AARHUS UNIVERSITY

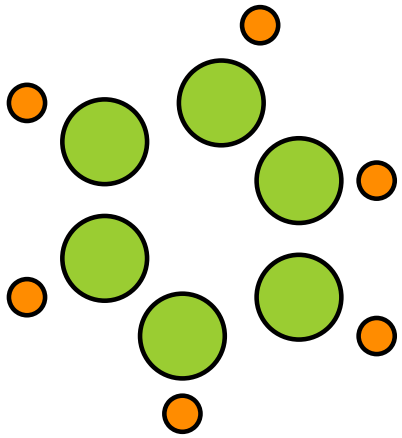
BACHELOR PROJECT

Attempting to improve a structure searching algorithm by using bundled atomic energies

Author
Tobias Rasmussen

Supervisor
Prof. Bjørk Hammer

June 15th, 2019



Abstract

The aim of this project is to create a method that can be used by the structure searching algorithm ASLA. This method should encourage building more coherent molecular structures and improve the performance of the algorithm. ASLA uses reinforcement learning to determine where to place a number of atoms in order to minimize the potential energy of the structure. In this project, ASLA must place 3 of both C- and H-atoms with 3 of both already placed as a template. The method designed uses the bonds of each atom in a structure to approximate its local energy, using it to calculate the global energy of the structure. The method proves successful at calculating expected local features of structures. This makes it possible to link the global feature vectors to the known DFT energy of the structure through a minimization problem using ridge regression. The method is able to predict most structure energies to within a tolerable degree: Predicting the energies of 500 random structures produced an R^2-value of 0.83 when comparing the estimate to the DFT energy. The energy prediction was implemented in a filter, which reduced the relative number of structures with isolated atoms to choose among with 72.3(58) %. The filter was then implemented into a new build-policy for ASLA and the performance of the altered algorithm was compared to its non-altered counterpart. The results indicate that the method significantly slows or hinders ASLAs structure search during early episodes, while failing to improve beyond the baseline performance during the remaining run time.

Resumé

Formålet med denne opgave er at skabe en metode, som kan bruges af struktursøgnings algoritmen ASLA. Metoden skal opfordre ASLA til at bygge mere sammenhængende strukturer og forbedre dens resultater. ASLA bruger *reinforcement learning* til at bestemme, hvor en række atomer skal placeres for, at de minimerer strukturens potentielle energi. I dette problem skal ASLA placere 3 C-atomer og 3 H-atomer, mens at yderligere 3 af hver slags allerede er placeret som en skabelon. Den designede metode bruger hvert atoms bindinger til at approksimere den lokale energi af atomet, som da bruges til at beregne den globale energi af strukturen. Metoden viser sig at virke til at bestemme de forventede lokale *features* af strukturer. Derfra er det muligt at udregne en sammenhæng mellem den globale *feature* vektor for en struktur og dennes udregnede DFT energi vha. et minimeringsværktøj kaldet *ridge regression*. Denne sammenhæng viser, at det er muligt at forudsige energien af de fleste strukturer ASLA bygger, indenfor en tolerabel usikkerhed: Forudsigelsen af 500 tilfældige strukturers energi gav en R^2-værdi på 0.83 når estimatet blev sammenlignet med DFT-energien. Denne energiforudsigelsesevne blev implementeret i et filter, som reducerede det relative antal strukturer med isolerede atomer, der kunne vælges imellem med 72.3(58) %. Dette filter blev da implementeret i ASLA som en ny byggepolitik og ydeevnen af denne ændrede algoritme blev derefter sammenlignet med dens uændrede modstykke. Resultatet indikerer, at metoden sænker eller hindrer ASLAs struktursøgning signifikant i de tidlige episoder, mens at dens ydeevne ikke formår at slå den uændrede algoritmes over den resterende køretid.

Preface

This project was written with 3rd year Bsc physics students in mind. Before starting this project, I have had a 5 ECTS numerical programming course in MATLAB 2 years prior. Parallel to this project, I took a 10 ECTS introductory programming course in Python, which is also the language the algorithm I worked on, ASLA, is written in.

Acknowledgements

I would like to thank my fellow student Frederik Fup Johansen, for assistance and good, technical discussions of this project.

Thank you to my girlfriend, Andrea Mølgaard Nielsen, who has been of great source of support and assistance during the whole project.

Finally, I would like to thank professor Bjørk Hammer, for the excellent guidance, for the exciting challenges, and for always having his door open to help during this project.

Contents

1 Introduction

When searching for new materials or new pharmaceutical drugs with desired medical properties, machine learning (ML) is a method that can be used to predict promising candidates among many possibilities to test. One example includes training a network with extensive amounts of data, i. e. known drugs and their properties, enabling the network to predict new drug indications [1].

Like with many other things in nature, mankind has let itself be inspired by the brain. This led to the development of Artificial Neural Networks (ANNs) in an attempt to model the processing capabilities of it [2].

1.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of ANN typically used in image processing [3]. CNNs are used for tasks such as labelling or predicting based on the image they are fed, used in e.g. facial recognition or for self-driving cars. The image is scanned by a kernel which output is passed through several hidden layers to an output layer, which can be anything from a single number to a vector to a matrix with the same dimensions as the input . A method called backpropagation is used to adjust the weights and biases of the hidden layers to improve the predictability of the CNN. [3]

1.2 Reinforcement Learning

Reinforcement Learning (RF) is an area of machine learning where a software agent attempts to maximize a reward based on the current state of some environment, choosing among a set of actions [4]. This kind of system is called a Markov Decision Process (MDP) and the goal is for the agent to find the optimal balance between exploration of the enviroment and exploitation of known actions that result in high reward [4].

The power of reinforcement learning has been demonstrated by the Google DeepMind-team and their AlphaGo Zero program which, taking no prior knowledge, beat their other champion-defeating program AlphaGo 100-0 in the board game Go [5].

2 ASLA

The Atomistic Structure Learning Algorithm (ASLA) aims to build atomic structures that optimize some property $\mathcal{P}$ [6]. ASLA is modelled as an MDP: The atoms it places on its board are converted to a reward, which ASLA will try to maximize [7]. ASLA will estimate where on the 'board' to place the next atom by analyzing the current board with a CNN, which outputs a value at each point on the board, indicating the estimated reward for putting the next atom there. An instance of ASLA (*agent*) works through 3 phases when building a new structure: The *building* phase, the *evaluation* phase and the *training* phase. An overview of each of these phases can be seen on figure 2.1.

In this project, ASLA will have one half of a benzene ring as a template and will search for optimal structures by placing additional 3 C and 3 H atoms. The property $\mathcal{P}$ that ASLA will try to optimize is in this case the potential energy. The optimal structure using a C_6H_6-setup is the benzene ring, resulting in a global minimum in potential energy of around $-70.17\,\mathrm{eV}$.

2.1 Building phase

During the building phase, the agent will use its current knowledge to estimate where to place atom t (a_t) given the current state of the 'board' (s_t), called a *grid*. The agent can only place atoms at discrete points on the grid, and at least one atom is prepositioned on the it as a starting point for the agent (a *template*) [7].

The agent awards each point on the grid some value (Q-value) between -1

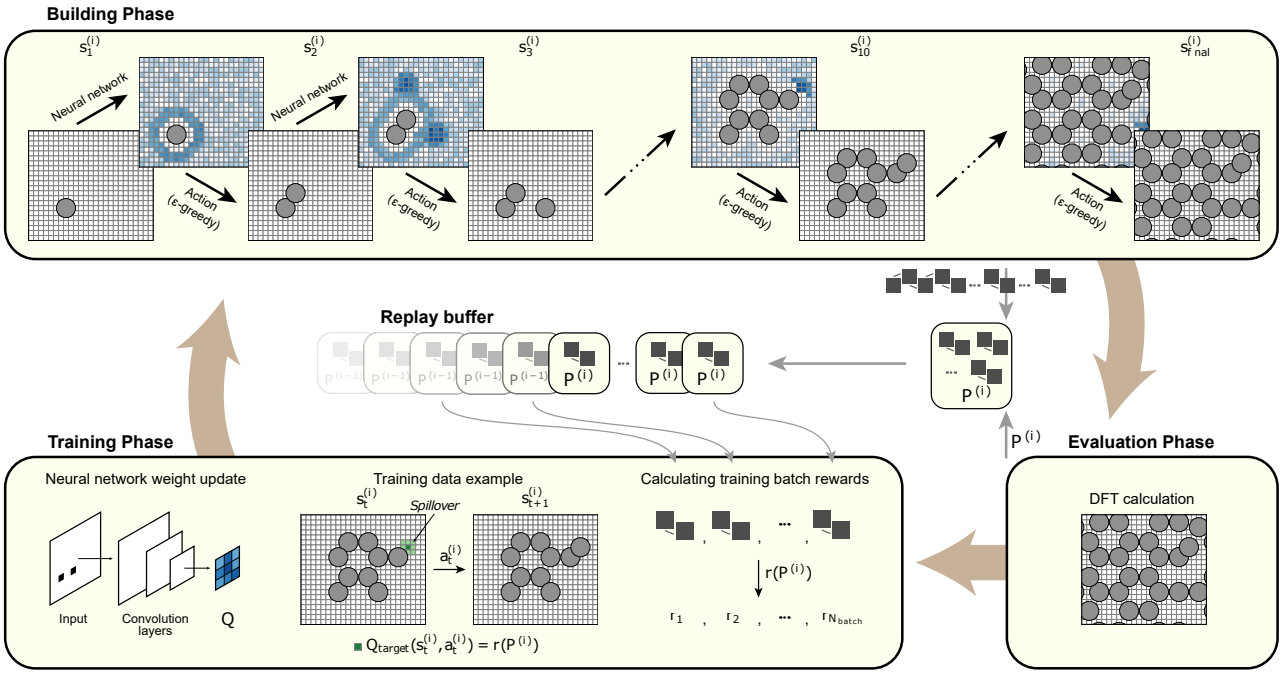


Figure 2.1: Overview of the three phases in ASLA. For each step t in the building phase of episode i, the current grid $s_t^{(i)}$ is inputted into the CNN, which outputs the Q-values of each gridpoint. Depending on the current build policy, an action $a_t^{(i)}$ is taken, placing an atom on the chosen grid point. This step is repeated until all atoms are placed. The completed structure $s_{final}^{(i)}$ is then passed onto the evaluation phase, where the property $\mathcal{P}^{(i)}$ is calculated (e.g. a DFT energy calculation). The results from the evaluation phase are then saved in the Replay Buffer, storing state-action pairs and their associated property. During the training phase, a number of state-action pairs are loaded from the replay buffer and used to train the CNN using backpropagation to improve future Q-value estimates. The illustration is taken from [6].

and 1 [7]. This value is the agents current estimate for the final reward of the complete structure if an atom is placed on that grid point. These Q-values are calculated using a CNN on the grid s_t [6]. Based on the current build-policy, a grid point is chosen and an atom is placed there. This step is repeated until there are no more atoms left to place (an *episode*).

To choose where to place the next atom, the agent has a *build policy* for each episode. The build policy used by ASLA is a modified ϵ-greedy policy with 2 parameters ϵ_ν and ϵ_0, both being small numbers less than 1 [7]. In the ϵ_ν-fraction of times, the agent chooses a random grid point among the

ν-fraction of highest Q-values. During the ϵ_0-fraction of times, the point is chosen randomly between *all* available points on the grid. During the remaining cases with probability $1 - \epsilon_\nu - \epsilon_0$ the point with the highest Q-value is chosen (*greedy* action). The parameters ϵ_ν and ϵ_0 implemented to encourage exploration of the grid in two different ways [7]. The completely random action with probability ϵ_0 attempts to encourage placing atoms at seemingly sub-optimal points in the hopes that the agent will find an even better structure than if it had chosen optimal points. The downside of this type of action is that getting out of a local minimum often requires several seemingly sub-optimal moves, which is not very likely to happen randomly. The ϵ_ν moves were introduced to counter this downside by avoiding grid points that are obviously bad choices (eg. too close to another atom) and instead choosing among points that the agent has found potentially rewarding [7].

2.2 Evaluation Phase

A structure is evaluated after it has been completed by the builder. The value of the target parameter $\mathcal{P}^{(i)}$ is then calculated using the property function $P(s^{(i)}_{final})$ [6]. The goal could be to make the thermodynamically most stable structure at $T = 0K$, in which case the parameter $\mathcal{P}$ would be the potential energy. The potential energy of a structure is calculated using density function theory (DFT), which is a computationally expensive quantum mechanical calculation [7].

After the DFT calculation, the energy, along the structure and each step of the building process is saved in the replay buffer.

2.3 Replay buffer

Previous state-action pairs are stored in the replay buffer. Each of these pairs has an associated property value that comes from the structure with the best property value that was built using that state-action pair [7]. During training, a batch of previous state action pairs are loaded from the replay buffer, the number of pairs being a parameter in ASLA [7]. Included in the training

batch is always the best structure ever made and some fraction of the most recent pairs as including these has a positive impact on the performance of the algorithm [7].

2.4 Training phase

During the training phase, a batch of state-action pairs $(s_t^{(i)}, a_t^{(i)})$ from previous episodes i are loaded from the replay buffer. Using these pairs, the CNN parameters are updated in a step called backpropagation. During this step, the root mean square error between the predicted Q-values and Q_{target}-values is minimized. The Q_{target}-values are calculated using the update rule [6]:

$$Q_{target}(s_t^{(i)}, a_t^{(i)}) = r(\mathcal{P}^{(i)}) \tag{2.1}$$

with $r \in [-1, 1]$ is the reward of episode i.

Furthermore, the training structures are rotated so that the CNN is taught rotational invariance. Lastly, to better simulate a continues potential energy surface, the Q-values from the backpropagation phase are smeared out to also affect nearby grid points , which results in a more smooth transition from higher to lower value coordinates and encourage exploring nearby grid points using the ϵ_ν-action from the ϵ-policy [6].

3 Bundled atomic energies using clustering

Using a feature vector $\mathbf{f}_{(i)}$ to describe each atom i in a structure, it is possible to estimate the local energy of each atom in the structure by grouping several local feature vectors linking the number and type of atoms in each group to the global energy of the structure [8].

3.1 Structure representation

As described in [8], molecular structures can be represented using vectors of integers.

A *local* feature vector, $\mathbf{f}$, describes the environment around each atom using a number of functions, of which there can be arbitrarily many

$$\mathbf{f}_i(\mathbf{r}_1, ..., \mathbf{r}_N) = \begin{bmatrix} f_i^1 \\ \vdots \\ f_i^L \end{bmatrix}. \tag{3.1}$$

Here i is an index over the N atoms in a molecular structure and the local feature vector $\mathbf{f}_i$ is calculated using L functions. Using one or few functions to describe the local feature vector might make different atoms indistinguishable from one another. For example, using a radially symmetrical function to describe the local environment of atom i will not be able distinguish configurations of 2 atoms located a distance r from atom i but with different bond angles [8]. In the previous case an angular function would make the configurations distinguishable.

The local feature vectors of several structures are then grouped into a number of *clusters*. This is done using either predetermined criteria or, if an unbiased method is desired, using a ML technique called clustering. Once the grouping criteria have been established, it is possible to construct a *global* feature vector of an entire structure from the number of atoms in each cluster for the given structure [8]

$$\mathbf{F}(\mathbf{f}_1, ..., \mathbf{f}_N) = [n_1, ..., n_c, ..., n_C]. \tag{3.2}$$

Here n_c is the number of atoms in cluster c with a total of C clusters.

3.2 Local energies

The energy of a structure is then proposed to be parameterized in the following way [8]:

$$E(\mathbf{r}_1, ..., \mathbf{r}_N) = \sum_{i=1}^{N} \boldsymbol{\epsilon}(\mathbf{f}_i) \approx \sum_{i=1}^{N} \boldsymbol{\varepsilon}_{c(i)} = \sum_{c=1}^{C} n_c \boldsymbol{\varepsilon}_c \tag{3.3}$$

where $\boldsymbol{\epsilon}(\mathbf{f}_i)$ is the local energy of atom i, which is defined by its feature vector, $\boldsymbol{\varepsilon}_c$ is the approximated energy shared by all atoms in cluster c, with $c(i)$ being the cluster index of atom i. Using this approximation, the local bundled energies of each cluster can be calculated using the following method [8]:

Let I enumerate a total of S structures. For each structure we have the relation

$$E_I = \sum_{c=1}^{C} n_{Ic}\boldsymbol{\varepsilon}_c, \tag{3.4}$$

in which $\boldsymbol{\varepsilon_c}$ is an unknown local energy as a function of feature-energy pairs $(\mathbf{F}_I, E_I)$ and n_{Ic} is the number of atoms in cluster c. This can be made into a matrix problem by looking at multiple structures

$$\begin{bmatrix} n_{11} & \cdots & n_{1C} \\ \vdots & \ddots & \vdots \\ n_{S1} & \cdots & n_{SC} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \vdots \\ \boldsymbol{\varepsilon}_C \end{bmatrix} = \begin{bmatrix} E_1 \\ \vdots \\ E_S \end{bmatrix}, \tag{3.5}$$

which can be rewritten as

$$\mathbf{X}\boldsymbol{\varepsilon} = \mathbf{E}. \tag{3.6}$$

The solution to equation (3.6) can be found by minimizing

$$\mathcal{E} = ||\mathbf{X}\boldsymbol{\varepsilon} - \mathbf{E}||^2, \tag{3.7}$$

which in some cases cannot be minimized [8]. This problem is overcome by introducing the parameter $\lambda > 0$ and using it to alter equation (3.7) using ridge regression. This altered equation is minimized by [8]

$$\boldsymbol{\varepsilon} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{E}, \tag{3.8}$$

with $\mathbf{I}$ being the identity matrix.

4 Motivation

A method to discourage ASLA from building structures that will have a bad energy reward is desired. This method should be able to, as a minimum, teach the algorithm not to build isolated single atoms away from the main structure.

I propose to implement a method able to estimate the energy of a given structure built by ASLA. The method will be based on making a representation of the structure as in (3.2) to calculate an estimate of the potential energy of the structure as seen in equation (3.5).

The hope is that this method will provide a computationally cheap estimate of the energy of the structure compared to the expensive DFT calculation. This could then be used to have the ASLA agent build many structures, estimating the approximate energy of each structure and then choosing randomly between the best e.g. 50% of the structures built. Thus, with a cheap computation, it would hopefully be possible to sort off many bad structures and instead use the expensive energy calculation on a structure that is worth exploring.

One could argue that simply disallowing the agent from building incoherent structures could easily be programmed into the algorithm. While this is true, the point of using reinforcement learning to search for structures is that the algorithm takes no prior knowledge of physics or chemistry before building. It is instead supposed to learn from its prior attempts and use those to improve its future attempts.

5 Clustering by atomic bonds

The aim of the simple structure evaluation is to discourage ASLA from choosing isolated atoms disconnected from the template structure, since these choices are not as stable as a single, coherent molecule. In order to solve this problem, the local feature vector of an atom will be a vector of integers, representing the number and type of atoms it is bonded to. Each composition of bonds corresponds to a cluster, which means that the bonds of an atom describes the cluster it belongs to.

The goal is to see this method of clustering, where the number and type of bonds of each atom determines its local energy, work and improve the performance of ASLA on this C_6H_6-structure search. Provided that it does, the aim would then be to design a more 'unbiased' local feature vector that doesn't directly determine what a good structure is and what is not. These local features would then be grouped using a clustering algorithm as described in chapter 3.

5.1 Structure representation

Each complete structure is analyzed using a feature function g, which for each atom in the structure will count all neighboring atoms within a 1.60 Å radius, which is then considered a bond between the two. Thus the local feature vector becomes

$$\mathbf{f}_i = g(\mathbf{s}^{(I)}_{final}) = \begin{bmatrix} n^{(i)}_C \\ n^{(i)}_H \end{bmatrix}. \tag{5.1}$$

Here i is the atomic index, $\mathbf{s}^{(I)}_{final}$ being the complete grid of structure I and n being the number of bonded atoms of type C or H.

The reason that the bonding range is 1.60 Å is that in the global energy minimum structure built with C_6H_6, benzene, the distance between some of

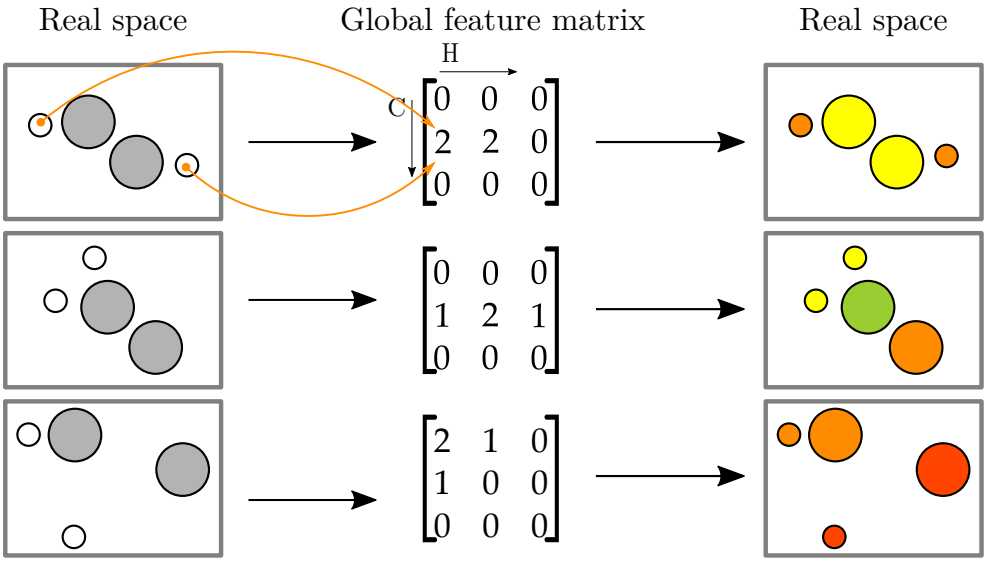


Figure 5.1: Visualization of how structures are represented in a feature matrix. For illustration purposes the structures only consist of C_2H_2. Each entry in the matrix corresponds to the number of atoms with some specific bonds. As an example with the top structure shown, both H-atoms have 1 C bond an 0 H bonds, which is shown in the matrix at row 2 column 1. After examining all atoms in the structure, the matrix is flattened into a vector to estimate the local energy of each cluster. The structures can then be colored by the total number of bonds of each atom.

the neighboring C-atoms on the grid is $\sqrt{10}/2\,\text{Å} \approx 1.58\,\text{Å}$. All of these bonds should be found when analyzing the structure to give an accurate representation of why the structures is so stable. This means that all C-atoms will have the same number of bonds and likewise for the H-atoms in the structure. This would not be the case for a lower bonding range of e.g. 1.5 Å, which would not represent benzene as a very stable structure to build.

The number and type of neighbors of one atom will be represented by a list of integers, one for each atom type in the structure. This list of integers is the coordinate of an entry in a tensor with dimensions $N_t + 1$, with N_t being the number of atoms of type t in the structure (e.g. for C_2H_2 it is a 3×3-matrix as seen in figure 5.1). Each entry in the tensor counts the number of atoms in the structure that match their coordinates, with each entry counting as a seperate cluster. The tensor is then flattened to create the global feature

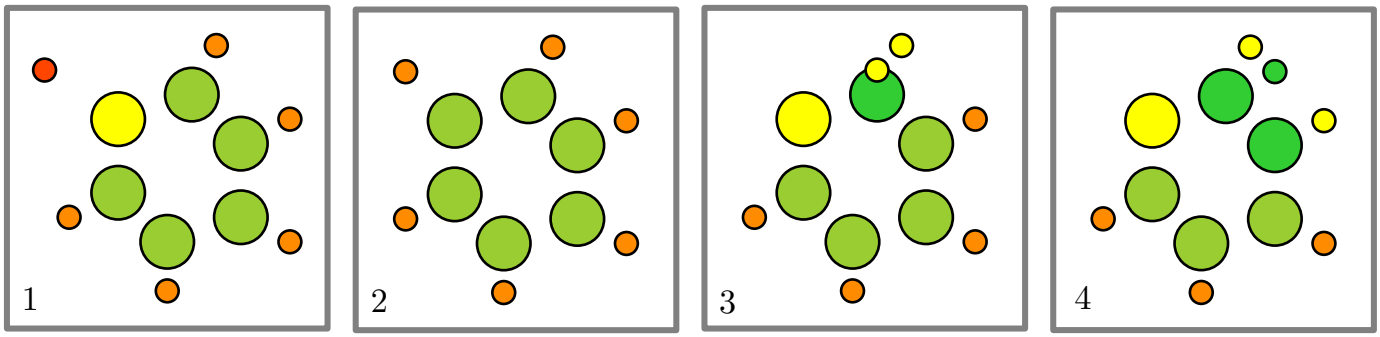


Figure 5.2: Demonstration of local features for each atom of a completed structure. **Frame 1:** The red H-atom is too far away from the rest of the structure and is considered isolated by the feature function g. **Frame 2:** Moving the H-atom to the right will bring it close enough to the remaining structure for g to consider it a bond. **Frame 3-4:** Further moving the atom to the right will also change the representation (color) of the other atoms as they become bonded or bonds break with this atom. Note in frame 3 that this representation only has a maximum distance and so it doesn't penalize placing atoms too close to each other.

vector $\mathbf{F}$ of that structure. When building $C_{N_C}H_{N_H}$-structures we get the following representation of the structure

$$\mathbf{F}_I(\mathbf{f}_1, ..., \mathbf{f}_N) = \begin{bmatrix} n_{0,0} & \cdots & n_{0,\,(N_H+1)} \\ \vdots & \ddots & \vdots \\ n_{(N_C+1)\,,0} & \cdots & n_{(N_C+1)\,,(N_H+1)} \end{bmatrix} \sim \begin{bmatrix} n_{00} \\ n_{01} \\ \vdots \\ n_{N_H N_C} \end{bmatrix} \tag{5.2}$$

Atoms are colorized according to the total number of bonds. The colorscheme for each number of bonds is: 0 - Red, 1 - Orange, 2 - Yellow, 3 - Yellowgreen, 4 - Limegreen. No structures with 5 or more bonds are depicted in this project, but the color is set to be darker shades of green for each bond above 4. Atom coloring is depicted figure 5.1.

5.2 Results

Using the demonstrated ability to count the bonds of each atom, it is then tested if it is possible to predict the energy of completed structeres by clustering local energies according to bonds as in equation (3.6). Each entry in the global feature vector $\mathbf{F}_I$ from equation (3.5) represents the number of atoms

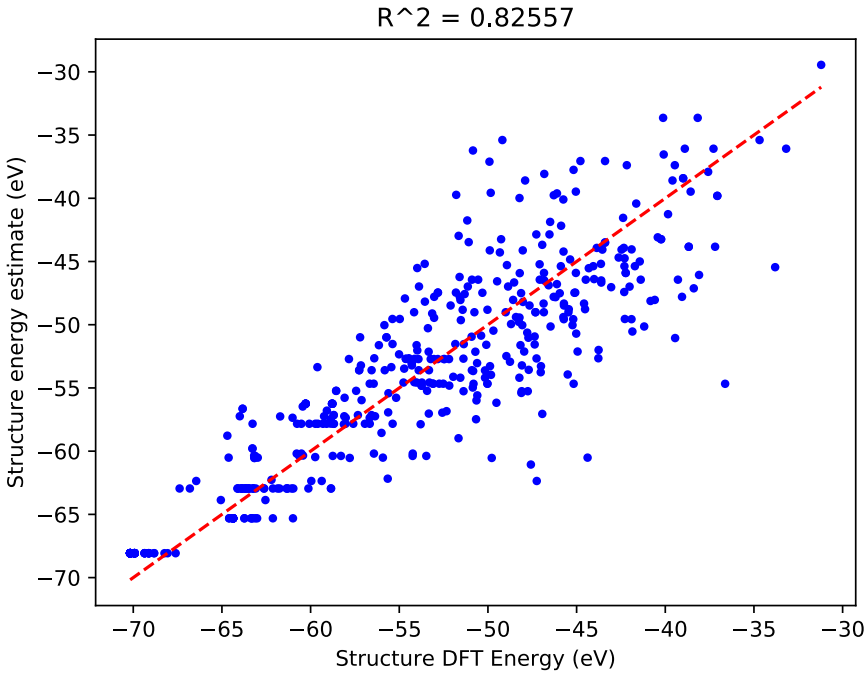


Figure 5.3: Demonstration of how estimating the potential energy of structures built by ASLA can be done using local energies dependent on the number and type of bonds of each atom in the structure. The 500 blue points come from a random sample of builds by an agent that found the global energy minimum of $-70.17\,\mathrm{eV}$. These structures have all had their potential energy estimated by a DFT calculation, which is their position on the first axis. Their position on the second axis is determined by their estimated energy. The red line indicates the points where the DFT energy and the estimate are equal.

with a specific number and type of neighbours in structure I.

As can be seen on figure 5.3 this method of using ridge regression to estimate the local energy of each bond-environment can successfully estimate the approximate energy of C_6H_6-structures built by ASLA. By calculating the features of 500 random structures from an agent that found the global minimum, an estimate for the local energies of the different bond-enviroments ε was calculated. Using these energies, the energy of another 500 random structures were calculated.

Using this method to estimate the energy of builds, it is then attempted to

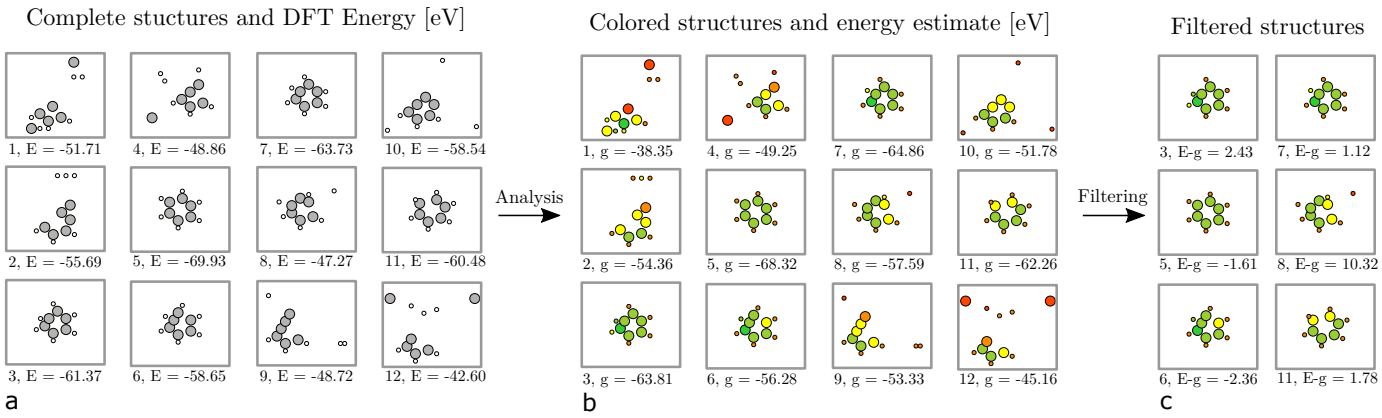


Figure 5.4: Demonstration of how the proposed filtering algorithm works. The estimated energy of each cluster is calculated from 500 structures. **a:** The agent builds N structures that are saved in a list. In this example the DFT calculation has been done to show the true target property of each structure. **b:** A feature vector is then calculated for each structure and using this, the energy of the complete structure is estimated. **c:** The filtering then consists of choosing the $N/2$ structures with the lowest estimated energy.

implement an algorithm that can choose e. g. the 50% best structures from a list. An illustration of this algorithm can be seen on figure 5.4. An enlarged version can be seen on figure A2 in the appendix.

Using a number of complete structures and their calculated DFT energies, the energies of the different bond-clusters is calculated. Then, the agent is asked to build N structures, making no expensive DFT calculation for any of them. The energy of each structure is then predicted, estimated from the found bonds of each atom in the structure. The structures are then sorted from lowest(most stable) to highest(least stable) with respect to their estimated potential energy. Some percentage of the structures with lowest energy are then kept. One structure is then chosen randomly among the remaining structures as the one to have a DFT energy calculation done. This chosen structure will likewise have all building steps and final structure saved in the replay buffer for training in future episodes.

To demonstrate the ability of the filter to remove structures with isolated atoms, 5 different estimates to ϵ were made. Each had the features from 500 structures made from an agent that had found the global minimum. Each of

these $\boldsymbol{\varepsilon}$ then predicted the energy of $N = 50$ random structures also made by the agent and recorded the number of structures containing isolated atoms, n_{pre}. After filtering, keeping only the 25 structures with lowest potential energy, the number of structures containing isolated atoms, n_post, was recorded. The relative change $(n_{post} - n_{pre})/n_{pre}$ is then calculated for the run, and each run is repeated 10 times. In total, for 5 different $\boldsymbol{\varepsilon}$ each make 10 runs, in each run building 50 structures and choosing the best 25. This resulted in a relative reduction of isolated structures of 72.3(58) %. On average, almost 3/4 of the structures with isolated atoms are removed by filtering using this method!

6 Implementation into ASLA

In this chapter, it is attempted to implement the structure filter from the previous chapter into the Atomistic Structure Learning Algorithm (ASLA).

6.1 Hyperparameter search

Before implementing the filter, it was firstly attempted to determine the best ratio of exploration/exploitation for the 'baseline' ASLA, which the filter will attempt to improve. Recall from chapter 2 that the parameter ϵ_0, used during the ϵ-policy builds, determines the chance for an action to randomly place an atom on the grid. To determine the optimal value of ϵ_0, 25 instances of ASLA were initiated with a run time limit of either 5000 episoded or maximum time limit of 24 h. After all instances had finished running, it was checked how many of the instances that had found the global minimum structure. The random chance was varied between 10% up to 70% and the results can be seen on figure 6.1. From the figure it can be seen that $\epsilon_0 = 4/10$ is the most fruitful setting, which is then used for all future instances of ASLA.

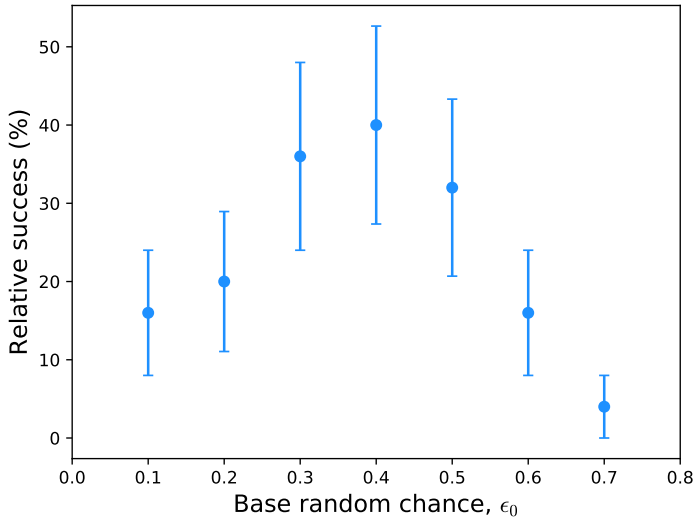


Figure 6.1: The chance to make a completely random action during ϵ-policy builds, ϵ_0, was varied between 1/10 through 7/10. After initiating 25 agents with each value of ϵ_0, the number of agents that found the global minimum solution were counted. The figure shows that the setting $\epsilon_0 = 4/10$ proved the most fruitful.

6.2 Method

To implement the filter into ASLA, a new build-policy is made, which will be called the ϵ-loop policy. Furthermore, ASLA will calculate and save the global feature vector of the latest 500 structures along with their DFT energies, so that each episode above 500 will have an $\boldsymbol{\varepsilon}$ to estimate future structure energies. Then, whenever the build-policy is ϵ, the new ϵ-loop policy will 80% of the time be chosen as the build policy instead. During the ϵ-loop policy, the agent will be asked to build 50 structures with the ϵ policy, but not complete the expensive DFT calculation for the structure. Each of the returned builds are saved and their energies are estimated using the current $\boldsymbol{\varepsilon}$. The structures are then sorted, and a final candidate is chosen among the 50% of candidates with lowest estimated energies. The final structure is then treated as the only structure built by the agent that episode, and DFT energy calculation and training is done as usual.

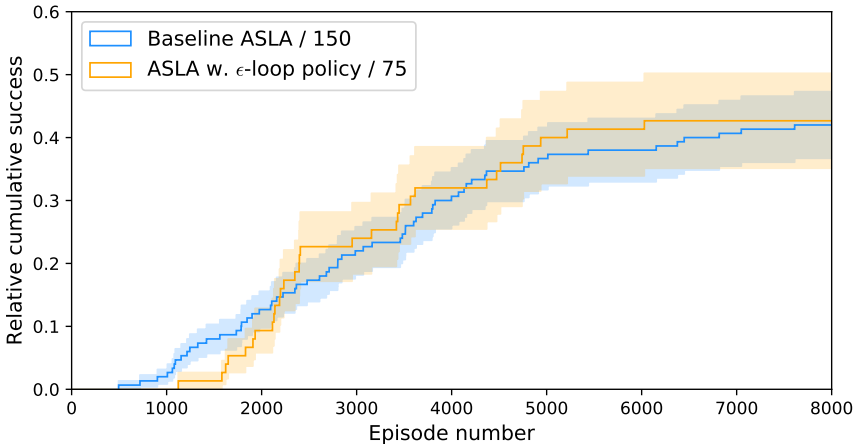


Figure 6.2: Resulting S-curve comparing the cumulative success of each type of agent to the episode in which the solution is found. The agents have a limit of 10,000 episodes or 24 h. As can be seen on the figure, the ϵ-loop agents all find the global solution significantly slower than the baseline. However, the ϵ-loop agents end up with the same fraction of agents finding the global solution as the baseline. Since the uncertainties of the 2 types of agents overlap after episode 2500, we cannot claim that the performance of one agent is significantly different from the others after that episode.

6.3 Results

After running both baseline ASLA agents and ASLA agents with the ϵ-loop policy, the number of agents that found the global minimum solution were counted. The performance of both algorithms are compared and illustrated in figure 6.2. As can be seen on the figure, the performance of the ϵ-loop agents are equal to if not slightly worse than their baseline counterparts. Due to the uncertainty of the performance results, the two algorithms cannot be said to be significantly different from each other, and so, it cannot be claimed that the ϵ-loop implementation improves the performance of ASLA. Due to the ϵ-loop agents being significantly slower in finding solutions, it seems that the implementation of energy estimation during the early episodes of building will hinder the structure search, while at later episodes it will simply match

the performance of the baseline ASLA.

7 Discussion

One significant drawback of the energy prediction method is its dependency on the structures it is 'trained' with. Due to the way the ridge regression works, (i. e. it is a minimization problem), the ability to predict energies that are outside the range of trained energies seems unlikely. This is a problem, since the first structures built by ASLA all have high energies and a small energy span early on, so the method cannot easily distinguish good from bad structures during the early builds. This problem is illustrated on figure A1 in the appendix.

This drawback can possibly explain the lack of performance of the implemented model into ASLA. As can be seen on figure 6.2, the ϵ-loop agents are significantly slower than their baseline ASLA counterparts: Almost 10% of baseline agents have found the global minimum structure before the first ϵ-loop agents begin to find it. This could possibly be due to the poor energy prediction filtering away potentially rewarding structures in the early episodes.

It should also be noted that the atomic filter's ability to remove structures with isolated atoms is likewise reduced during early episodes, due to the uncertainty in energy prediction and thereby the sorting of structures. When trained using random structures from up to episode 1500, the relative reduction was instead 66.1(42) %.

Another source of error regarding energy prediction is the fact that the structure representation doesn't penalize or mark atoms for being unfavorably close to on another. Instead, all atoms within the set bonding range are counted in the same cluster. As seen in a Lennard-Jones potential, too small interatomic distances result in a, potentially large, repulsion rather than attraction between the atoms. This could possibly have been avoided by using

another method to represent the atom, as is discussed later. With regards to the performance of the algorithm, this might also explain some of the slow performance seen in the early episodes. In the early episodes, ASLA tends to place atoms too close to one another, and due to the way the structure representation is currently designed, these close atoms are simply counted as bonds, which when estimating the ε of several structures, will result in clusters with many neighbors having a high energy, making it unfavorable to choose structures with many bonds in the filter.

A particular example of another representation would be to describe each atom in a structure by the sum of distances to each other atom of each type in the structure. For a number of structures, these features could then be clustered using some clustering algorithm (e.g. K-means). This approach would be much more general than the current method of guessing that atomic bonds are good molecular descriptors.

8 Conclusion

In an attempt to discourage ASLA from building structures with isolated atoms, a representation of ASLA structures was designed. This representation was based on the bonds of each atom in the structure, namely the number and type of atoms the atom was bonded to. This representation was shown to work in figure 5.2, where a single atom was moved across a colorized structure, which demonstrated how atoms around it had their features updated as the atom came within or left bonding distance.

Using the assumption from (3.3), it was demonstrated that the structure representation could be used to predict energies of new structures based on features of structures already built by ASLA. The ability to predict energies is illustrated on figure 5.3, where the estimated energy was plotted against the DFT energy, with a resulting R^2-value of ≈ 0.83. This was then used to

implement a filter, where the estimated energy of proposed structures built by ASLA were used in an attempt to filter away structures containing isolated atoms. The relative reduction in structures containing isolated atoms among the proposed builds were reduced by 72.3(58) % by choosing the half of the structures with the lowest estimated energies.

This filtering method was then implemented into ASLA by introducing a new building policy, ϵ-loop. The performance of these new agents were compared to that of the unaltered baseline agents in figure 6.2. The results indicate that the implementation hinders the structure optimization during the early episodes while not increasing the fraction of agents that find the global minimum compared to the baseline agents. Among the reasons that explain this result are that the energy predictions are a great deal more uncertain during early episodes which makes it harder to filter away bad structures. It is also suspected that taking atoms too close to each other into account in the structure representation would have had positive effect on the performance.

Bibliography

[1] Michel Dumontier Remzi Çelebí, Özgün Erten. Machine learning based drug indication prediction using linked open data. *CEUR-WS*, vol. 2042, 2017. Accessed 14-6-19 via `ceur-ws.org/Vol-2042/paper30.pdf`.

[2] Raúl Rojas. *Neural Networks A Systematic Introduction*. Number ISBN: 978-3-642-61068-4. Springer, Berlin, Heidelberg, 1996. `doi.org/10.1007/978-3-642-61068-4`.

[3] Wikipedia. Convolutional neural networks. Accessed 14-6-19 via `en.wikipedia.org/wiki/Convolutional_neural_network`.

[4] Wikipedia. Reinforcement learning. Accessed 14-6-19 via `en.wikipedia.org/wiki/Reinforcement_learning`.

[5] Karen Simonyan Ioannis Antonoglou Aja Huang Arthur Guez Thomas Hubert Lucas Baker Matthew Lai Adrian Bolton Yutian Chen Timothy Lillicrap Fan Hui Laurent Sifre George van den Driessche Thore Graepel Demis Hassabis David Silver, Julian Schrittwieser. Mastering the game of go without human knowledge. *Nature*, 355, 2017. doi: `doi.org/10.1038/nature24270`.

[6] Mathias S. Jørgensen, Henrik L. Mortensen, Søren A. Meldgaard, Esben L. Kolsbjerg, Thomas L. Jacobsen, Knud H. Sørensen, and Bjørk Hammer. Atomistic structure learning. *arXiv e-prints*, page arXiv:1902.10501, Feb 2019.

[7] Henrik Lund Mortensen. Structure optimization using reinforcement learning. Progress report, Aarhus University (unpublished), April 2019.

[8] Søren A. Meldgaard, Esben L. Kolsbjerg, and Bjørk Hammer. Machine learning enhanced global optimization by clustering local environments

to enable bundled atomic energies. *The Journal of Chemical Physics*, 149(13):134104, 2018.

Appendix - Extra figures

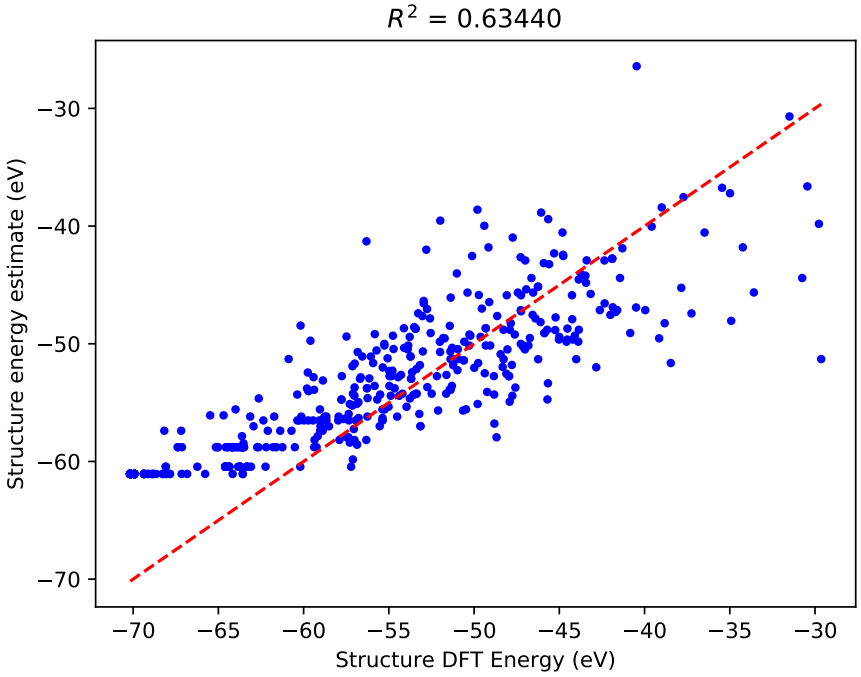


Figure A1: This figure illustrates the possible problem that ϵ-loop agents in ASLA may have encountered. The ε has been fitted using 500 structures from episodes up to 1000. It has then predicted the energy of 500 structures from episode 1001 and onwards, chosen randomly. Note that DFT energies below ≈ -62 eV will be assigned that energy. The energy range of the structures that are used for training seems to determine the prediction range. This could be a problem, since the structure filter works by sorted structures after predicted energies.

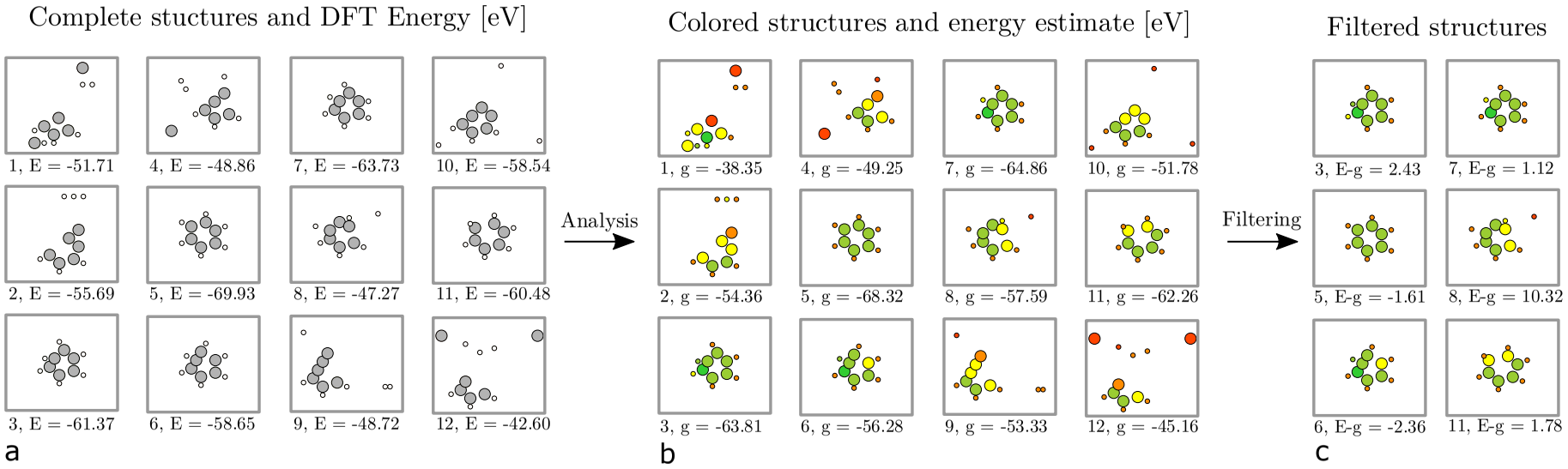


Figure A2: Enlarged version of figure 5.4.