetic algorithm applied to Si_n, n = 8-80. *J. Chem. Phys.* (2018) 149, 074313.
32. Artrith N, Urban A, Ceder G. Constructing first-principles phase diagrams of amorphous Li_xSi using machine-learning-assisted sampling with an evolutionary algorithm. *J. Chem. Phys.* (2018) 148, 241711.
33. Yang H, Wong MW. Automatic Conformational Search of Transition States for Catalytic Reactions Using Genetic Algorithm. *J. Phys. Chem. A* (2019) 123, 47, 10303-10314.
34. Szustakowski JD, Weng Z. Protein structure alignment using a genetic algorithm. *Proteins* (2000) 38, 4, 428-440.

35. Van Batenburg FHD, Gultyaev AP, Pleij CWA. An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *J. theor. Biol.* (1995) 174, 269-280.
36. Chung YG et al. In silico discovery of metal-organic frameworks for precombustion CO_2 capture using a genetic algorithm. *Sci. Adv.* (2016) 2:e1600909.
37. Shayeghi A, Götz D, Davis JBA, Schäfer R, Johnston RL. Pool-BCGA: a parallelised generation-free genetic algorithm for the ab initio global optimization of nanoalloy clusters. *Phys. Chem. Chem. Phys.* (2015) 17, 2104-2112.
38. Deaven DM, Ho KM. Molecular Geometry Optimization with a Genetic Algorithm. *Phys. Rev. Lett.* (1995) 75, 2, 288-291.
39. Xiao Y, Williams DE. Genetic algorithm: a new approach to the prediction of the structure of molecular clusters. *Chem. Phys. Lett.* (1993) 215, 1-3, 17-24.
40. Hartke B. Global Geometry Optimization of Clusters Using Genetic Algorithms. *J. Phys. Chem.* (1993) 97, 9973-9976.
41. Pearl J. *Heuristics: Intelligent search strategies for computer problem solving* (1984). United States.
42. Li XT, Yang XB, Zhao YJ. Geometrical eigen-subspace framework based molecular conformation representation for efficient structure recognition and comparison. *J. Chem. Phys.* (2017) 146, 154108.
43. Kuhn HW. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* (1955) 2, 83-97.
44. Goedecker S. Linear scaling electronic structure methods. *Rev. Mod. Phys.* (1999) 71, 1085-1123.
45. Hohenberg P, Kohn W. Density functional theory (DFT). *Phys. Rev.* (1964) 136, B864.
46. Mueller T, Hernandez A, Wang C. Machine learning for interatomic potential models. *J. Chem. Phys.* (2020) 152, 050902.
47. Mueller T, Kusne AG, Ramprasad R. Machine Learning in Materials Science: Recent Progress and Emerging Applications. In *Reviews in Computational Chemistry* (2016) 29, 186.
48. Shapeev AV. Moment Tensor Potentials: a class of systematically improvable interatomic potentials. *Multiscale Model Simul.* (2016) 14, 1153-1173.
49. Novikov IS, Gubaev K, Podryabinkin EV, Shapeev AV. The MLIP package: Moment Tensor Potentials with MPI and Active Learning. *Pre-print.* (2020) arXiv:2007.08555.
50. Gubaev K. *Machine-Learning Interatomic Potentials for Multicomponent Alloys* (Doctoral dissertation, 2019). Skolkovo Institute of Science and Technology, Moscow, Russia.
51. Goreinov SA et al. How to find a good submatrix. In *Matrix Methods: Theory, Algorithms And Applications: Dedicated to the Memory of Gene Golub* (2010), 247-256.

Biographical statement

I graduated from Johns Hopkins University in 2019 with a Bachelor of Science degree in Chemical and Biomolecular Engineering with a second major in Materials Science and Engineering. I started pursuing computational research in my senior year of college, when I joined the Mueller Research Group to conduct a self-motivated research project on the first principles modeling and computational design of conductive metal-organic frameworks as intercalation cathodes for alkali-ion energy storage. Prior to this, I was a research assistant at the Johns Hopkins Center for Nanomedicine in the Kannan Lab, where I worked with Dr. Joshua Porterfield (a PhD candidate at the time) to study the impact of sugar functionalization on the pharmacokinetics and biodistribution of dendrimer drug delivery vehicles. With the publication of this thesis, I have completed the requirements for my Master of Science degree in Materials Science and Engineering.