

# **Using a Neural Network to Explore the Conformational Space of MurD**

by **Louis Sayer**

supervised by **Dr Matteo Degiacomi**

April 2022

Summative report submitted for the degree

Master of Physics (MPhys)

at the

Centre for Material Physics

Department of Physics



## Abstract

Determining the transition states of proteins is difficult using Molecular Dynamics techniques and even harder with experimental methods. This is owed to the relatively high potential energy of transition states and therefore low chance of being sampled, as described by Maxwell-Boltzmann statistics. But transition states are important in understanding how proteins interact with their environment and can strongly inform protein and drug design. In this project a generative convolutional neural network, developed by the Degiacomi group, is used to study the conformational space of a bacterial ligase, MurD which has a mass of 47kDa. MurD has two native states it transitions between: an open state and a closed state in which it would naturally contain a ligand. MD simulations, run by the same group, provide the input conformations. By creating a conformational space we have direct access to the transition states of this protein and can create unseen conformations. A loss function is used which aims to minimise the difference between input and output conformations while maintaining a low potential energy. Different parameters of the neural network are tested to find the optimal parameters to reconstruct and create conformations, as measured by their similarity to input structures (RMSD) and statistical potential energy (DOPE score). We find that the original parameters, with the exception of one, give the best results of these two measures and are also more time efficient.

# Contents

|     |  |    |
|-----|--|----|
| 1   | Introduction . . . . .   | 3  |
| 1.1 | Proteins . . . . .   | 3  |
| 1.2 | Molecular Dynamics . . . . .                                     | 6  |
| 1.3 | Machine Learning . . . . .                                       | 6  |
| 2   | Materials and Methods . . . . .                                  | 8  |
| 2.1 | Molecular Dynamics . . . . .                                     | 8  |
| 2.2 | Overview of Molearn . . . . .                                    | 9  |
| 3   | Project Objectives . . . . .                                     | 12 |
| 4   | Results and Discussion . . . . .                                 | 12 |
| 4.1 | Latent Space . . . . .   | 12 |
| 4.2 | Atoms selected in the Conformations . . . . .                    | 14 |
| 4.3 | Effect of Kernel Size . . . . .                                  | 18 |
| 4.4 | Effect of Latent Space Dimension . . . . .                       | 21 |
| 4.5 | Comparison of Output Conformations to Experimental Intermediates | 23 |
| 4.6 | Training Times . . . . .   | 26 |
| 5   | Conclusion and Future Work . . . . .                             | 27 |
| 6   | Acknowledgements . . . . .                                       | 28 |

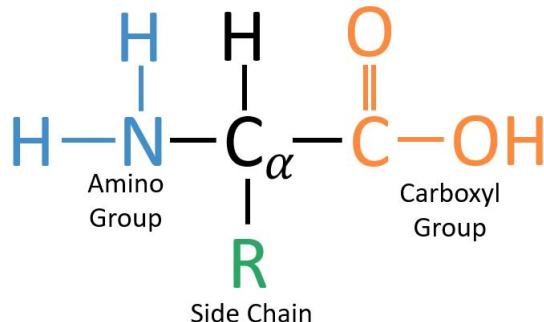
## 1 Introduction

### 1.1 Proteins

It is difficult to overstate the importance of proteins in life. All life is composed of cells and proteins are involved in virtually every cellular process. They participate in a plethora of roles including metabolism, cell signalling, structure and immunity [1]. Without them, cells would be rendered unfunctional. Proteins are important in so many reactions because of their ability to bind to other molecules. It is therefore the structure of a protein which determines its function. Their study makes them useful in many applied settings, such as drug design and industrial enzymes. One application of protein design is that in vaccine development where one could create protein cages as biological scaffolds for immunogens [2].

Proteins are coded for in structure by DNA. Three consecutive nucleotides in a DNA strand correspond to a codon which specifies an amino acid. Long chains of amino acids

are called polypeptides, which may fold (often with other polypeptides) in specific conformations to form a protein. The structure of an amino acid can be seen in Fig. 1.



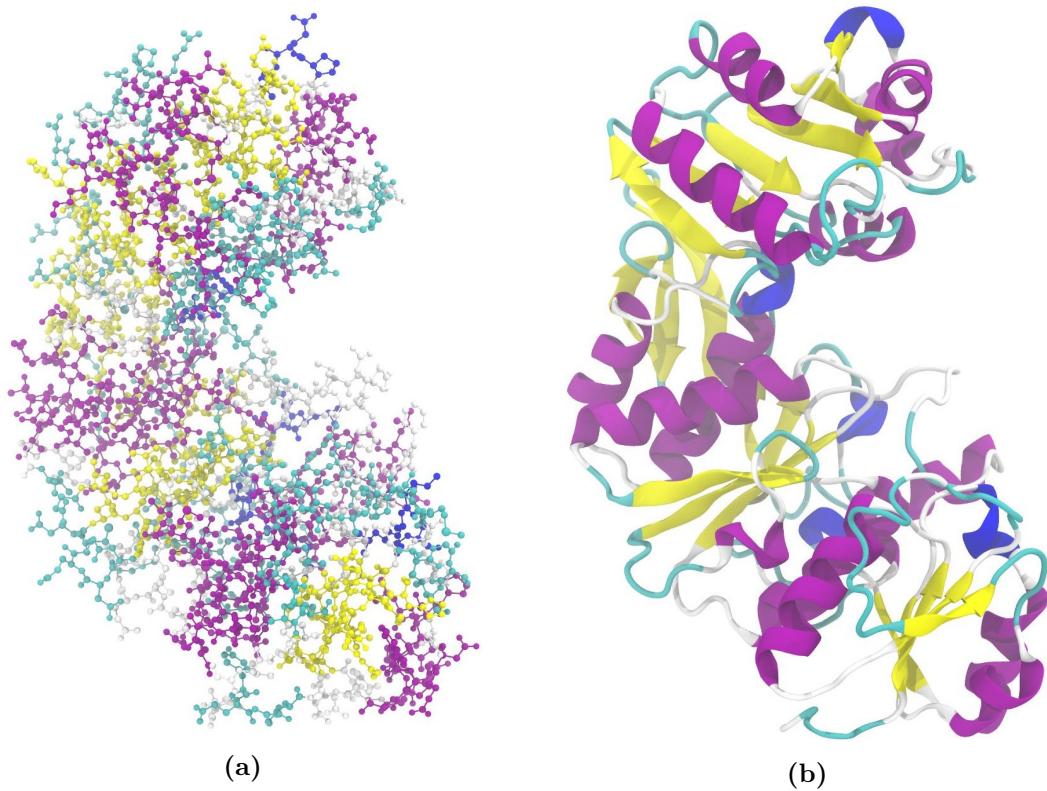
**Figure 1:** The structure of an amino acid consisting of a common amino group, carboxyl group and a variable sidechain. The sidechain is bonded to the C $\alpha$  and the first carbon in the sidechain is called the C $\beta$ .

The amino acid chain is called the primary structure and folds into common configurations such as alpha helices and beta sheets (secondary structure). The final conformation is called the tertiary structure. The secondary and tertiary structures of the protein in this work, MurD can be seen in Fig. 2. These images were produced in VMD, as were all other protein figures in this work [3].

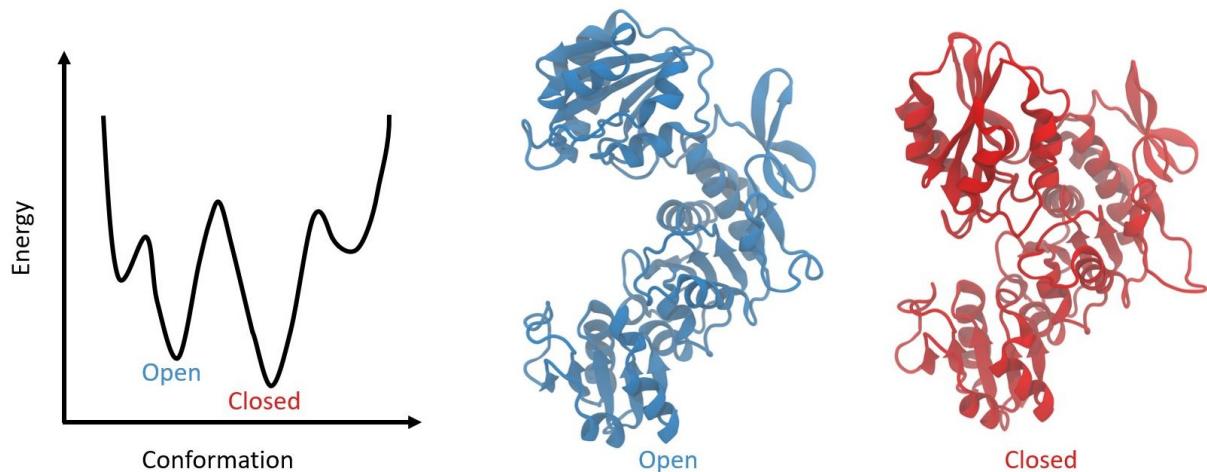
Understanding protein structure is important because ligand binding is specific [2]. Ligands are ions or molecules which bind to proteins to form a complex. Their binding sites have specific shapes and the atoms forming a site will determine the non-covalent bonds that can be made with a ligand [4]. Proteins are not rigid molecules however and their structure depends on environmental variables such as pH, temperature, pressure, an oxidising environment and the presence of other molecules [5]. Proteins change shape according to their environment and each conformation can be represented as a point in a high dimensional Euclidean space. This landscape can be imagined as many valleys and mountains representing areas of high and low potential energies. The protein is more likely to be found in areas of low potential but as it interacts with its environment it loses and gains energy. Fig. 3 shows the 2D version of this.

X-ray crystallography which is at present the most common technique for observing proteins requires them to be trapped in crystals [6]. The conformations that are viewed are those of the lowest energy and so we require Molecular Dynamics (MD) and machine learning to sample the whole conformational space. These tools can help reveal cryptic binding sites which would not be seen with experimental techniques. Another application of machine learning is being able to engineer proteins which can change states at different speeds and therefore change the rate at which reactions happen [7].

MurD is a ligase in E.coli, which is involved in the pathway of peptidoglycan biosynthesis [8].



**Figure 2:** The tertiary structure of MurD, shown with bonds and atomic spheres in (a) and with secondary structure components in (b). The colouring shows the presence of different secondary structure sections such as  $\alpha$  helix in purple,  $\beta$  sheet in yellow, 3-10 helix in dark blue and a turn in light blue.



**Figure 3:** The potential energy of a protein for different conformations. For this case the conformations are 1D but the full dimensionality is actually as many DOF in the conformation. Transition states are situated between minima and are less likely to be sampled. For MurD the more stable conformations are the open and closed states.

## 1.2 Molecular Dynamics

Molecular dynamics simulations provide the conformations which are used to train the neural network. Experimentally determining conformations using techniques such as X-ray crystallography, NMR and electron microscopy is expensive, time-consuming and will only sample the most energetically favourable conformations. MD takes an initial, known atomic arrangement of atoms and evolves it according to bonded and unbonded forces between the atoms, known as a force-field [9]. This way, an ensemble of conformations can be produced for the protein. Ideally, the simulation will take place for an infinite time so that all conformations are sampled, as stated by the ergodic theorem [10]. The probability of a conformation being sampled is inversely proportional to its potential energy and therefore not all of the conformational space is represented.

The simulation time should also match the time it takes for processes to occur within the system. This motivates for having a long time step over which the numerical integrals are computed so that fewer calculations are required. However, two problems arise with this. One is that because of Nyquist's theorem, the timestep must be at least half the period of the fastest vibration in order that information is not lost on the accurate location of the atoms [11]. For continuous potentials an overly large timestep can result in locating the atoms as if they had greater frequencies. Another problem results from truncation error which can lead to atoms overlapping and their potentials rapidly increasing.

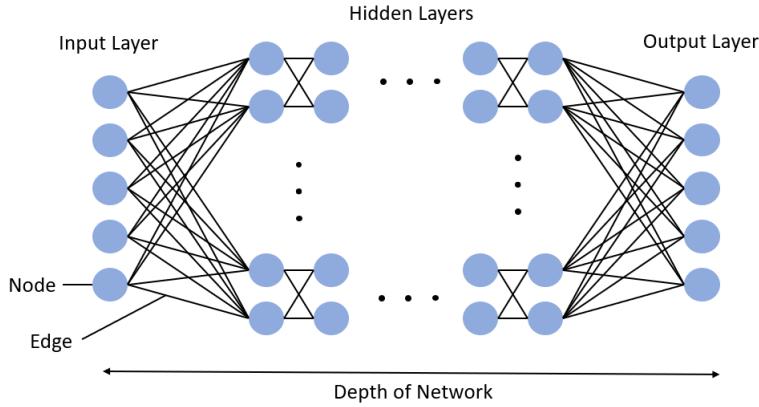
For MurD the potential energies of the intermediate states are much higher than those of the open and closed states. Even for simulations with millions of conformations only a handful may be of an intermediate state. It is possible to bias simulations such that rarer events occur, or to match experimental structures such as experimentally determined intermediate states. However, this only increases the likelihood of higher energy states being found. Simulations still have to be carried out for long periods of time, often days or weeks.

Machine learning offers two solutions to these problems. With the method used in this network intermediate states can be directly sampled from a conformational space which can be compared for their plausibility against known intermediates. This allows us to generate unseen structures and consider how a protein moves between states. Machine learning also takes considerably less time with a full training run taking as little as five-six hours.

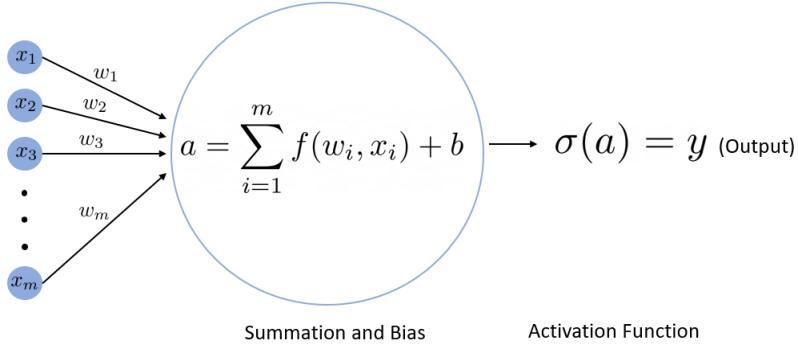
## 1.3 Machine Learning

Machine learning is a form of artificial intelligence which uses complex algorithms to understand and make predictions about data. Neural networks belong to this class of algorithms, which are designed with inspiration from biological neural networks found in the brains of animals. The key component of a neural network is a node which mimics neurons in the brain. A neural network has several layers of nodes which are connected to other nodes with edges. The data is fed through an input layer where it passes through several layers to the output layer where it may then make a classification or prediction.

The general structure of a neural network can be seen in Figs. 4 and 5.



**Figure 4:** The structure of a neural network featuring an input layer, one or more hidden layers and an output layer.

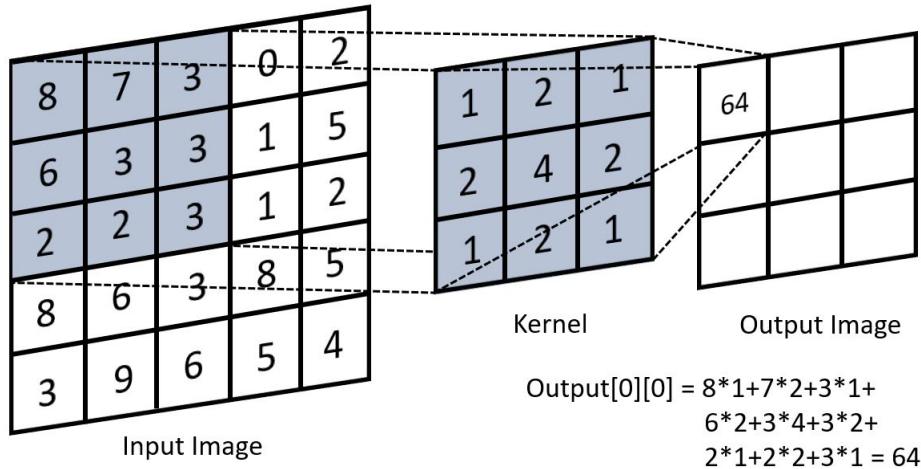


**Figure 5:** The process of calculating the output of a node given the input from all the connected nodes.  $x_i$  and  $w_i$  denote the  $i^{th}$  node and weight. It involves calculating the transfer function (summation and bias) which is fed to an activation function which is commonly a sigmoid and hence non-linear.

The output of the neural network is normally fed to a loss function which needs to be minimised such that the output is as close as possible to the desired result. For example, in number image classification where images of numbers are fed to the network one wants to minimise the difference between the number predicted by the neural network and the actual number input [12]. The weights of the neural network, i.e. the parameters of each neuron, are changed by attempting to minimise the loss function in a process called back-propagation [13]. It uses gradient descent. Starting from the output layer the gradient of a weight is calculated with respect to the loss function. The magnitude and sign of the gradient determines how the weight and bias will be altered. The weights of preceding layers are changed iteratively in this way until all the weights have been updated.

There are many different types of neural networks. The architectures so far shown belong to artificial neural networks (ANN) with ‘classical neurons’. In this project we used a convolutional neural network. A CNN uses kernels to pass over the dataset which

perform operations on the elements to reduce the dimensionality of the dataset [14]. This is often used in image processing where the pixels are operated on by an image filter (kernel) and can be seen in Fig. 6.



**Figure 6:** The process of calculating one of the pixel values in the output image. A kernel, with weights shown in the cells passes over the input image. A dot product is calculated between the values of the pixels in the input image with the weights in the kernel. The kernel sweeps across the whole image reducing the resolution from  $5 \times 5$  to  $3 \times 3$ . If the kernel operated on the output image the resolution could be reduced to 1. The number of times the resolution is reduced is the number of layers of the network. The number of kernels per layer is the depth or number of channels.

Depending on the way in which the neurons are arranged networks can either be generative or classifiers. Generative networks aim to generate new data with the same statistics as the training data e.g. to generate photographs of human faces [15]. Classifiers aim to predict the class of an input e.g. what item of clothing is in a picture [16]. Deep learning refers to a neural network with many layers (normally three or more). Deeper neural networks differ from normal neural networks in that they can perform better feature extraction. Autoencoders carry out unsupervised learning. In this paper a deep convolutional autoencoder (CNN) is used.

## 2 Materials and Methods

### 2.1 Molecular Dynamics

For this project's MD simulations, X-ray crystallography was used to provide the initial arrangements for MurD. Nuclear magnetic resonance (NMR) is more applicable to lighter proteins since it is difficult to derive the protein's structure based upon distances between atoms, which NMR provides. Although, more recently NMR has been able to resolve structures of up to  $\sim 100\text{kDa}$  which would resolve Mur-D, which has a mass of 47kDa [17]. Cryo-electron microscopy can generally resolve structures from 100kDa [18]. The

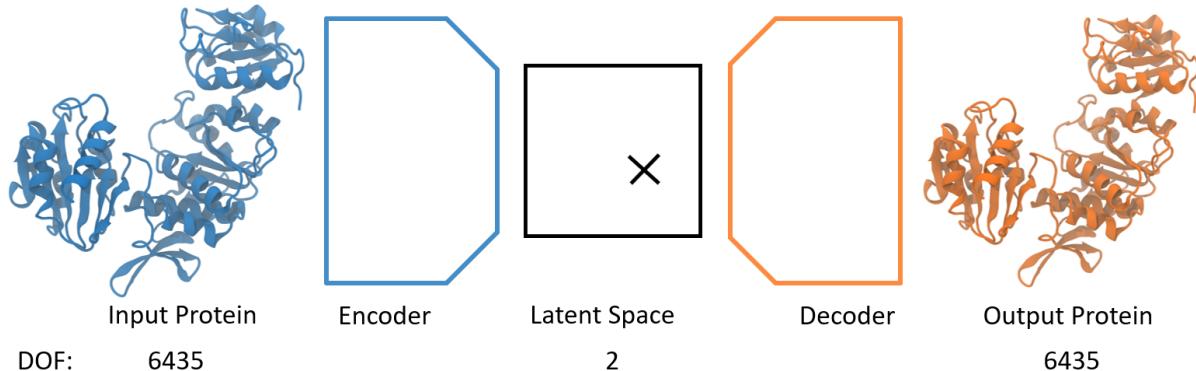
difficulty with X-ray crystallography is finding the right conditions under which the protein can arrange into a crystal.

This project relied upon three MD simulations performed by the Degiacomi group [5]. Two separate simulations were performed on an experimentally determined open and closed state. This resulted in two ensembles which maintained their spatial arrangement. The training set combined 2507 conformations from the closed ensemble and 1913 from the open ensemble. Normally with machine learning (ML) a testing set is separated from the training set which the network is not trained on and allows us to later test the network. In this project the network was challenged by being provided with conformations from a separate MD simulation. This simulation came from an experimental conformation which had its ligand removed. The simulation followed the protein transitioning from the closed state to the open state.

For the simulations a timestep of 2fs was used for a simulation time of 200ns. The temperature was kept constant at 300k using Langevin dynamics and a pressure of 1 atm was kept constant via a Langevin piston. The SHAKE algorithm was used to restrain covalent bonds whose frequencies are highest so that a larger timestep could be used [5].

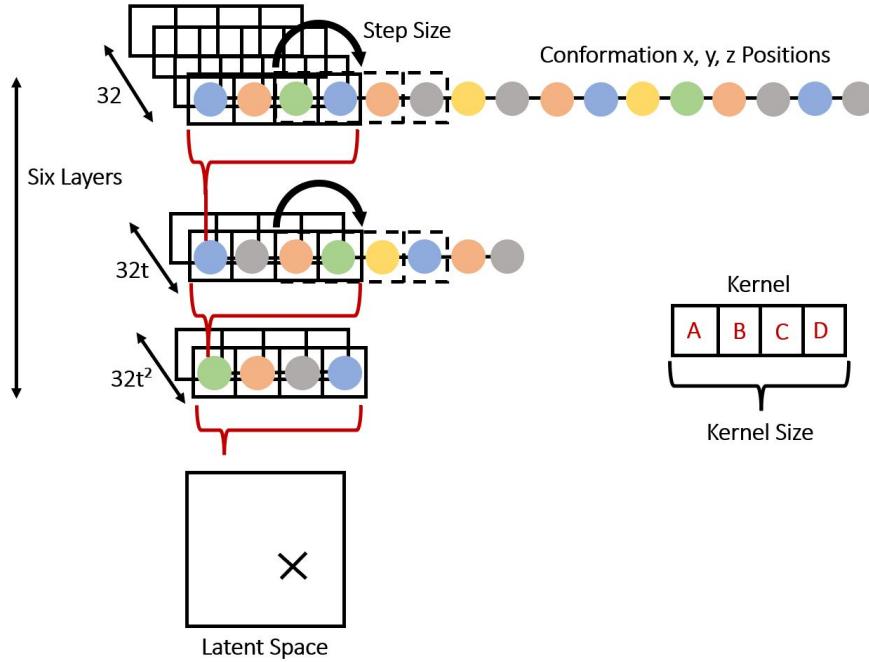
## 2.2 Overview of Molearn

Molearn, the neural network used in this project, was developed by the Degiacomi research group [19] [20]. Put simply, it takes the conformations of a protein produced from MD and uses these to more accurately predict unseen conformations. It does so by training itself on the conformations, passing them through an encoder and decoder. Here the dimensionality of the protein is reduced to two dimensions and enlarged again, losing information in the process. The two dimensions to which it is reduced can be shown on a latent space. Certain points in the latent space therefore correspond to real structures the network has encoded. But new 2D points of interest can be decoded from the latent space and reveal new structures which have not been seen before. Fig. 7 shows the general structure of Molearn.



**Figure 7:** The general structure of Molearn consisting an encoder and decoder. The encoder has six layers which reduces the dimensionality of an input conformation to 2, where it can be represented on a latent space. The decoder re-expands this to the full number of DOF.

Molearn performs dimensionality reduction by performing convolutions on the conformation x, y, z positions (conformation array). The dimension is halved each time until it reaches two dimensions. The last layer before the two dimensions has the kernel and stride automatically set by the network to reach 2D. This process can be seen in Fig. 8.



**Figure 8:** The process of encoding a conformation to a 2D point in Molearn. A kernel, shown on the right as a  $4 \times 1$  grid, passes along the x, y, z positions of the backbone and multiplies each coordinate by a weight in each kernel cell A, B, C, D. It skips two amino acids every step and therefore halves the dimension of the input iteratively through the six layers until there are 2 values. The depth of each layer, i.e. the number of kernels, is given by  $32t^n$  where n is the layer. The network improves its predictions by changing the weights of the kernels by back-propagation.

The conformations are inputted to Molearn from the MD simulations as PDB files which contain the x, y, z coordinates of all conformations. Different atom selections can be made on the MD conformations. The main trajectory of the protein is determined via the backbone so this is important to include. The approximate positions of the sidechains can be informed by the positions of the C $\beta$  atoms. Including more atoms introduces more calculations than needed. This is why originally the atoms selected were the backbone+C $\beta$ . Other atom selections inputted to the network such as solely the backbone or all atoms were considered in this paper. The kernels only perform operations on the positions of the atoms, they have no sense of what atomic elements are.

The size of the kernels is also arbitrary and can be chosen alongside other network parameters to reach the desired dimensionality reduction. For our networks we wanted to halve the dimension each time. This meant choosing the parameters so that Eq. 1 was satisfied with  $L_{in} = 2L_{out}$  where  $L_{out}$  is the output dimension and  $L_{in}$  is the input dimension [21].

$$L_{out} = \frac{L_{in} + 2 \times padding - dilation \times (kernel size - 1) - 1}{stride} + 1 \quad (1)$$

The *padding* is the addition of extra atoms to the conformation array which have values of 0. The *dilation* (default 1) is the spacing between kernel elements. The *kernel size* (default 4) is the number of elements in the kernel. The *stride* (default 2) is the step size between convolutions. In this paper we investigated differing kernel sizes (4, 6, 8, 12) which required changing the padding to 1, 2, 3, 5 respectively.

The latent space dimension the conformations are projected onto is similarly arbitrary. 2D was chosen as it can easily be visualised but more dimensions can be used to preserve more information in a network pass. We investigated latent space dimensions of 2, 3, 5 and 10.

The training occurred in 200 epochs (with 500 iterations) with the full training set passing through the neural network at each epoch. The aim is to complete the minimum number of epochs required to optimise the weights of the network without overfitting. Overfitting occurs when the network specialises at reconstructing the training dataset and hence performs poorly when given another dataset.

In each iteration 16 random conformations were taken as a batch, one of which is chosen to be encoded and decoded. Its mean squared error (MSE) is calculated with respect to the input conformation. This error (loss) forms one part of the loss function: the MSE loss function. A second conformation is randomly picked from the 16 and is encoded too. A halfway linear interpolation is calculated between the two latent space coordinates. This interpolated point is also decoded and its potential energy is determined by measuring the bonds, angles, dihedral and non-bonded forces. This forms the second half of the loss function: the physics based loss function. The weights are updated by attempting to minimise the loss function through back-propagation. The interpolated structure will not always be between open and closed states, however by having this loss function the network improves in calculating intermediary states. Once the training has finished we can ask the network to decode any latent space coordinates and generate new unseen structures.

The code used for the project was written in Python, including the network which used Pytorch [22]. Training runs were carried out on the NCC supercomputer in the Computer Science department at Durham University. This used GPUs which are particularly efficient at running neural networks. Five repeats were taken for each training run (apart from latent dimensions 3, 5 and 10). Training runs varied between 5 and 16 hrs. For a kernel size of 4 we chose three atom selections: backbone, backbone+C $\beta$  and all atoms. For a kernel size of 6, 8 and 8 we chose two atom selections: backbone+C $\beta$  and all atoms. For latent dimensions of 3, 5 and 10 we chose the atom selections: backbone+C $\beta$  and all atoms.

### 3 Project Objectives

As described in the Molearn section, the neural network has a defined architecture which depends on certain parameters. These are defined such that the dimensionality of the conformations are reduced and expanded again in the encoding and decoding process. Eq. 1 defines this relationship. The parameters were never formally investigated and we thought the parameters which could have most of an effect were the kernel size, latent space dimension, number of channels and stride. We also considered changing the atoms selected for the conformations before they were inputted to the neural network. For this project it was decided to test the effect of the kernel size, latent space dimension and atom selection. Increasing the kernel size increases the complexity of the network and we thought it would enable better modelling of the change in structure between conformations. A larger latent space dimension allows more information to be preserved in the encoding and decoding process. Currently, other software is used to reconstruct the conformations and we wanted to see if Molearn could achieve this with better results (by inputting all atoms), or if decreasing the number of atoms would allow the network to track the trajectory of the backbone even better.

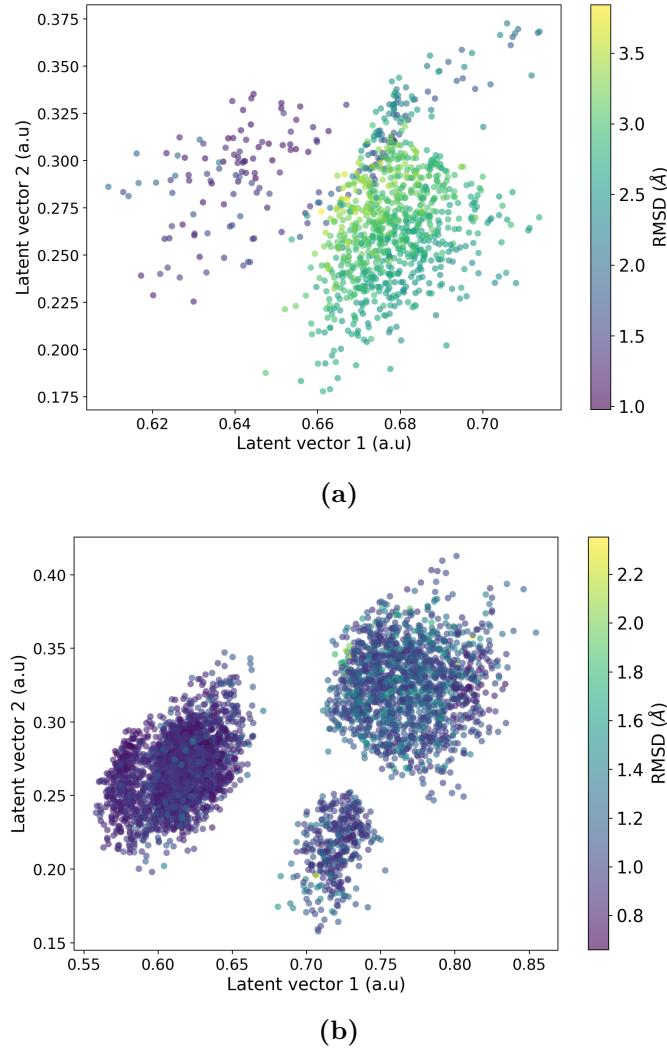
We can test the efficacy of the network through several ways. The similarity of the input and output conformations can be measured with the RMSD (root mean squared deviation). A measure of the potential energy of the output conformations could be determined with the DOPE score (discrete optimised potential energy) [23]. We could test the ability of the network to discover new intermediate structures by decoding interpolations between open and closed latent space clusters. All these methods were employed to test the effects of these parameters in this project.

## 4 Results and Discussion

### 4.1 Latent Space

Once a training run had finished, the neural network weights were passed to a Python script. The script loaded the network file and was able to encode/decode the training/test set conformations as desired. The results of projecting the conformations from the training set and test set onto a latent space can be seen in Fig. 9.

The clusters formed in Fig. 9 show the nature of the conformations provided. In (a) the two clusters have distinct colours and likely correspond to the open and closed states of MurD. The closed states are more rigid and their shapes do not deviate very much from each other, principally because there are more bonds between the moving and static domain. The open states are more flexible because the bonds are weaker allowing the moving domain to shift position more easily with respect to the static domains. The network will have seen similar states from the training set when reconstructing closed states but may be seeing quite distinct structures in the open states. This division of clusters is supported by the fact there are much fewer states in the low RMSD cluster of (a) than in the high RMSD cluster which is a reflection of the true number of closed and



**Figure 9:** The 2D projection of all the encoded conformations in a test set (a) and training set (b) from one of the neural networks. One of five neural networks were chosen to display the latent space. The colour of a point is given by the RMSD between an input conformation and its encoded/decoded conformation. There are 4420 points for the training set and 1000 points for the test set.

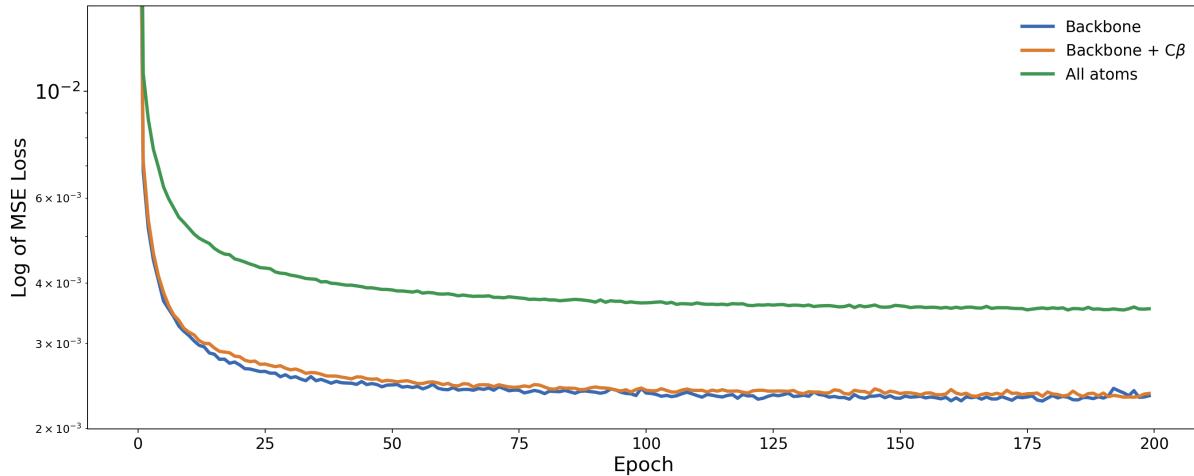
open states in the test set (1:9). The training set (b) has a more even distribution of open and closed states of 1:1.3. This probably contributes to a lower overall median RMSD of the training set ( $0.98 \text{ \AA}$ ) compared to the training set ( $2.78 \text{ \AA}$ ).

However, the main reason that the training set is better is because the two datasets come from completely different MD simulations. Normally a dataset is divided in two into the training and test set but here two different starting conformations in MD provided the datasets. The network's weights were optimised with the type of conformations seen in the training set so it is expected the training set will perform better.

## 4.2 Atoms selected in the Conformations

When Molearn was developed the conformations inputted to the network were modified such that only the backbone and the first carbon ( $\beta$ ) was included. This was done in order to effectively capture the trajectory of the backbone which generally determines how the protein folds. The  $\beta$  carbon was included so that the sidechain could effectively be reconstructed afterwards. Here we present results of the efficacy of the network when two different atom selections were made: just the backbone, and all the atoms.

Five training runs were completed for each atom selection. Fig. 10 shows the Mean Squared Error loss (calculated for the loss function after a single pass) as a function of epoch for the three different atom selections averaged over the five repeats. It was important that the loss converged so that we knew the weights had been optimised and that further training would be detrimental to the network’s performance.



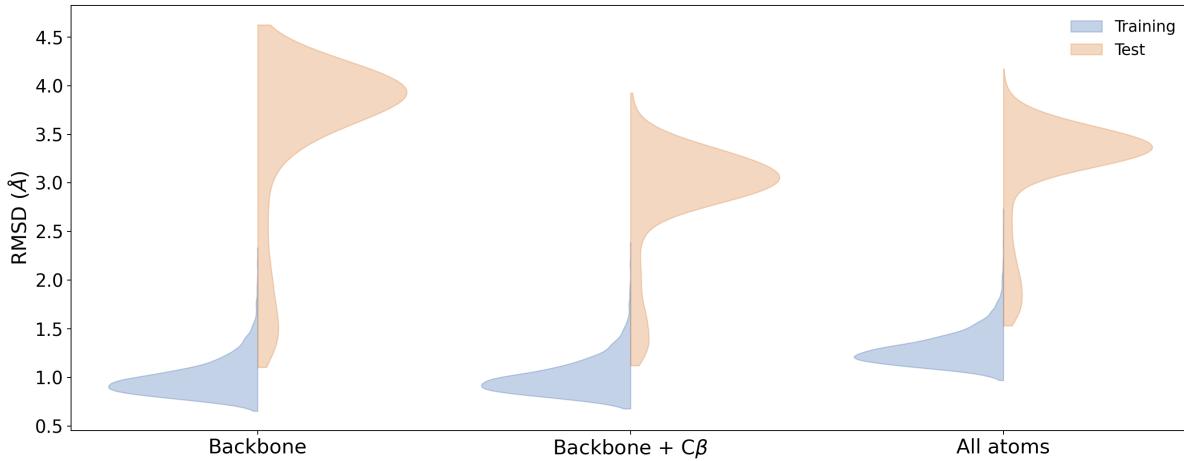
**Figure 10:** The log of the MSE loss as a function of epoch for three different atom considerations: Backbone, Backbone+C $\beta$  and All atoms. Each line is an average of five repeats. All lines clearly plateau to a minimum, although the minimum for the all atoms is considerably bigger.

It can be seen that the converging MSE loss for the all atoms is  $\sim 1.5\times$  greater than the converging loss of the backbone or backbone+C $\beta$ . It is reasonable that a higher number of atoms is more difficult to encode/decode as there is more information for the network to manage. The kernel weights have to be adjusted while considering the positions of more atoms. However, this cannot explain why the all atoms convergence is much higher because the backbone and backbone+C $\beta$  have a different number of atoms but a similar convergence. It is more likely the network was struggling to reconstruct the positions of the sidechains which cannot be easily tracked on the latent space.

It is difficult to tell from Fig. 10 alone that the networks were overfitting. The network’s performance at each epoch when given the test set would be able to show this. This would require saving the network’s weights at each epoch which requires substantial storage (37GB for a single training run with a kernel of 4, 59GB for a kernel of 12). However, it appears from Fig. 10 that fewer epochs could be used without affecting the network’s performance significantly.

The convergence of the MSE loss function indicates that a network has optimised its weights, but is not an indication of a network's reconstruction ability. The RMSD and DOPE score were used for this as they score the spatial similarity of the input and output conformation and the potential energy of the output conformation respectively.

Fig. 11 shows the RMSD distributions for the three atoms selections with a kernel of 4.

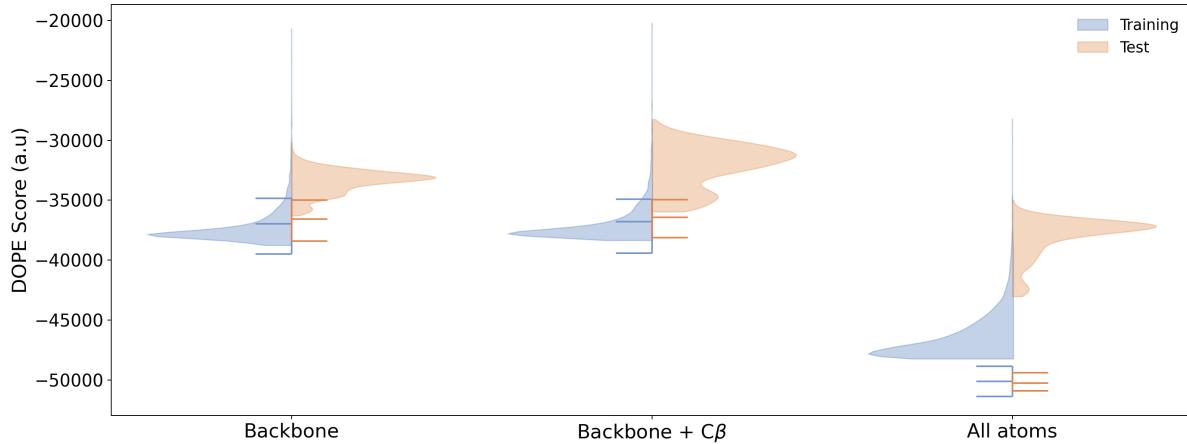


**Figure 11:** RMSD distributions of the training set containing 4420 output conformations, as shown in blue, and the test set containing 1000 output conformations as shown in orange. All distributions are an average from five networks. This is for three atoms selections for the conformations: only the backbone, backbone+C $\beta$  and all the atoms.

The RMSD is a score of the difference in position of the atoms between two conformations, in this case between the input protein and output protein. A score as low as possible is desirable because this means the conformation is reconstructed accurately and little information has been lost in the process of dimensionality reduction and expansion. As can be seen from Fig. 11 the ability of all atom selections to reconstruct the conformations is much better with the training set. The median values are 0.95, 0.97 and 1.26 for the backbone, backbone+C $\beta$  and all atoms respectively which are good results. A value of under 1 Å is generally considered very good. The test set values are larger with median values of 3.83, 3.01 and 3.33 Å which are reasonable. From this it is clear the backbone+C $\beta$  selection's performance is the best. The test set is comprised of, as noted above, closed states which are similarly shaped between the training and test set and open states which are more flexible and differ more. The 'tail' in the test set plots corresponds to the closed states, of which the RMSD score is similar to the training set. The main bell curve corresponds to those open states which are completely unseen.

It is difficult to speculate on the reasons behind the difference between the RMSD in each of their atom selections. The all atoms selection likely performs worse than backbone+C $\beta$  because the network has to learn how to position more atoms in the reconstruction process. The network weights are therefore less specialised as more information has to be captured. The all atoms has 1.5× more atoms than the backbone+C $\beta$  (3286 to 2145).

Another score important in determining the quality of a conformation reconstruction is the DOPE score. It is a measure of the physical validity of a conformation as it measures the conformation's potential energy. It is an arbitrary score and therefore the distribution of the output conformations have to be compared to the distribution of the input conformations. Fig. 12 shows these distributions.



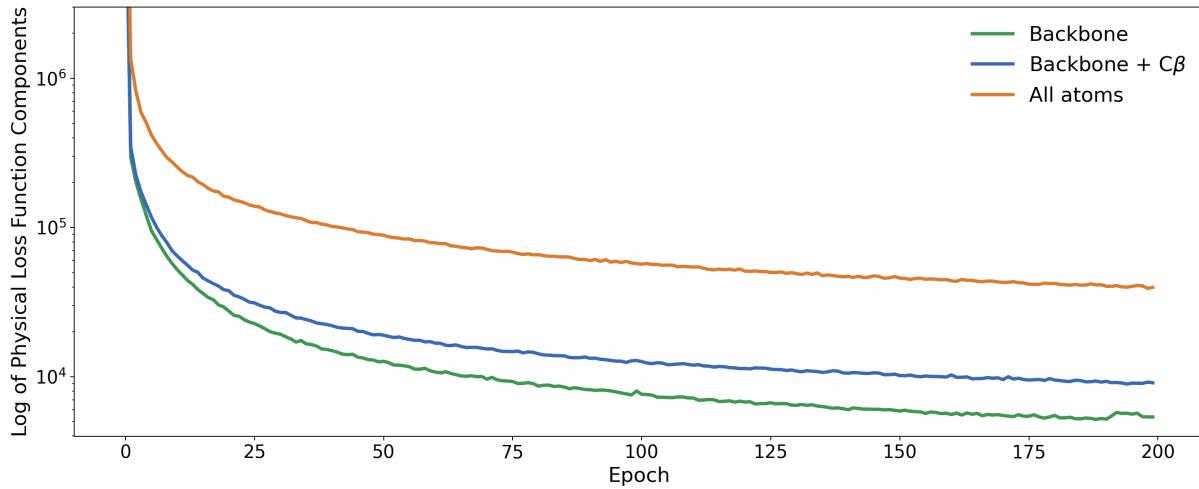
**Figure 12:** The DOPE score distributions of the output conformations of the training set and test set for three atom selections: backbone, backbone+C $\beta$  and all atoms. The coloured bars give the median and extrema values of the input DOPE scores (not output as in the distributions) for each atom selection to which the output conformations can be compared. All distributions are an average from five networks.

The DOPE score is a statistical measure of potential energy and because of Maxwell-Boltzmann statistics it is more probable that conformations are found with a low energy. A low DOPE score is therefore more physically realistic and indicative of a good reconstruction. The DOPE score scales with the number of atoms. All the output conformations are reconstructed with their full atoms before the DOPE score is calculated. For this reason it is plausible that the all atoms selection in Fig. 12 have lower scores. The decoded conformations of the backbone selection are comparable to the backbone+C $\beta$  even though it has fewer atoms. One reason for this may be that the conformations have all their atoms (apart from hydrogen) re-added before the DOPE score is calculated. This is done with Modeller software and for the backbone it will optimally place the C $\beta$  [24]. The placement may be better than the network's attempt and lead to a lower DOPE as seen here.

The median DOPE score of the input conformations for the backbone and backbone+C $\beta$  are higher than the DOPE scores of the output conformations. This may seem counter-intuitive because one would expect the networks to reconstruct the conformations with some physical error which would lead to a higher score. However, the networks have a physical loss function which aims to minimise the potential energy of the output conformations. The MD simulations by contrast aim to keep a constant energy at constant NPT.

The median DOPE score of the all atoms output conformations are considerably higher

than the input. The RMSD distribution of the output, as can be seen in Fig. 11 is however reasonable. This indicates that approximately the atoms are in the correct positions but the network struggles to fine-tune the positions such that interatomic distances are realistic. One can make sense of the DOPE score results by looking at the components of the loss function which aims to minimise interatomic potential energies. Fig. 13 shows the sum of the physical loss function components as a function of epoch for the three atom selections.



**Figure 13:** The log of the physical loss function components over epoch for three different atom selections: backbone, backbone+C $\beta$  and all atoms. The physical loss function is composed of four separate loss functions associated with the interatomic bonds, angles, torsions and non-bonded energies. These four loss functions are summed to give the physical loss function. Each atom selection is an average of five repeats.

From Fig. 13 we can see that the network is better at minimising the physical loss function components for lower numbers of atoms. The ratio of the final physical loss function components between backbone, backbone+C $\beta$  and all atoms is 1:1.7:7.4. This differing convergence between the atom selections is likely the cause of their difference in DOPE score as the loss function is based off similar metrics. It is understandable the network struggles to lower the physical loss function for the all atoms and backbone+C $\beta$ . With the same number of layers and kernel size the network has to learn to encode/decode a conformation with both more atoms and different interatomic potentials. For solely the backbone, Modeller will optimally place these atoms later. The network weights have to generalise more to accommodate for the increase in information to encode. It looks from Fig. 13 that increasing the number of epochs might give a small improvement for the all atoms and backbone+C $\beta$ .

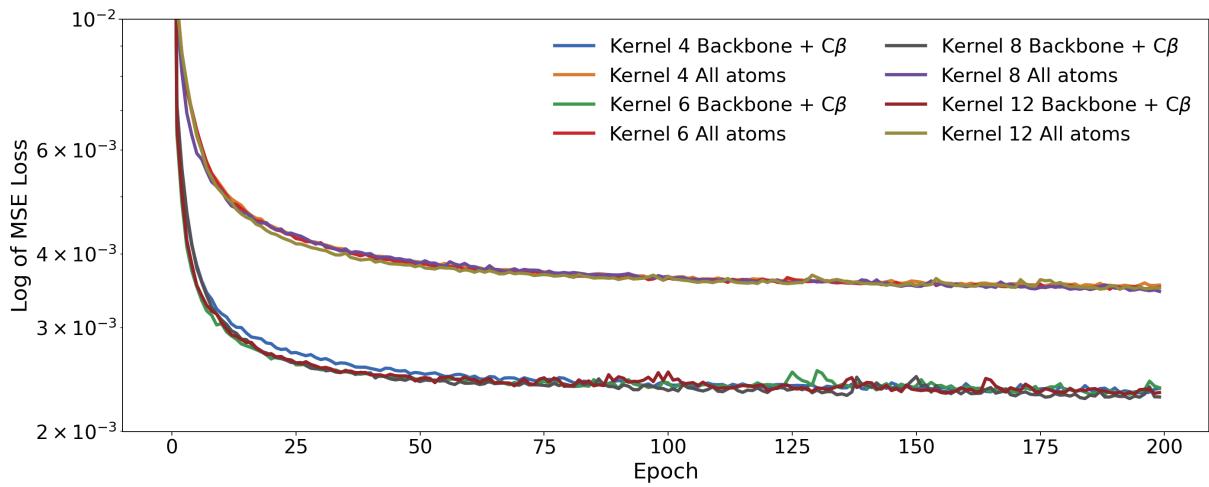
From these results it was chosen that further benchmarks would be conducted on networks with atom selections of backbone+C $\beta$  and all atoms. Although the backbone atom selection performed better on the DOPE scores we can use energy minimisation procedures to improve this. Poor shape, as indicated by the RMSD is more difficult to recover. We also wanted to test the neural network by increasing the number of atoms

provided with the aim that Molearn will be able to handle all atoms in the future.

### 4.3 Effect of Kernel Size

Another parameter we wanted to investigate in this project was the kernel size. The network originally used a kernel of 4 as this is often used in convolutional neural networks. It also meant that in one iteration the most relevant interatomic potentials could be captured (particularly up to the torsional angle which occurs between four atoms). Increasing the kernel size increases the number of other atoms considered when performing a pass and could improve performance. We investigated this with kernel sizes of 4, 6, 8 and 12.

As with the kernel 4 case it was important to check the MSE loss functions were converging. Fig. 14 shows the results of the four kernels for backbone+C $\beta$  and all atoms.

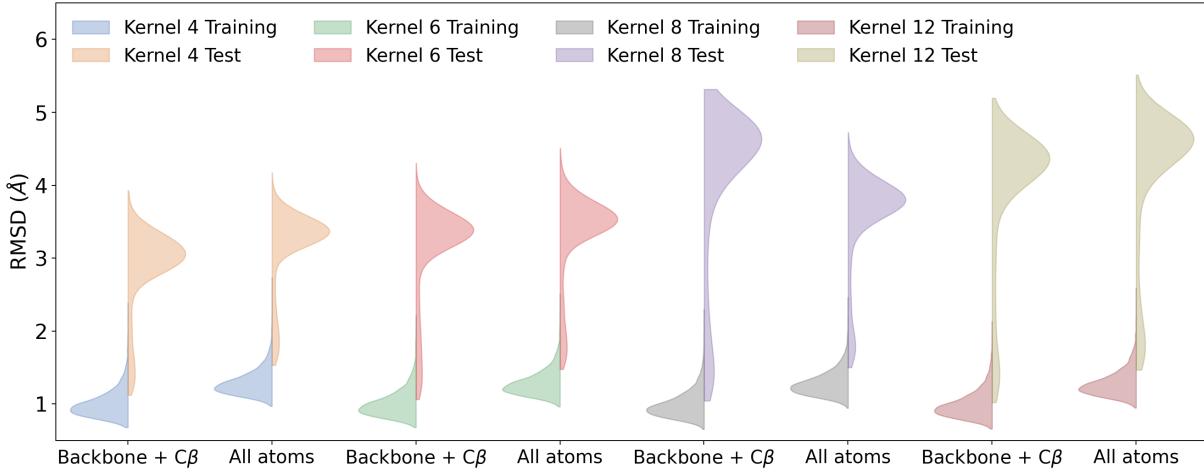


**Figure 14:** The log of the MSE loss as a function of epoch for four different kernel size (4, 6, 8, 12) for the atom selections backbone+C $\beta$  and all atoms. It is difficult to distinguish the lines but the all atom selections lines form one plateau above the lines of the backbone+C $\beta$ . All lines are the means of results from five networks.

It can be seen from Fig. 14 that kernel size does not significantly affect the convergence of the MSE loss function. We can see that the atom selection, as before, is the dominant cause of any change in the MSE loss function. We knew the networks were running long enough because all the lines were converging. We plotted the distributions of RMSD for all combinations of the four kernels and atom selections. Fig. 15 shows these results. Table 1 shows the median values of the RMSD for the same data.

The characteristic patterns of Fig. 11 are also exhibited in this plot. The test set distributions all have a main bell curve corresponding to the open states and a tail corresponding to the closed states. The training set RMSD distributions likewise have considerably smaller RMSDs.

From looking at Fig. 15 and Table 1 (below) it is quite clear that RMSD increases with larger kernel size for the test set for both backbone+C $\beta$  and all atoms. The whole distributions noticeably shift higher and this is reflected with the higher median values. By contrast the training set distributions are very homogeneous amongst the two atom



**Figure 15:** The RMSD distributions of the output conformations of the training set and test set for four different kernel sizes (4, 6, 8, 12) with two different atom selections: backbone+C $\beta$  and all atoms. All distributions are averages of five repeats.

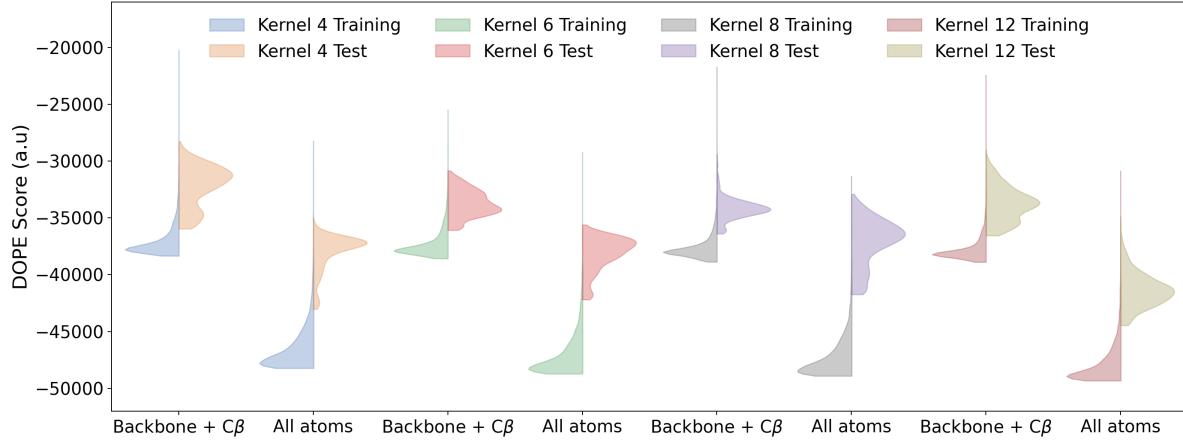
**Table 1:** The median RMSD(Å) values of the output conformations compared to input conformations of the training and test sets for four different kernel size (4, 6, 8, 12) for two atom selections: all atoms and backbone+C $\beta$ .

| Kernel Size | Backbone+C $\beta$ |      | All Atoms |      |
|-------------|--------------------|------|-----------|------|
|             | Training           | Test | Training  | Test |
| 4           | 0.97               | 3.0  | 1.26      | 3.3  |
| 6           | 0.97               | 3.3  | 1.25      | 3.5  |
| 8           | 0.96               | 4.5  | 1.25      | 3.8  |
| 12          | 0.96               | 4.3  | 1.25      | 4.5  |

selections. This indicates overfitting. One problem with increasing the kernel size without changing the number of channels is that the total number of weights in the networks increases. More complex networks, such as those with a higher number of weights, are more prone to overfitting. We decided not to decrease the number of channels accordingly because this can have a separate effect on a network’s performance. To determine the effect of both would require having all kernel sizes for all channel sizes which would take considerable time to run.

It was thought that by increasing the kernel size the network would be more aware of the positions of further atoms and that this could improve performance. This may be less important than we thought when we consider the receptive field of a kernel. This is the portion of the data (or section of atoms) which can illicit a change deeper in the network. The dimensionality reduction means that the atoms a few layers deep can see larger proportions of the data further up. For example they are affected by 4 atoms one layer up and then  $4^2$  atoms the layer above that. This means atoms distant from each other in the first layer can affect each other deeper down. We do not need to increase the kernel size for the network to be aware of atoms far away from its location in the chain.

The DOPE score distributions were also calculated and the results are shown in Fig. 16. Table 2 shows the median DOPE score values for the same distributions.



**Figure 16:** The DOPE score distributions of the output conformations of the training set and test set for different kernel sizes (4, 6, 8, 12) with two atom selections: backbone+C $\beta$  and all atoms. All distributions are averages from five repeats.

**Table 2:** The median DOPE score values of the output conformations of the training and test sets for four different kernel size (4, 6, 8, 12) for two atom selections: all atoms and backbone+C $\beta$ .

| Kernel Size | Backbone+C $\beta$ (a.u $\times 10^3$ ) |       | All Atoms(a.u $\times 10^3$ ) |       |
|-------------|---|-------|-------------------------------|-------|
|             | Training                                | Test  | Training                      | Test  |
| 4           | -37.5                                   | -31.8 | -47.1                         | -37.5 |
| 6           | -37.7                                   | -33.8 | -47.7                         | -37.8 |
| 8           | -37.9                                   | -34.3 | -47.9                         | -36.8 |
| 12          | -38.1                                   | -33.8 | -48.5                         | -41.5 |

The results of the different kernel sizes on the DOPE scores are more homogeneous. The median values of both the test and training set (both atom selections) decrease a little with higher kernel size but not significantly. There is not sufficient difference for us to consider any of the networks better than another. The DOPE score is a statistical potential and therefore based on probability. A good DOPE score is therefore not indicative of a good structure because although the positions may be reasonable the interatomic potentials of the angles and torsions may not be realistic. A poor DOPE score however is a signature of a bad structure. All these distributions are reasonable.

From these results it was determined that the best networks were those with a kernel of 4 and an atom selection of backbone+C $\beta$ . The RMSD of these networks were the best and the DOPE scores were all very similar. It is worth noting that it is more important that the shape of these conformations are good than their energy. Energy minimisation can be carried out on the conformations after the neural network but it is difficult to improve

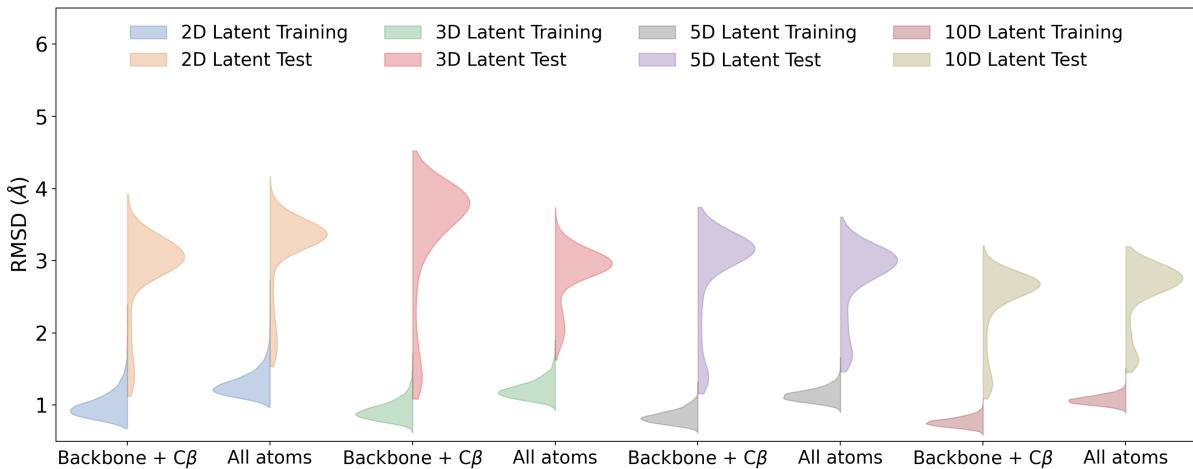
their shape. A kernel of four was the most simple network and these are favoured over more complicated ones. It may be possible that larger kernels would perform better if the networks were altered elsewhere but that would require further work to establish.

#### 4.4 Effect of Latent Space Dimension

A final parameter that we investigated in this project was the latent space dimension. The dimension of the latent space is the bottleneck of the network and constrains how much information can be transferred from the encoder to decoder. It is difficult to discern how exactly the network encodes information onto the latent space. By looking at the latent space clusters in the training set it appears that it tries to capture the movement of the domain of MurD which opens and closes. We thought that increasing the latent space dimension would allow the networks to improve this encoding of the trajectory of the backbone. We particularly hoped this for the more disordered open states where the backbone has more flexibility and is more difficult to encode. As well as the latent space dimension of 2 we also tried 3, 5 and 10 for both backbone+C $\beta$  and all atoms.

The training runs for these dimensions were performed at the end of this project and therefore only single runs were completed for each case. We checked the convergence of the MSE and physical loss function components and found they converged as in other training runs. As seen before the all atoms runs converged to a value on average  $1.8 \times$  higher than the backbone+C $\beta$  runs. The convergence was negligibly different between latent space dimensions.

Fig. 17 shows the results for the four different latent space dimensions for backbone+C $\beta$  and all atoms. Table 3 shows the median RMSD values for the same distributions.



**Figure 17:** The RMSD distributions of the output conformations of the training set and test set for different latent space dimensions (2, 3, 5, 10) with two atom selections: backbone+C $\beta$  and all atoms. The 2D latent space distributions are an average of five repeats and the rest are from single training runs.

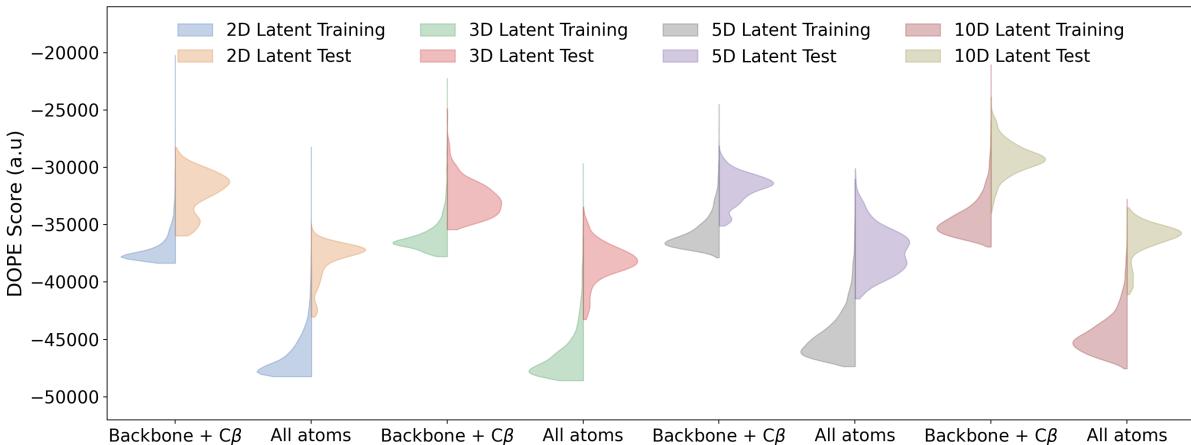
As can be seen from Fig. 17 and Table 3 a higher latent space dimension has inconclusive effects on the RMSD for backbone+C $\beta$  but decreases it for all atoms (in the test

**Table 3:** The median RMSD values of the output conformations of the training and test sets for four latent space dimensions (2, 3, 5, 10) for two atom selections: all atoms and backbone+C $\beta$ .

| Latent Dimension | Backbone+C $\beta$ ( $\text{\AA}$ ) |      | All Atoms ( $\text{\AA}$ ) |      |
|------------------|-------------------------------------|------|----------------------------|------|
|                  | Training                            | Test | Training                   | Test |
| 2                | 0.96                                | 3.01 | 1.26                       | 3.33 |
| 3                | 0.90                                | 3.67 | 1.20                       | 2.92 |
| 5                | 0.83                                | 3.10 | 1.13                       | 2.92 |
| 10               | 0.76                                | 2.63 | 1.07                       | 2.71 |

set). The backbone+C $\beta$  with a latent dimension of 10 has the lowest median RMSD at 2.63  $\text{\AA}$ . It is quite unexpected that the higher latent dimension networks perform better with all atoms than backbone+C $\beta$  as more information needs to be decoded. The tail of the RMSD is higher for the higher latent dimension networks but the median is still lower than the previous best of 3.01  $\text{\AA}$ . This indicates that these networks are likely performing better on the disordered open states as more information can be preserved of the domain shifts with the higher dimensions. The networks producing the backbone+C $\beta$  distributions with higher latent dimensions (except 10D) may be overfitting as the training set remains low while the test increases quite significantly. To confirm these results more repeats would need to be taken.

The distributions of the DOPE scores for the four latent space dimensions can be seen in Fig. 18. Table 4 shows the median DOPE values for the same distributions.



**Figure 18:** The DOPE score distributions of the output conformations of the training set and test set for different latent space dimensions (2, 3, 5, 10) with two atom selections: backbone+C $\beta$  and all atoms. The 2D latent space distributions are an average of five repeats and the rest are from single training runs.

The training set distributions of both atom selections had an increasing median DOPE score alongside the latent space dimension, although all values are firmly negative. The test set values are more homogeneous which indicates the potential energies are probably quite good for all the distributions.

**Table 4:** The median DOPE score values of the output conformations of the training and test sets for four different latent space dimensions (2, 3, 5, 10) for two atom selections: all atoms and backbone+C $\beta$ . The 2D latent dimension distribution median is from an average of five repeats but the other latent space dimensions are from single training runs.

| Latent Dimension | Backbone+C $\beta$ (a.u $\times 10^3$ ) |       | All Atoms(a.u $\times 10^3$ ) |       |
|------------------|---|-------|-------------------------------|-------|
|                  | Training                                | Test  | Training                      | Test  |
| 2                | -37.5                                   | -31.8 | -47.1                         | -37.5 |
| 3                | -36.4                                   | -32.9 | -47.1                         | -38.2 |
| 5                | -36.1                                   | -31.6 | -45.2                         | -37.2 |
| 10               | -34.8                                   | -29.2 | -44.9                         | -35.9 |

Networks with higher latent dimensions should be able to perform at least as well as those with fewer (without considering overfitting). If using 2D were optimal then the network should learn to disregard more dimensions and solely use the two. If the latent space had as many dimensions as DOF in the conformations there would be a one to one mapping and there would be no error. A disadvantage with higher latent space dimensions is that they cannot easily be visualised, particularly above 3D. It is not easy to see if the conformations are being arranged into clusters and latent space points we would like to decode cannot be chosen so easily.

#### 4.5 Comparison of Output Conformations to Experimental Intermediates

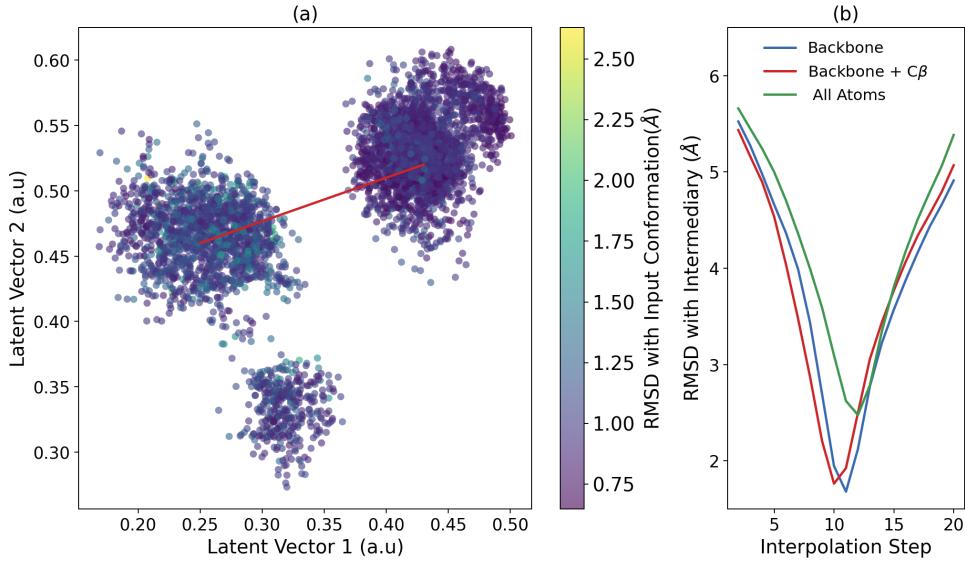
As discussed in the previous section the networks with a kernel of four and atom selection of backbone+C $\beta$  were overall deemed the most favourable networks for a 2D latent space. For this reason the following plots principally used networks with these parameters.

The network's loss function is trained by calculating the MSE between an interpolation conformation and a known intermediary. The interpolation conformation is found by interpolating between two random structures in a batch of 16. Over thousands of batches the network improves in guessing between two structures which will often be a closed and open state. The network therefore learns how to make a good estimate of an intermediate structure.

Fig.19 shows a possible example of this process. Fig. 20 shows some of the interpolations plotted in Fig.19 (b).

Once the networks with the backbone+C $\beta$  atoms selected with a kernel of four were deemed the best we compared structures obtained by decoding the latent space to an experimentally known intermediate structure of MurD.

Interpolations are a good way of guessing intermediate structures but they are linear and do not cover the whole space between open and closed clusters. Here we looked at the whole latent space's similarity to a known intermediate. This was done by splitting the latent space of one of the five neural networks into a  $50 \times 50$  grid. Each cell's 2D coordinates were decoded into their conformations and then the RMSD of that structure



**Figure 19:** Plot (a) shows the latent space for one of the networks with a kernel of 4 and an atom selection of backbone+C $\beta$ . A red line shows a possible interpolation between two random structures in a batch. The interpolation will be halfway across this line. Plot (b) shows the RMSD between interpolations (between two random structures in one of the batches) and the known experimental intermediary. 20 interpolations are calculated and the distributions are shown not only for the backbone+C $\beta$  selection (the latent space of which is shown) but also for backbone and all atoms. The lowest RMSDs are 1.68, 1.76 and 2.48 Å for the backbone, backbone+C $\beta$  and all atoms selections respectively.

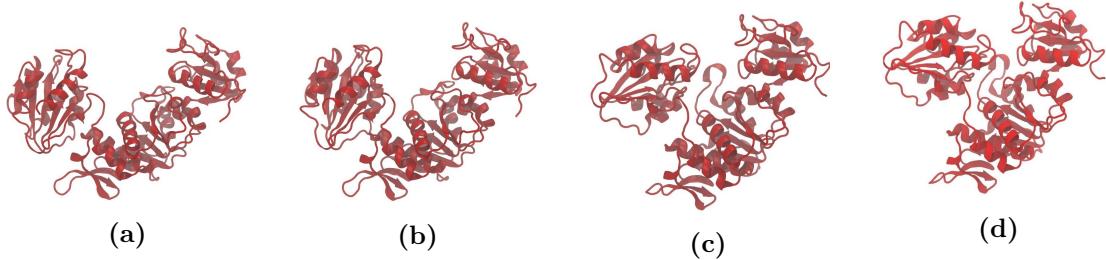
with the known structure was calculated. Fig. 21 shows this plot alongside latent space plots of the same network for the test set and training set structures.

In plot (a) of Fig. 21 we can see that the region of lowest RMSD is situated, as expected, between the two main clusters (which correspond to the opened and closed states) in plot (b). This is the case even though the latent space is not linear. The fact that a large region of the latent space has a similar shape to the known structure indicates that decoded conformations from the latent space can generate unseen but reasonably shaped conformations. Even if the energy is not unfavourable, this can be altered with energy minimisation.

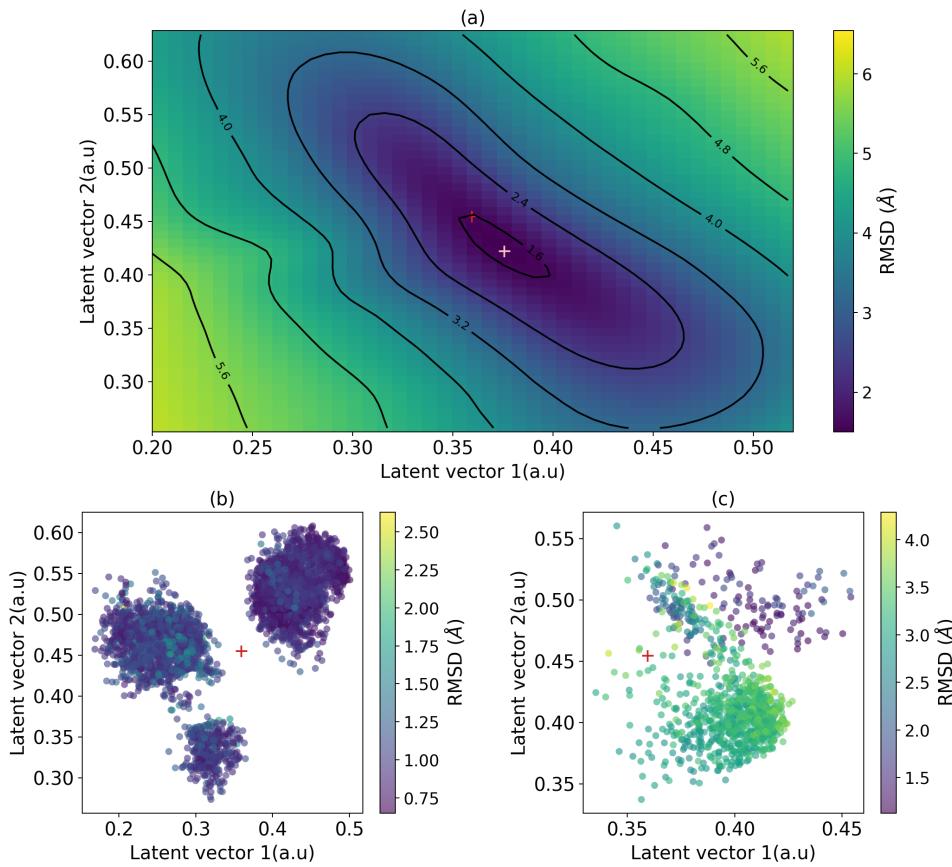
Plot (c) shows that the network is specialised to the training set as the clusters are not as distinct and the red cross does not lie between the two clusters.

Fig. 22 shows the conformation with the lowest RMSD compared to the experimental conformation aligned with said experimental conformation. This shows what a difference of 1.50 Å is like.

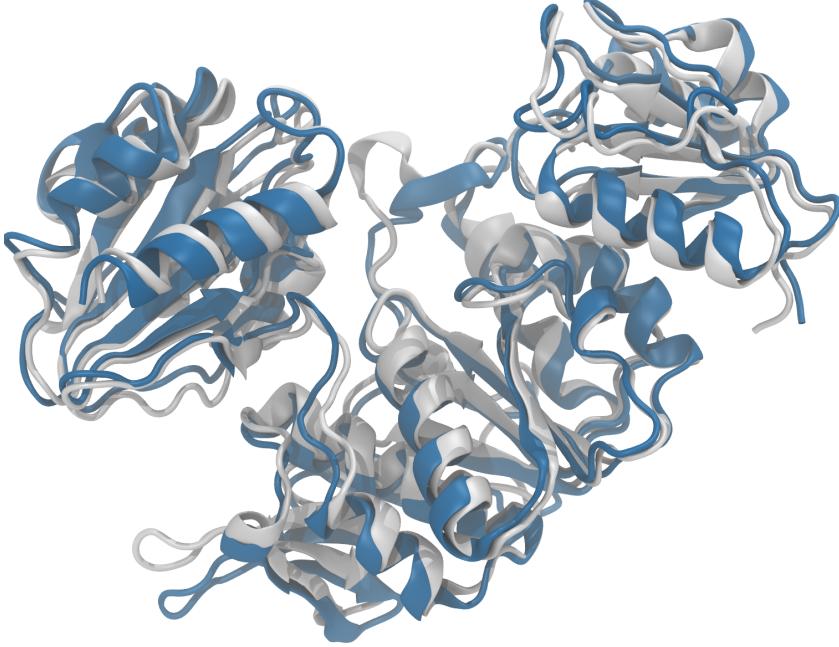
The two conformations in Fig. 22 are very similar in shape. Two regions where the conformations differ distinctly are two loops; one in the bottom left of the figure and the other in the top centre. Loops are very flexible and can change shape easily between conformations.



**Figure 20:** Four of the 20 interpolations between two random structures in the training set for one of the networks with a kernel size of 4 and atom selection of backbone+C $\beta$ . The interpolations are the same as those shown in the red line of Fig. 13 (b). The plot was created in VMD. The interpolations quite clearly show the transition from the open state where the two domains are far apart and the closed state where the two domains are close together.



**Figure 21:** Plot (a) shows the RMSD between output conformations decoded from the latent space and an experimentally known intermediate conformation. The latent space is divided into a  $50 \times 50$  grid the coordinates of which are decoded into the output conformations. The red cross shows the latent space coordinates of the experimentally known structure and the pink cross shows the latent space coordinates with the lowest RMSD (1.50 Å). Plot (b) shows the latent space of the training set with the red cross marking the location of the encoded experimental structure. Plot (c) shows the latent space of the test set with the same experimental structure.



**Figure 22:** A latent space decoded conformation shown in blue, aligned with an experimental conformation shown in white. The conformation decoded from the latent space is the same as that with the pink cross in Fig.21 (a). It has an RMSD from the experimental conformation of 1.5 Å.

#### 4.6 Training Times

The networks were trained in iterations for each parameter definition. Five repeats were taken each time. They were trained on GPUs but despite this the training runs took a considerable amount of time. Table 5 shows the average time taken for each run for the different atom selections and kernel sizes. Table 6 shows the time taken for each run for the different atom selections and latent space dimension.

**Table 5:** The average time over five networks for a training run to complete. The networks have different kernel sizes (4, 6, 8, 12) and atom selections (backbone, backbone+C $\beta$  and all atoms). Times are shown in the format hours:minutes. In parentheses are shown the times in minutes per atom.

| Kernel Size | Backbone    | Backbone+C $\beta$ | All Atoms    |
|-------------|-------------|--------------------|--------------|
| 4           | 5:14 (0.18) | 6:13 (0.17)        | 11:16 (0.21) |
| 6           | N/A         | 6:16 (0.18)        | 10:42 (0.20) |
| 8           | N/A         | 6:31 (0.18)        | 11:51 (0.22) |
| 12          | N/A         | 7:28 (0.21)        | 13:43 (0.25) |

As can be seen from Table 5 the run times generally increase with kernel size, both absolutely and proportionally to the number of atoms. This is expected as there are more weights performing operations on atoms which also need to be updated. Atom selections with more atoms see a proportionally larger run time as well. From Table 6 we see that

**Table 6:** The time for a single network to train, apart from 2D latent which is an average of 5. The networks have different latent space dimensions (2, 3, 5, 10) and atom selections (backbone, backbone+C $\beta$  and all atoms). The kernel size is 4 for every network. Times are shown in the format hours:minutes. In parentheses are shown the times in minutes per atom.

| Latent Dimension | Backbone    | Backbone+C $\beta$ | All Atoms    |
|------------------|-------------|--------------------|--------------|
| 2                | 5:14 (0.18) | 6:13 (0.17)        | 11:16 (0.21) |
| 3                | N/A         | 5:55 (0.17)        | 11:27 (0.21) |
| 5                | N/A         | 5:55 (0.17)        | 11:58 (0.21) |
| 10               | N/A         | 7:28 (0.21)        | 12:13 (0.22) |

the atom selection is again more dominant than changing the latent dimension. It appears that the run time may increase with latent dimension, at least for all atoms, but this is difficult to say without performing more repeats.

Another factor which could be a reason for the longer run times is that the GPUs become hotter as they run which decreases performance.

## 5 Conclusion and Future Work

A neural network, developed by the Degiacomi Group, was used to reproduce and predict conformations of MurD. The ability of this neural network to reconstruct and predict the intermediary conformations of MurD has been shown before [19]. We investigated the effects of changing the kernel size, latent space dimension and atoms selected on the efficacy of the networks, as measured by the RMSD and DOPE score distributions of the output conformations. From these results we suggest retaining a kernel size of 4 and atom selection of backbone+C $\beta$ . The latent space dimension should be further investigated as these results are promising. A good intermediate between 2D and higher dimensions would be 3D as this latent space could still be visualised with the correct software. Molearn has serious potential to inform protein design on intermediary states to aid drug design and other unpredictable applications.

The atom selections inputted to the network were the backbone, backbone+C $\beta$  and all atoms. We found that overall the backbone+C $\beta$  was the best selection because its RMSD distribution was the lowest. Although the DOPE score was higher this can be fixed with energy minimisation once the proteins are reconstructed with all their atoms.

There were similar results for the different kernel sizes where we used a kernel size of 4, 6, 8, 12. A kernel size of 4 with the backbone+C $\beta$  atom selection performed best with the lowest median RMSD score of 3.0Å. The DOPE scores did not vary drastically between kernel sizes with the atom selection having a bigger effect. Unexpectedly the backbone atom selection had a lower DOPE score than the backbone+C $\beta$  even though the whole protein was reconstructed before calculating the DOPE score. The DOPE score of the input test set conformations was -36,400 a.u and the output conformations with a kernel size of 4 with the backbone+C $\beta$  selection was -31,800 a.u. These results were the average

distributions from five repeats.

We also investigated changing the latent space dimension. We found generally increasing the latent dimension decreased the median RMSD. The lowest median score from the test set was 2.63 Å with a 10D latent space with the backbone+C $\beta$  atom selection. The DOPE score seemed to be affected negligibly by the latent dimension. This improvement in the scores was expected as more information on the domain shifts can be preserved with higher dimensions. Only one training run was completed for higher latent space dimensions so more repeats would need to be taken. It would be interesting to isolate the open states to be put through the network to see if these disordered states see a particular improvement in reconstruction.

A more complete method of assessing the potential energies of the conformations would have been to use a Ramachandran plot to analyse the torsional angles of the conformations. The DOPE score is grounded in probability theory and assesses the cartesian coordinates. It does not include other aspects of the potential energy.

Another advantage to both a smaller atom selection and kernel size is the considerable decrease in run time. The backbone+C $\beta$  atom selection took nearly half the time of the all atoms for all kernel sizes. The kernel size had a minor effect on the run times.

We also assessed the network's ability to predict intermediate states. We found that the region in latent space spanning the open and closed state clusters can be decoded to have great similarity to known intermediate states. This shows the network's ability to predict intermediate states of MurD.

## 6 Acknowledgements

The author would like to sincerely thank Dr Matteo Degiacomi for his supervision in the project, showing great patience teaching all the material and always being available to answer questions. The author would also like to extend his thanks to Sam Musson for his help with coding and aid in understanding the neural network.

## Bibliography

- [1] Gruber, A., Durham, A., Huynh, C., et al., editors. Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach. Bethesda (MD): National Center for Biotechnology Information (US); 2008. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK6818/>
- [2] Kuhlman, B., Bradley, P., Advances in protein structure prediction and design. Nat Rev Mol Cell Biol 20, 681–697 (2019). <https://doi.org/10.1038/s41580-019-0163-x>
- [3] Humphrey, W., Dalke, A. and. Schulten, K., VMD – Visual Molecular Dynamics, Journal of Molecular Graphics, vol. 14, pp. 33–38, 1996

- [4] Du, X., Li, Y., Xia, Y. L., Ai, S. M., Liang, J., Sang, P., Ji, X. L., & Liu, S. Q. (2016). Insights into Protein-Ligand Interactions: Mechanisms, Models, and Methods. International journal of molecular sciences, 17(2), 144. <https://doi.org/10.3390/ijms17020144>
- [5] Degiacomi, M., Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space, Structure 27, 2019, 1034–1040
- [6] Number of Released PDB Structures per Year, 2022, RCSB PDB Protein Data Bank, <https://www.rcsb.org/stats/all-released-structures>, 19/04/22
- [7] Noé, F. and Tkatchenko, A. and Müller, K. and Clementi, C., Machine Learning for Molecular Simulation, Annual Reviews, 2020, 71 1 361-390
- [8] UniProtKB - P14900 (MURD ECOLI), 23/02/22, UniProt, <https://www.uniprot.org/uniprot/P14900>, 19/04/22
- [9] Noé, F., De Fabritiis, G., Clementi, C., Machine learning for protein folding and dynamics, Current Opinion in Structural Biology, Volume 60, 2020, Pages 77-84, <https://doi.org/10.1016/j.sbi.2019.12.005>.
- [10] Moore, C., Ergodic theorem, ergodic theory, and statistical mechanics, PNAS, 2015, 112 (7) 1907-1911
- [11] Colarusso, P. Kidder, L. Levin, I., Lewis, E., Raman and Infrared Microspectroscopy, Encyclopedia of Spectroscopy and Spectrometry, Elsevier, 1999, Pages 1945-1954, <https://doi.org/10.1006/rwsp.2000.0402>.
- [12] Neural Networks, 05/10/17, 3blue1brown, <https://www.3blue1brown.com/lessons/neural-networks>, 19/04/22
- [13] Teaching, 2021, CWKX, <https://cwkx.github.io/teaching.html>, 19/04/22
- [14] Yamashita, R., Nishio, M., Do, R., Togashi, K., Convolutional neural networks: an overview and application in radiology, Insights into Imaging, 9, pages 611–629 (2018)
- [15] Karras, T. et al., Alias-Free Generative Adversarial Networks, NVIDIA, 23 Jun 2021, <https://arxiv.org/abs/2106.12423>
- [16] Quickstart Tutorial, 2022, PyTorch, [https://pytorch.org/tutorials/beginner/basics/quickstart\\_tutorial.html](https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html), 19/04/22
- [17] Tzakos, A., Grace, C., Lukavsky, P., Riek, R., NMR Techniques for very large Proteins and RNAs in Solution, Annual Reviews, 9 June 2006, Vol. 35:319-342
- [18] Carroni M., Saibil H., Cryo electron microscopy to determine the structure of macromolecular complexes, Methods, 2016;95:78-85. doi:10.1016/jymeth.2015.11.023

- [19] Ramaswamy, V., Musson, S., Willcocks , C., Degiacomi, M., Deep Learning Protein Conformational Space with Convolutions and Latent Interpolations, Phys. Rev. X 11, 011052 – Published 15 March 2021
- [20] Molearn, 03/12/21, Github, <https://github.com/Degiacomi-Lab/molearn>, 19/04/22
- [21] Conv1d, 2019, PyTorch, <https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html>, 19/04/22
- [22] TORCH.NN, 2019, PyTorch, <https://pytorch.org/docs/stable/nn.html>, 19/04/22
- [23] Shen, M., Sali, A., Statistical potential for assessment and prediction of protein structures. Protein Sci. 2006;15(11):2507-2524. doi:10.1110/ps.062416606
- [24] Sali, A. and Blundell, T., Comparative protein modelling by satisfaction of spatial restraints, Journal of molecular biology, vol. 234, no. 3, pp. 779–815, 1993