

# Structure-conditioned Graph Generative Models for *De Novo* Drug Design

Charles Harris

`charles.harris17@imperial.ac.uk`

Bioinformatics and Theoretical Systems Biology Project

MSc Bioinformatics and Theoretical Systems Biology  
Imperial College London

Supervisors:

Professor Michael Bronstein (Imperial, Twitter),  
Professor Bruno Correia (EPFL)

Word count: 5,490

September 2021

## Abstract

The early stages of drug development usually revolve around the discovery of a molecule with high binding affinity to a target protein. Traditionally, this has been done with docking when a three dimensional structure of the target is available. Machine learning has recently been proposed as an alternative; however, almost all machine learning-based molecular generation methods to date fail to make molecules that bind well to targets. We postulate that this is primarily due to the lack of structural information used when generating candidate molecules or, when used, the protein representation is insufficient to capture the complex structural and chemical nature of the target. In this work we take established methods in molecule generation but condition the generative stages with information rich, geometric deep learning based descriptors of molecular surfaces in protein pockets. We show that while conditioning the generative process with pocket information does improve our objective function during training and validation, this does not reliably translate into generating molecules with better binding scores when assessed via docking. This suggests that better training criteria and techniques will be needed for these methods to produce drug-like molecules that bind effectively to a target.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Background . . . . .	4
1.1.1	Graph Neural Networks . . . . .	4
1.1.2	Variational Auto Encoders . . . . .	4
1.1.3	Deep Graph Generation . . . . .	4
1.1.4	dMaSIF . . . . .	6
1.1.5	Attention . . . . .	6
1.2	Related work . . . . .	6
1.3	Aims . . . . .	7
<b>2</b>	<b>Methods</b>	<b>7</b>
2.1	Dataset . . . . .	7
2.2	Molecule representation . . . . .	7
2.3	Graph Generation . . . . .	7
2.3.1	Pocket Attention . . . . .	9
2.4	Training . . . . .	10
<b>3</b>	<b>Results</b>	<b>11</b>
3.1	Training . . . . .	11
3.2	Generated molecules . . . . .	12
3.3	Docking . . . . .	12
<b>4</b>	<b>Discussion</b>	<b>15</b>
4.1	Generated molecules . . . . .	15
4.2	Limitations of training objective . . . . .	16
4.3	Limitations of docking assessment . . . . .	16
4.4	Limitations of pocket attention . . . . .	16
<b>5</b>	<b>Conclusion</b>	<b>17</b>
<b>6</b>	<b>Acknowledgements</b>	<b>17</b>
<b>Appendices</b>		<b>23</b>
<b>A</b>	<b>Calculating Attention</b>	<b>23</b>
<b>B</b>	<b>Training and model implementation</b>	<b>23</b>

# 1 Introduction

Drug design is the iterative process of optimising a molecule to have a set of specific properties and a desired therapeutic effect. This is made difficult by the large search space of possible drugs (Polishchuk et al. 2013) and the discontinuous nature of the optimisation functions (Stumpfe & Bajorath 2012). Furthermore, there are many criteria that need to be optimised in order for a molecule to be considered a suitable drug, ranging from pharmacokinetics (Bertrand et al. 2017) to toxicity (Yang et al. 2018), but the early stages of drug development usually focuses on the discovery of a suitable molecule with high binding affinity to a target we wish to modulate. Traditionally, this has been performed by Virtual Screening, where we search large libraries of commercially-available compounds to assess their potential binding to a target, usually via ligand docking if a structure is available (Muegge & Oloff 2006).

Many have proposed machine learning as a possible alternative to human-made or rule-based design strategies (Venkatasubramanian et al. 1994, Besnard et al. 2012, Hussain & Rea 2010, Gaudelot et al. 2021), which have their own set of biases, at the early stage of the drug design process. Research in the field of Deep Learning (LeCun et al. 2015), in particular Geometric Deep Learning (GDL) (Monti et al. 2017), has begun to focus on how we can endow machine learning architectures with properties conducive to the study of (bio-)molecules (Atz et al. 2021). For example, respecting the symmetries seen in three-dimensional (3D) molecules (Fuchs et al. 2020) and higher-order interactions in molecules (Bodnar et al. 2021). All of this culminated in the recent advances in protein and RNA structure prediction (Jumper et al. 2021, Chowdhury et al. 2021, Townshend et al. 2021), which suggests that these methods have the capability to learn fundamental principles of protein structure beyond human understanding. However, the same leap in performance has yet to be seen in other areas, such as structure-based drug design.

Early attempts to generate molecules used SMILES string representations (Kusner et al. 2017, Gómez-Bombarelli et al. 2018) with an increasing focus on graphs (Liu et al. 2018, Imrie et al. 2020, Simm et al. 2020, Satorras et al. 2021), which are a natural way of representing a molecule. However, very few methods have attempted to include any 3D structural information of the target protein as prior information and those that have are limited by poor representations of protein structure, in particular surface chemistry (Richards 1977), despite its importance for generating potent and selective molecules. Furthermore, previous work has not conditioned key stages of the generative process on structural information when used (Imrie et al. 2020). It is not surprising then that many methods produce molecules with binding affinities worse than the examples used in training and available in public databases of compounds (Cieplinski et al. 2020).

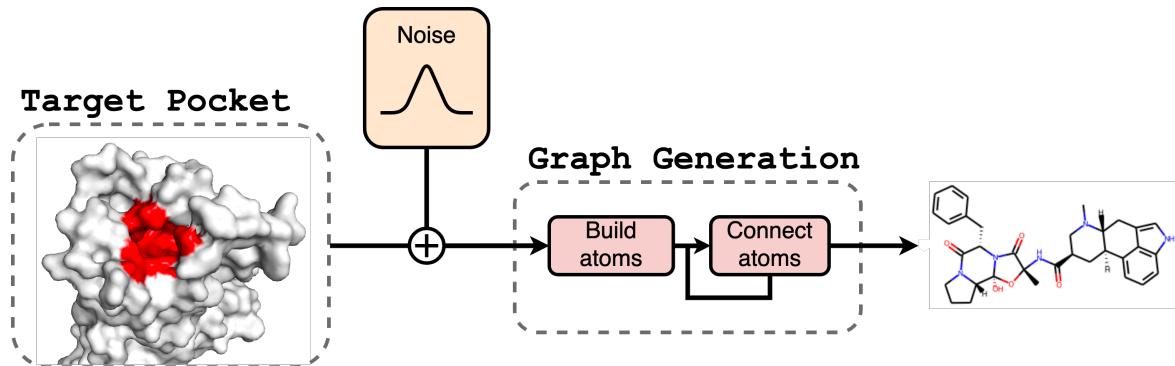


Figure 1: Overview of our approach. We take a surface representation of the target pocket and random noise (to allow diversity in the generated molecules) and perform an iterative graph generation loop to construct a candidate molecule.

In this work, we aim to address many of the issues in previous approaches by building our generation process around the target pocket of interest (Figure 1). This is done with highly expressive and efficient GDL based descriptors of the protein surface. Furthermore, we ensure that every stage of the generative process is conditioned on the prior information provided by the structure to guide generative decisions.

## 1.1 Background

### 1.1.1 Graph Neural Networks

Graph Neural Networks (GNNs) extend the traditional neural network architecture to operate over graphs. They are an example of GDL (Bronstein et al. 2017) as they exploit the relational aspect of the graph structure as an inductive/geometric prior (Battaglia et al. 2018). Formally, we define a graph  $G = \{V, E\}$  as a set of nodes  $v \in V$  and edges  $uv \in E$ . Node  $v$  has  $F$  dimensional features  $h_v \in \mathbb{R}^F$ , which is stored in a feature matrix  $\mathbf{H} \in \mathbb{R}^{|V| \times F}$ . For a multi-relational graph, the edges can be stored in an adjacency tensor  $A \in \mathbb{E}^{|V| \times |R| \times |V|}$ , where  $R$  is the set of relations (e.g bond types). The neighbours of node  $v$  are denoted as  $\mathcal{N}_v = \{w | uw \in E\}$ .

All flavours of GNNs (Bronstein et al. 2021) are based on the basic message-passing principle (Figure 2a):

$$m_v^{t+1} = \sum_{u \in \mathcal{N}_v} c_{uv} \psi(h_v^t)$$
$$h_v^{t+1} = \phi(h_v^t, m_v^{t+1})$$

where the current graph with node features  $h^t$  is transformed into  $h^{t+1}$  at the next layer,  $\phi$  and  $\psi$  are Multi Layer Perceptrons (MLPs) and  $c_{uv}$  is the weight of the edge connecting nodes  $u$  and  $v$ .  $m_v^{t+1}$  are the messages sent to node  $v$  after some permutation-invariant aggregation operation (in this case  $\sum$ ) is applied to the neighbours.

A common problem with GNNs is that after each layer of message-passing, the node features are the result of the aggregation of an exponentially growing number of neighbours, but these still need to be represented in a fixed vector size, leading to loss of information. To help mitigate this, the Gated Graph Neural Network (Chung et al. 2014) uses a Gated Recurrent Unit (Cho et al. 2014) as the aggregation function. This introduces a notion of memory into the model and allows it to forget redundant information whilst remembering important information (e.g. whether the atom is part of a ring).

### 1.1.2 Variational Auto Encoders

Most generative models broadly fall into two classes; Variational AutoEncoders (VAEs) (Kingma & Welling 2013) and Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). VAE (Figure 2b) based approaches have shown the most promise for molecular graph generation (Liu et al. 2018, Imrie et al. 2020), hence this project focuses on VAEs. Broadly, a VAE consists of an encoder-decoder setup (where both are neural networks) which takes some input data  $X$  and encodes it into a vector representation  $\mathbf{Z} \sim q_\phi(\mathbf{Z}|X)$ , called the *latent representation* or *embedding*. The decoder network then reconstructs the input from the latent representation such that  $X^* \sim q_\theta(X^*|\mathbf{Z})$ . Once trained, we can generate synthetic examples by sampling random  $Z$  from the latent space and passing this through the decoder.

We train a VAE by minimising the Evidence Lower-Bound Loss (ELBO) (Kingma & Welling 2013). This consists of a reconstruction term that ensures that the latent vector  $Z$  is sufficient for reconstructing the original input  $X$  and a KL divergence term that ensures the latent space is constrained to a normal distribution (so it is not sparsely populated with meaningful samples):

$$L = \text{Reconstruction Loss} + \text{KL Divergence}$$

$$L = \mathbb{E}_{\mathbf{Z} \sim q_\phi}(p_\theta(X|\mathbf{Z})) + KL(\mathcal{N}(0, I), \mathcal{N}(\mathbf{Z}_\mu, \exp(\log(\mathbf{Z}_\sigma))))$$

Note, there are many assumptions made when constructing  $Z$  in a VAE, please see Kingma & Welling (2013) for details.

### 1.1.3 Deep Graph Generation

Similar to a VAE, a graph VAE (Figure 2c) takes a graph as input, embeds it to a latent space  $Z$  (using a GNN) and using a decoder attempts to reconstruct the original graph. Early work decoded the graph all-at-once (Kipf & Welling 2016), but this was shown to be ineffective for molecular generation as node independence is assumed, whereas atoms tend to be connected in specific configurations (Li et al. 2018,

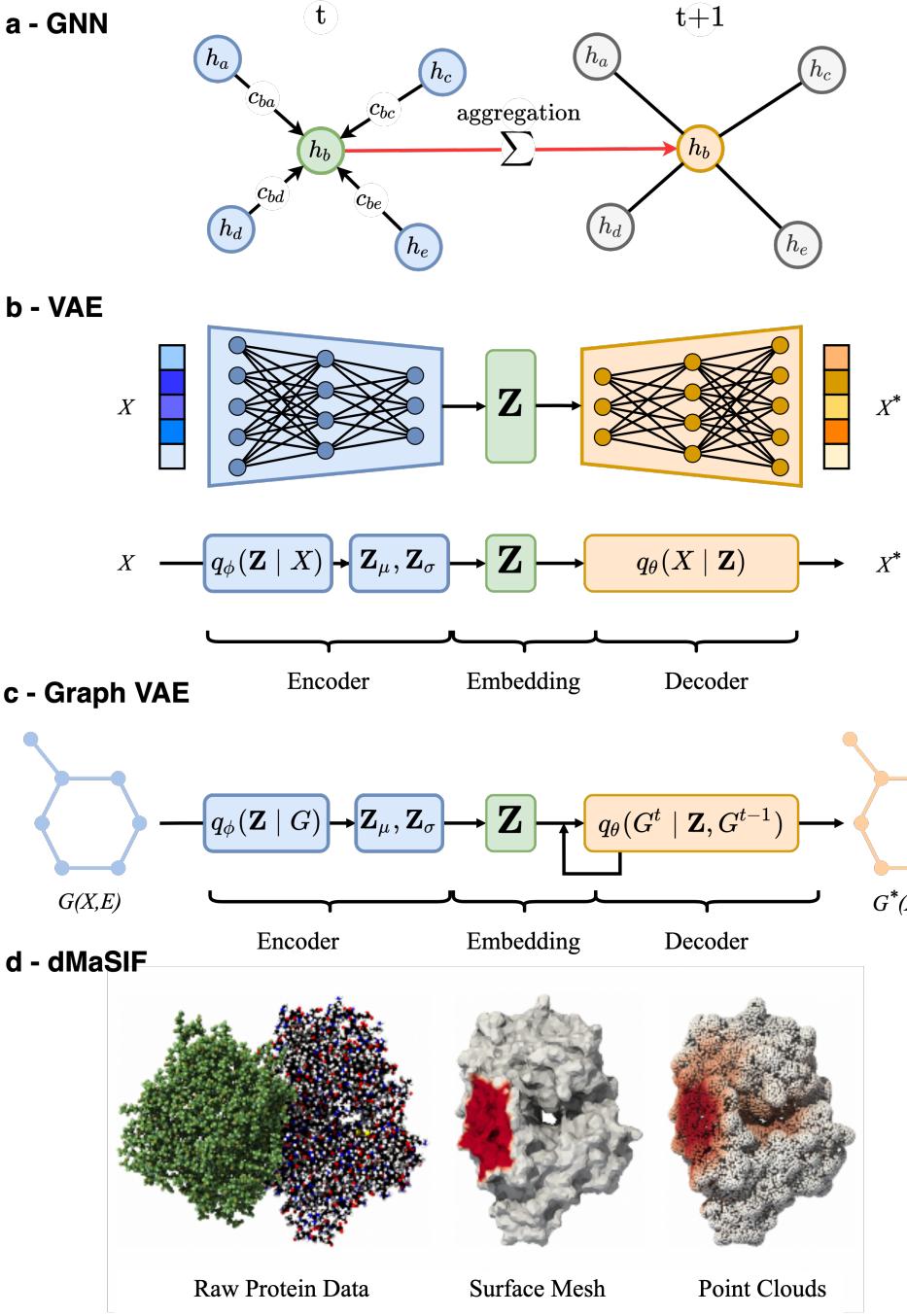


Figure 2: (a) For each layer in the GNN, the features of a particular node (green) are updated by sending the features of the neighbouring nodes (blue) as messages. These are aggregated together (red) to update the node features in the next layer (orange). (b) An overview of a VAE. The encoder embeds  $X$  into a low dimensional latent representation  $Z$  and the decode reconstructs the original data from  $Z$  to make  $X^*$ . Both the encoder (blue) and decoder (orange) are neural networks (e.g. MLPs or GNNs). (c) Overview of an autoregressive graph variational autoencoder. (d) An overview of the dMaSIF representation of protein surfaces (Left) Raw atomic protein data of two proteins forming an interaction. (Center) A surface rendering of the binding interface (red). (Right) How dMaSIF sees proteins. The surface is represented as a set of point clouds with chemical and geometric features. Quasi-geodesic convolutions transform these features into interaction embeddings/fingerprints. Another neural network can be used to predict for a downstream task (in the case of this example protein binding interface prediction). Renderings were taken from Sverrisson et al. (2021) with the permission of the author.

Liao et al. 2019). Recently, node-by-node, or autoregressive approaches have shown better promise for molecular generation (Liu et al. 2018, Imrie et al. 2020). Here, the encoder derives a latent representation for each node in the graph  $z_v \in Z$  and the decoder iteratively reconstructs the graph via a series of partial graphs  $G^0, G^1, \dots, G^T$  where each partial graph is conditioned on the previous graph and the latent representations  $G^{t+1} \sim q_\theta(G^{t+1}|G^t, \mathbf{Z})$  (Li et al. 2018).

#### 1.1.4 dMaSIF

dMaSIF (differentiable molecular surface interaction fingerprinting) (Sverrisson et al. 2021) is a GDL method for learning the interaction patterns on protein surfaces and how this governs their interaction with other molecules. It is a faster and more lightweight version of the original MaSIF (Gainza et al. 2020). Both methods rely on the hypothesis that interactions between proteins and other biomolecules can be predicted using the topology and distribution of chemical features on protein surfaces.

In the dMaSIF model (Figure 2d), these two features (chemical and geometric) are represented as point clouds on the protein surface over which geodesic (the shortest path along a surface) convolutions are performed. This transforms the point cloud features on the surface to make the "surface interaction fingerprints", which are highly expressive for the purpose of interaction prediction as they capture the local chemical and geometric properties of protein surfaces (e.g. in a binding pocket).

#### 1.1.5 Attention

Traditional convolutions/aggregations on objects such as graphs and text suffer from the assumption that the relationship between two related pieces of information must share the same weight, while this is rarely the case in real world data. The attention mechanism has been proposed to alleviate this problem. This allows us to calculate how much *attention* we should pay to related information when making predictions for a downstream task (Veličković et al. 2018). Attention has been very beneficial for modelling systems where we need to consider both long-range dependencies and the relationship between a large number of entities, but only an uncertain subset are vital to model the dynamics (e.g. structure prediction (Jumper et al. 2021, Chowdhury et al. 2021)). Furthermore, protein language models that use attention have been shown to be *unsupervised* learners of protein structure (Vig et al. 2020, Rao et al. 2020), highlighting their relevance in structural bioinformatics.

More formally, attention can calculate how much a *query* (e.g. the atoms in a ligand) should attend to a particular *context* (e.g. parts of the protein pocket) instead of assuming that all relationships have a fixed or predetermined weight. Note, in models as Graph Attention Networks (Veličković et al. 2018) and Transformers (for sequences) (Vaswani et al. 2017) the query and context are the same, this is called self-attention. We explain how attention is calculated in Appendix A.

## 1.2 Related work

Early work on generating SMILES strings with VAEs showed that they can covert discrete molecule representations to and from a continuous representation in the form of the latent space. This allows for efficient exploration and optimisation of new molecules by searching through the this trained latent space (Kusner et al. 2017, Gómez-Bombarelli et al. 2018).

Constrained Graph Variational Autoencoders (CGVAEs) (Liu et al. 2018) inspired much of the VAE-based work in the field. CGVAE introduced a helpful valency constraint procedure which masks possible design choices during generation which would lead to a violation of atom valency requirements. Most other work extend this by tailoring the model to a specific problem in drug design and adding some limited pocket information. DeLinker (Imrie et al. 2020) uses the distance and angle between fragments with known poses to generate carbon based linkers. The improved DEVELOP (Imrie et al. 2021) instead uses 3D pharmacophoric information encodes with a 3DCNN (a type of neural network designed for computer vision) for fragment linking and scaffold elaboration. DeLinker is obviously limited in that it has no knowledge of the surface chemistry when designing a linker (and limited geometric constraint) and DEVELOP uses information only available for a limited number of targets. Furthermore, the 3DCNN used in DEVELOP is not invariant to the rotational symmetry, a key requirement to modelling protein pockets.

Other approaches have also used Generative Adverserial Networks (Prykhodko et al. 2019), Normalising Flows (Satorras et al. 2021, Zang & Wang 2020) and Reinforcement Learning (RL) (Olivecrona et al. 2017, Blaschke et al. 2020, You et al. 2018, Atance et al. 2021). A detailed systematic review

of graph generation strategies are available in Guo & Zhao (2020) and specially for *de novo* molecular design in Xia et al. (2020) and Engkvist et al. (2020).

### 1.3 Aims

The aims of this works is as follows. (i) Investigate whether adding prior knowledge on the target protein improves the accuracy of the molecule generation process and (ii) validate the hypothesis that dMaSIF representations of protein surfaces help us to achieve achieve this. (iii) See how far molecular generative models can be taken without the added information from modelling the protein-ligand fit. (iv) See whether incorporating an attention mechanism between the ligand and the protein allows us to better leverage information about the nature of the ligand-protein interaction without explicitly modelling binding pose.

## 2 Methods

### 2.1 Dataset

We use the refined version of the PDBBind (Liu et al. 2015) dataset. This contains 3706 protein-ligand pairs of co-crystallised structures (although this work does not use the 3D coordinates). For each protein-ligand pair, we extract the binding pocket by only considering atoms in the protein that are within 5 Å of the bound ligand (as to include the entirety of the surround pocket and not just the binding interface for the sample). We split the dataset into train, validation and testing datasets using 80%, 10% and 10% of the data respectively.

### 2.2 Molecule representation

All molecules are converted into a graph representation with atom labels  $\mathbb{L}$  and bond types  $\mathbb{B}$  (Table 1) as one-hot encodings. Protein surfaces are represented using dMaSIF point clouds and embedded into a single vector  $P$  describing the whole pocket by dMaSIF quasi-geodesic convolutions (Sverrisson et al. 2021). dMaSIF also uses one-hot encodings of atom types in the protein. Therefore, no physico-chemical information about the atoms or bonds are given to the model (except valency constraints (see Section 2.3)). In the following sections, we will use the terms atom/node and bond/edge interchangeably depending on the context.

Feature	Set	Encoding
Atom type	$\mathbb{L} = \{C, H, O, N, Cl, Br, I, S, P\}$	one-hot
Bond type	$\mathbb{B} = \{\text{single}, \text{double}, \text{triple}, \text{aromatic}\}$	one-hot

Table 1: The sets of features used to represent a molecular graph

### 2.3 Graph Generation

For simplicity, we describe how we generate molecules once the model is trained first, then explain the training procedure in Section 2.4. We take inspiration from Liu et al. (2018) for the overall architecture as well as Li et al. (2018) and Imrie et al. (2020) for structural conditioning. We generate molecules node-by-node (Figure 3) using the following schema:

1. **Atom initialisation:** we initialise the set of atoms the molecule is to be made from in a similar way to CGVAE (Liu et al. 2018) except we add a new conditioning stage:
  - (a) **Node feature initialisation:** we sample a random variable from a Gaussian distribution  $Z \sim \mathcal{N}(0, I)$  to initialise the node features.
  - (b) **Atom conditioning (Figure 3a):**  $Z$  is then conditioned to the target structure by concatenating  $Z$  with the one-dimensional pocket embedding  $P$  (made with dMaSIF) and transforming this back to the target node feature dimensions with an MLP ( $\text{MLP}_c$ ):

$$Z_{\text{con}} = \text{MLP}_c([Z_{\text{init}}, P])$$

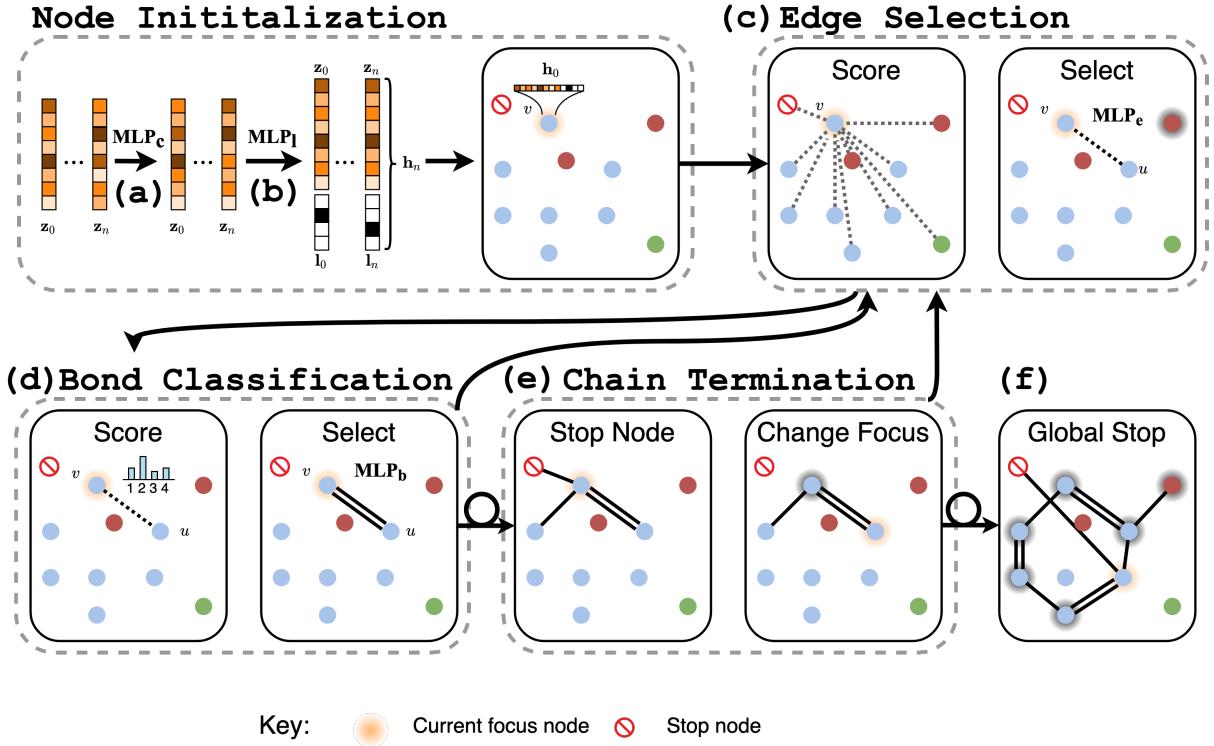


Figure 3: Overview of our molecule generation procedure. (a) Node representations are conditioned by information from the target pocket. (b) Node labels are predicted. We then enter an iterative loop between c-e to construct the molecule. (c) We score all possible choices of edges  $uv$  and select the highest probability. (d) New edges are classified into bond types. (e) If the stop node was selected, we terminate extention of the current branch and refocus onto another atom. (f) Once all possible focus atoms have been considered, the global stop node is selected and the process terminates. Looped arrows indicate that multiple iterations may take place before that path is chosen. Note, we are not showing the state encoding or pocket information here for simplicity. Adapted from Liu et al. (2018).

(c) **Atom classification (Figure 3b):** we then assign atom type labels  $l$  to each node where  $l_v \in \mathbb{L}$ . This is done by an MLP ( $\text{MLP}_l$ ) that takes the (pocket conditioned) node features as input and predicts atom types (carbon, oxygen etc) in the form of a one-hot encoding.

$$L = \text{MLP}_l(Z_{\text{con}})$$

This is concatenated to the node features such that  $h_v = [z_v, l_v]$  for all node features  $h_v \in H$ . From these, we can calculate the global features  $H_{avg}^t$  and  $H_{avg}^{init}$ , which are the average features of the connected nodes in the graph at generation step  $t$  and the average features of all nodes at initialisation respectively.

2. **Generation loop:** the model then performs an iterative loop to generate the molecule, this is done in a breath-first manner (i.e. building chains one at a time). A focus node  $v$ , from which we will branch off parts of the molecule, is chosen at random:

(a) **State encoding:** similarly to CGVAE (Liu et al. 2018), after every generation step with a new graph  $G^t$ , we delete  $H^{t-1}$  and recompute new node features taking into account their (possibly) changed neighborhood. This is done with a Gated Graph Neural Network (GGNN) ( $G_{\text{dec}}$ ):

$$H^t = G_{\text{dec}}(H^0, E^t)$$

Note that since we use the initial node features  $H^0$ , instead of  $H^{t-1}$ , the new node representations  $H^t$  are independent of the generation history of the graph.  $E^t$  are the edges in the graph at time step  $t$ . Note, we have not shown this in Figure 3 as we wish to focus on how the molecule is generated.

- (b) **Edge selection (Figure 3c):** to actually build the molecule, we need to decide which edges are going to be added between the atoms. This is framed as a choice of all possible nodes  $u$  we can choose from to add to the current focus node  $v$ . Each possible choice of edge  $uv$  is encoded in a feature vector  $\phi_{uv}^t$ :

$$\phi_{uv}^t = [h_u, h_v, t, d_{uv}, H_{avg}^t, H_{avg}^{\text{init}}, P]$$

where  $h_u$ ,  $h_v$ ,  $H_{avg}^t$ ,  $H_{avg}^{\text{init}}$  and  $P$  are as described above,  $t$  is the current time step and  $d_{uv}$  is the graph distance between  $u$  and  $v$  (allows us to construct rings of suitable size). Providing all of this information here means that the molecule generation procedure is conditioned on both the current partial graph *and* the target pocket which the molecule is being built into. Naively,  $\phi_{uv}$  can be thought of as representing how satisfied each atom is with its status in the current graph and its relationship to the pocket. We select an edge  $uv$  by passing all the feature vectors for edge choices  $\phi_{uv}^t \in \Phi^t$  to an MLP ( $\text{MLP}_e$ ):

$$uv = \text{MLP}_e(\Phi^t)$$

- (c) **Bond classification (Figure 3d):** the bond type  $b$  of the new edge is then decided where  $b \in \mathbb{B}$ . We provide the single vector of the chosen edge  $\phi_{uv}^t$  to another MLP ( $\text{MLP}_b$ ).

$$b = \text{MLP}_b(\phi_{uv}^t)$$

- (d) **Chain termination (Figure 3e):** in order to control the generative process (and terminate chain extension at appropriate points) we use a special stop node. Once selected we remove node  $v$  from the list of possible future focus nodes and refocus onto a new node to continue building.

3. **Global stop (Figure 3f):** once all possible candidate focus nodes have been exhausted, the generation process stops.

We employ the same valence masking procedure that was introduced in CGVAE (Liu et al. 2018), which ensures atoms/certain bond types are added only if the valence shells are not already satisfied.

### 2.3.1 Pocket Attention

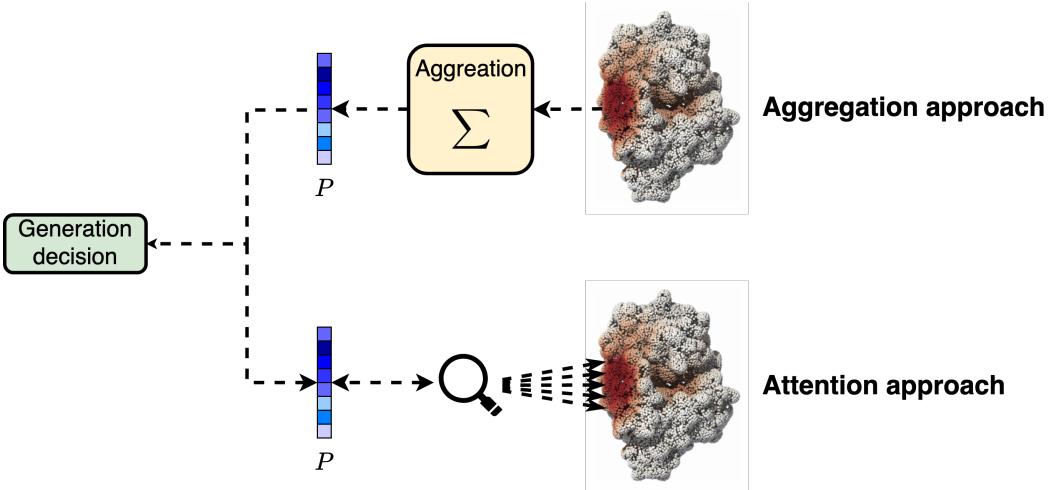


Figure 4: The differences between the aggregation and attention approaches. In aggregation, the whole of the protein pocket is aggregated by some permutation-invariant function (e.g. `sum` or `mean`) into a vector of fixed size. In the attention approach, the model is able to consider all points in the pocket, compute attention weights and perform a weighted sum (as in Figure 9) to make the a protein vector to guide each decision individually. Protein rendering taken from Sverrisson et al. (2021) with the permission of the author.

We propose two kinds of models. The first is as described above and uses a single vector  $P$  to embed the whole of the target pocket.  $P$  is calculated by aggregating all of the information from the dMaSIF point clouds into a single vector representation. As with the aggregation of messages sent in GNNs (Veličković et al. 2018, Corso et al. 2020), this aggregation process introduces over-smoothing of high-frequency information and does not allow for anisotropy (i.e. all positions in the pocket are considered equally).

Instead, we propose a second model that uses an attention mechanism between the ligand and the pocket to allow the generation procedure to be guided by the local context within certain regions of the pocket (Figure 4). This is implemented as explained in Appendix A where the query is either the atom conditioning (Step 1b), edge selection (Step 2b) or bond classification (Step 2c) stages. The context is always the set of dMaSIF point clouds describing their local physio-chemical environment. The intuition here is that this allows us to make decisions based on where the model thinks an atom might bind *without* explicitly performing docking in anyway. To the best of our knowledge, this is the first time an attention mechanisms has been used in this way and is an attempt at *unsupervised* learning of ligand docking.

## 2.4 Training

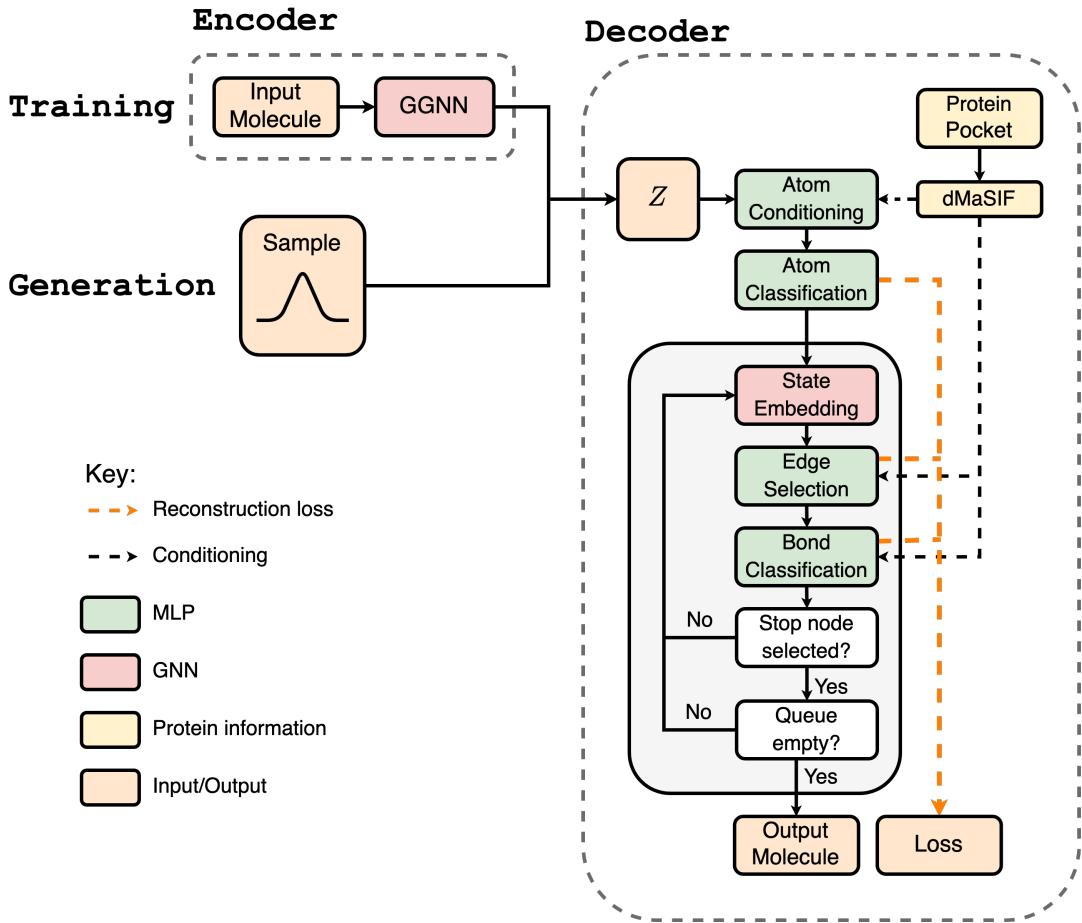


Figure 5: Overview of our VAE architecture. During training, the latent vector  $Z$  is constructed by the encoder and during generation, we sample  $Z$  from a Gaussian distribution. The decoder is a schematic representation of Figure 3. Conditioning vectors can be constructed by either the aggregation or attention approach. Not depicted here is the KL divergence of the loss function.

We train our model in a VAE manner with a graph encoder-decoder setup (Figure 5). The encoder is a GGNN ( $G_{enc}$ ) which embeds the training graph into  $Z$ . The decoder then takes  $Z$  and follows the same steps as in Section 2.3 to attempt reconstruction of the input molecule. Figure 5 shows clearly where the pocket information conditions generative decisions. Note, during generation, we do not use the encoder and instead sample a random  $Z$  from a Gaussian distribution as the seed for molecule generation (as described in Section 2.3).

Our aim is not to simply train a VAE that learns to reproduce the distribution of molecules seen in the training dataset (Yang et al. 2020). Instead, we want design choices to be principally governed by the target pocket and the latent vector  $Z$  to allow us to have diversity in the generated molecules. Therefore, we intentionally constrain the size of  $Z$  to be a low-dimensional vector such that it does not contain all the information to reconstruct the molecule and ignore the pocket information.

Our model cannot be trained end-to-end with a single, global loss function as this aggregates all of the design decisions into one long gradient for back propagation, leading to the optimization of the most common case (i.e. adding carbon atoms everywhere possible). This is called mode-collapse and is a common problem in many generative models, especially GANs (Srivastava et al. 2017).

Instead, we use the Teacher Forcing approach introduced in Li et al. (2018) which allows for individual supervision of each generation decision. We construct a series of ‘ground-truth’ construction steps in the form of a list of partial graphs ( $G^{*0}, G^{*1}, \dots, G^{*T}$ ), where  $T$  is the number of edges in the final graph. We construct this list via a breath-first search (Beamer et al. 2012) of the target molecule. During training, the model is ‘forced’ through the correct sequence of design choices and then we train my maximising the probability the model would have made the correct decision when generating the molecule during testing.

In practice, this is done by minimising the negative log probability that our model made the correct decisions towards the ground truth steps given the conditioning information. This is done both at the atom classification step, where the reconstruction loss for the node labels ( $L_l$ ) is;

$$L_l = \sum_{u \in V} -\log p(x_u = x_u^* | [Z, P])$$

and the edge selection and classification steps ( $L_g$ ) to made the new graph  $G^t$  is:

$$L_g = \sum_t -\log p(G^t = G^{*t} | [G^{*t-1}, P])$$

Note, the edge selection and bond classification steps are supervised seperated with two loss functions (as shown in Figure 5)

Our overall loss function (the function we aim to minimise during training via stochastic gradient decent and backpropagation (LeCun et al. 2015)) is therefore the sum of all reconstruction terms and the KL divergence (see Section 1.1.2):

$$L_{total} = L_l + L_g + L_{KL}$$

We detach vectors (breaks the flow of information through the model) after each generation choice as to improve the speed and memory efficiency of the model. See Appendix B for further details on training and model implementation.

## 3 Results

### 3.1 Training

Figure 6a shows the evolution of training and validation loss overtime for a typical training run. The validation loss follows closely behind the training loss, suggesting that out model is not over-fitting to the training data (Yang et al. 2020).

To assess the impact of the pocket information on accurately reconstructing the molecule during training (a different task from generating potent new molecules), we perform an ablation study where we remove certain features of the model and compare the validation losses (Figure 6b). Considering each model took 1-2 days of training, we use 3 runs for reach category to get a consensus. We train two models as described in Sections 2.3 and 2.4 either with and without the pocket attention mechanism (Figure 4). Adding pocket attention resulted in a mean validation loss of 82.6 (arbitrary units) versus 81.4 without pocket attention (i.e. aggregation approach). Given the negative result here, we have decided not to evaluate any more models with pocket attention (both for computational reasons and brevity). However, we do discuss future work in the Discussion. For the ablation study, we evaluate two (aggregation approach) models where we remove pocket information from a part of the generative process to interpret importance. Removing atom conditioning (Step 1a) and edge section conditioning (Steps 2b and c) results in validation losses of 90.2 and 93.0 respectively, a marked decrease in performance. Finally, removing all the pocket information entirely (giving an architecture similar to CGVAE (Liu et al. 2018)) results in a validation loss of 96.7.

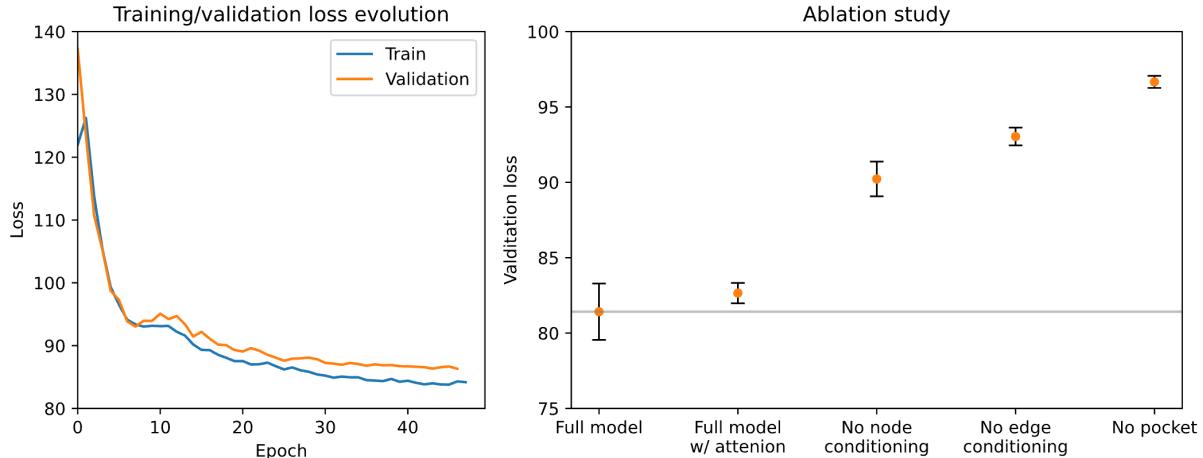


Figure 6: (a) Training and validation evolution over training. (b) Ablation study: certain parts of the model architecture are removed and we see the effect on validation loss (lower is better). Orange point is the mean, error bars are 95% confidence intervals and the grey line is the baseline performance with all the pocket information.

### 3.2 Generated molecules

As a demonstration of our model’s ability to reconstruct ligands from a known point in the latent space, we encode and decode a random selection of molecules not seen during testing (Figure 7a). Note, this is not *de novo* synthesis as the encoder ( $G_{enc}$ ) gives the correct  $Z$  to seed reconstruction the molecule. Qualitative inspection of the majority of our generated molecules shows that reconstruction accuracy is on par with other VAE based methods (Liu et al. 2018, Li et al. 2018, Imrie et al. 2020). Common qualitative issues during testing are (i) a large proportion of non-carbon atoms in the molecule, which can lead to (ii) early termination of generation (resulting in very small molecules), (iii) very large ring structures and (iv) molecules that are likely non-synthesiseable (Figure 7b).

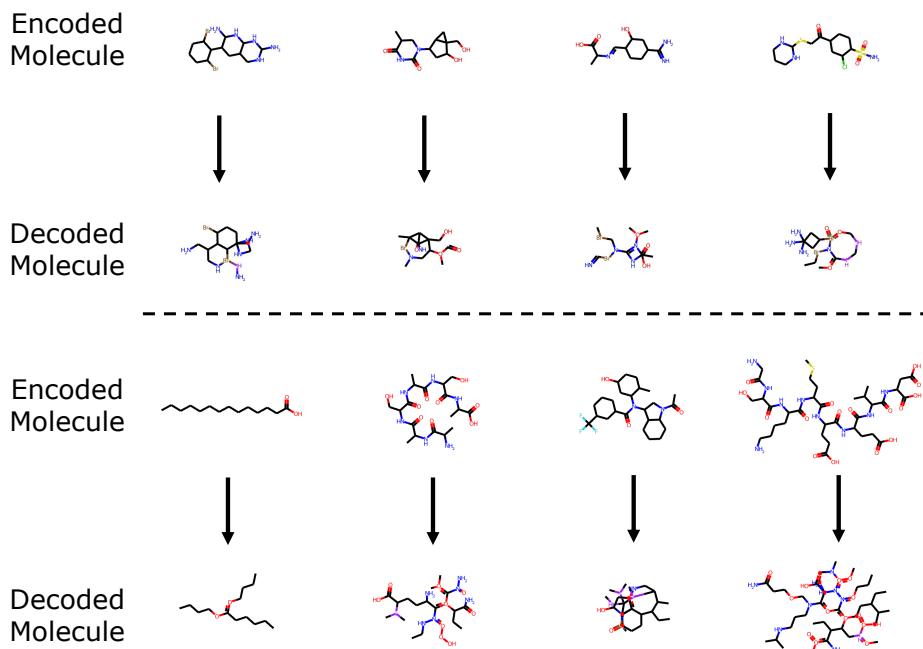
### 3.3 Docking

The aim of this work is to show that dMaSIF fingerprints can be used to generate new molecules with high binding affinity for a target. Therefore, purely assessing our loss function of reconstructing known molecules is not a robust assessment for our models task. To study this, we perform docking of our *de novo* generated molecules to two common, G-protein coupled receptor (GPCR) targets; (i) 5-HT1B (PDB: 4IAQ), a serotonin receptor (Wang et al. 2013), and (ii) ACM2 (PDB: 3UON), a muscarinic acetylcholine receptor of the autonomic nervous system (Haga et al. 2012). For both targets, the aim is usually to design an agonist (increases activation) or antagonist (blocks activation) that selectively binds to the main receptor binding site (de Boer & Koolhaas 2005, Bailey & Hay 2006). We use the same procedure for docking as in Cieplinski et al. (2020), which uses a derivative of AutoDock Vina (Trott & Olson 2010) to dock generated molecules to a target. 250 candidate molecules are generated per experiment per target and are either fully conditioned on the pocket ('Full model') or with no structural information at all ('No pocket').

The results from the docking experiment, including selected measures of drug-likeness of the generated molecules (QED, ring count and LogP) are assessed in Figure 8a. Both targets, the mean docking score is not increased significantly by adding pocket information and virtually all molecules have a docking score far below the human designed examples in the PDB files (except a few outliers). We also break down individual components of the docking score (repulsion and hydrogen bonding scores). Very little difference is seen in repulsion of the generated molecules for both models but the model with no pocket information generated molecules with significantly higher mean satisfaction of hydrogen bonds.

For the drug-likeness metrics, there appears to be no significant correlation with positive or negative results when we add/remove pocket information, suggesting that this is determined by the stochastic path taken and local minima found during training rather than the architecture. More evaluations may be needed. It should also be stressed that our model was not directly supervised to optimise these

a



b

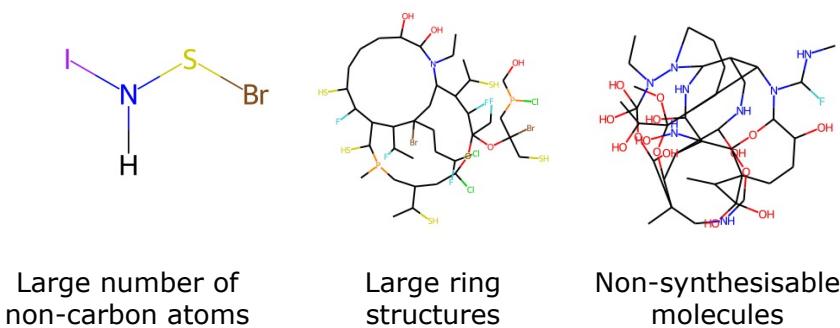


Figure 7: (a) A random selection of test molecules encoded and decoded by the VAE model with pocket information. (b) Examples of common qualitative errors seen during testing.

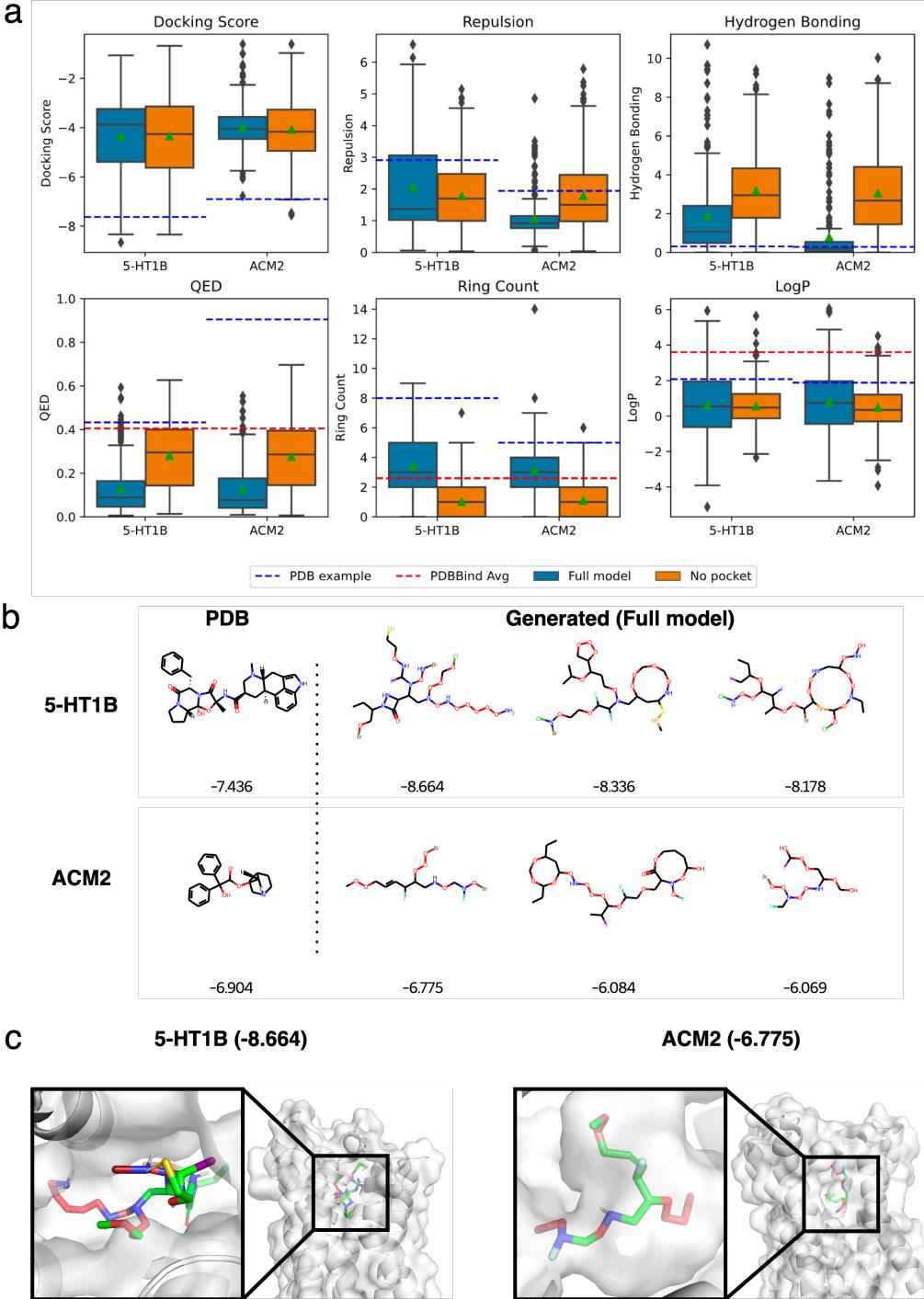


Figure 8: (a) Results of docking scores and drug-like characteristics from each model. Edges of box are the quartiles, the black line in the box is the median and the green triangle is the mean. The whiskers show the rest of the distribution and 'outliers' (black diamond). (b) Three of the highest scoring binders for both targets with an example drug candidate from the PDB shown for reference. Numbers under molecules are docking scores. (c) Docking pose for molecule with best docking score for both targets visualised, number in brackets in docking score.

Model	5-HT1B	ACM2
Ours (Full model)	-4.363 (0.915)	-3.981 (0.919)
Ours (No pocket)	-4.336 (0.950)	-4.080 (0.952)
CVAE (Gómez-Bombarelli et al. 2018)	-4.955 (0.901)	-4.836 (0.905)
GVAE (Kusner et al. 2017)	-4.955 (0.901)	-5.422 (0.898)
REINVENT (Olivecrona et al. 2017)	<b>-9.774 (0.506)</b>	<b>9.775 (0.467)</b>

Table 2: Mean AutoDock predicted score of all models for both targets (lower is better). Number in brackets is the diversity of the generated molecules (as calculated in Cieplinski et al. (2020)). All data (expect for our model) taken from Cieplinski et al. (2020).

properties in any way. However, QED scores and rings counts of the generated molecules are significantly different for both models. LogP scores are relatively invariant to the addition of pocket information. All mean scores for drug-likeness are worse than the mean of the training dataset.

We also compare our mean docking score against other methods in the literature. Specifically, two VAE based methods, Chemical Variational Autoencoder (CVAE) (Gómez-Bombarelli et al. 2018) and Grammar Variational Autoencoder (GVAE) (Kusner et al. 2017), as well as REINVENT (Olivecrona et al. 2017), which is a Recurrent Neural Network (Mikolov et al. 2010) trained using RL to optimise molecular properties. Note, none of these approaches include target information, rather they learn to reproduce the distribution seen in their training dataset (usually ZINC (Irwin et al. 2020), which is orders of magnitude larger than our dataset). Our model under performs against both targets, both with and without pocket information, with REINVEST achieving best results on both tasks (Table 2).

## 4 Discussion

In brief, the addition of pocket information to the generative process significantly and reliably improves performance with regards to our loss function during validation (Figure 6b) but this does not translate into generating molecules *de novo* for the task of stable binding, at least when assessed via docking (Figure 8a and b). This could be due to a number of reasons; (i) the training scheme is insufficient to tune the properties of the molecules for the desired effect (strong affinity/complementarity to the target), (ii) we are limited by the quantity and quality of our training data, (iii) the docking is not a reliable assessment of the diverse molecules generated by both models. It is likely all of these factors play a part in our results.

### 4.1 Generated molecules

Our model is no doubt limited by the size and quality of our training dataset, a fundamental limitation to the effective training of deep learning models (Sun et al. 2017). We use a small dataset ( $\sim 4,000$  protein-ligand pairs) when compared to similar methods (e.g. CGVAE used 134,00-250,000 molecules (Liu et al. 2018)) and are a sample of all ligands seen the PDB. Therefore, it is not surprising that our generated molecules represent those seen in the PDDBind dataset (Liu et al. 2015) (long, branching aliphatic chains) rather than the two human designed drug candidates seen in Figure 8b (complex aromatic ring systems).

Larger datasets were out of the scope of this project given the time constraints for model training and assessment, however, we discuss alternatives as possible future work. The Database of Useful Decoys-Enhanced (DUDE) (Mysinger et al. 2012), which contains 22,886 active drug-target pairs and 50 decoys for each active molecule will likely improve results during training and the drug-likeness of our molecules. Difficultly has been had using DUDE for machine learning in the pasts as models tend to learn the rules for negative creation (Yang et al. 2020). However, we propose that, if supervised carefully, the use of decoys during training should teach a future model the negative design principles needed to increase specificity to the target of interest whilst reducing undesired activity on structurally related proteins. Future work could also see if the use of ZINC (a database of 1.4 billion drug-like compounds compounds (Irwin et al. 2020)) could improve results via pre-training (Erhan et al. 2010) and then fine-tuning on examples for which we have the structure of the target.

While it is not attempted here, jointly predicting both the graph structure and the binding pose simultaneously will likely improve performance as generation decisions can explicitly be guided by the protein-ligand fit. This could be done using the binding pose data from either PDDBind or DUDE.

However, it is very unlikely that ligand poses will be accurately predicted initially, meaning this will likely need to be modelled via an iterative process. A possible implementation is a two-stage approach where a binding pose hypothesis is first predicted in vector representation space (via a very deep equivariant neural network (Fuchs et al. 2020)) then converted into coordinates via a GDL method (similar to how AlphaFold2 first generates a structural hypothesis in vector space with the 'EvoFormer' module then converts this into Cartesian coordinates with the structure module (Jumper et al. 2021)).

## 4.2 Limitations of training objective

Fundamentally, our approach (and other VAE-based methods that try to reason over structure (Imrie et al. 2020, 2021)) is limited by the fact that our training objective is to take the correct sequence of decisions to reconstruct a known molecule. While pocket information does aid in this task (Figure 6b), this is far removed from the more biologically important question of how to design molecules with strong and selective affinity to a target and drug-like properties. Furthermore, for computational reasons, we need to arbitrarily decide one path which describe the "correct" sequence of decisions to construct the molecule, ignoring all other possible construction paths even if they produce the same ligand (however, Liu et al. (2018) showed that this has a limited negative effect on performance). Unfortunately, this reflects the capabilities of the current incarnation of generative models. We explore possible alternatives.

Reinforcement Learning (RL) is a branch of machine learning where an agent learns to take actions in a environment to maximise some pre-specified reward (Kaelbling et al. 1996). Having seen considerable success in the domain of games (Mnih et al. 2013, Silver et al. 2017), in particular games that are highly dynamic and provide imperfect information for decisions (Vinyals et al. 2019), many are beginning to explore how deep RL techniques can be applied to tasks in drug design. To date, RL has mainly been used to fine-tune pre-trained generative models to generate *de novo* molecules with either (i) greater QED scores (Olivecrona et al. 2017), (ii) smaller or larger molecular weight (You et al. 2018) or (iii) maximal activity for a specific target via a Quantitative Structure Activity Relationship (QSAR) model (Atance et al. 2021). However, none of these methods refine the structure and dynamics of the ligand-protein fit by considering the whole protein (and any flexibility) during learning. Skwark et al. (2020) used RL to iteratively design a protein binder to have an increasingly stabilised interaction with the SARS-CoV-2 spike protein by minimising a standard protein free-energy function (ROSETTA (Alford et al. 2017)). We propose that similar strategies, while expensive, are applicable for creating novel and potent molecules (e.g. using AutoDock (Trott & Olson 2010)).

## 4.3 Limitations of docking assessment

Much can be said on the accuracy and reliability of docking (Chen 2015, Wang et al. 2016) and scores should be interpreted with a degree of caution. Indeed, during virtual screening, the success of docking is usually measured in the number of new and interesting scaffolds discovered for a target rather than the number of high affinity hits (Muegge & Oloff 2006). Therefore, it is possible that our method (compared to the no pocket model) is better at discovering novel chemical structures for the target of interest, but docking does not have sufficient resolution to show this.

Future work could try to identify whether our model has the creative ability to generate novel scaffolds that are complementary to the target beyond what is seen in the training dataset. This is could be done by comparing the Bemis-Murcko scaffolds (Bemis & Murcko 1996) of the training dataset versus those generated by the model (with and without pocket information) via t-SNE dimensionality reduction (Van der Maaten & Hinton 2008) or Tanimoto similarity (Butina 1999).

Our work could also easily be extended to perform linker design and scaffolds elaboration (as in Imrie et al. (2020) and Imrie et al. (2021)), which could prove a better assessment of dMaSIF as we are more concerned with the selection of appropriate functional groups instead of building a whole molecule *de novo*. This was not done here due to difficulty constructing the breath-first search paths between the partial and whole ligands.

## 4.4 Limitations of pocket attention

We propose that our relatively simple pocket attention mechanism fails to improve performance as it (i) does not have any notion of spatial reasoning for the point clouds on the protein surface (they are presented as a one-dimensional list with no coordinates) and (ii) does not leverage the symmetries present in proteins. This could be remedied by adding a positioning encoding mechanism, similar to that seen in the Transformer architecture (Vaswani et al. 2017) but extended to work in 3D, which would allow the

attention mechanism to infer spatial relationships between point clouds (e.g. the *spatial encoding* from Ying et al. (2021)). Alternatively, we could simply concatenate the Cartesian coordinates of the point clouds if we created an attention mechanism which is invariant to the various symmetries seen in proteins (similar to Invariant Point Attention from AlphaFold2 (Jumper et al. 2021)). Notably, symmetries in the SE(3) group (Fuchs et al. 2020), with comprises rotations and translations but not reflections (as to respect chirality). The use of a multi-headed attention mechanism would likely be beneficial modelling complex interactions between a protein and drug as each "attention head" is able to attend to different positions on the protein surface and learning different representation sub-spaces (Vaswani et al. 2017) ,

There is currently a large disparity between the number of molecules which we know are active for a particular target and molecules for which we have 3D coordinates of the bound molecule (Mysinger et al. 2012). Interestingly, when we visualise the attention weights between amino acids in protein language models (trained only with sequences) we see that they correlate closely to the actual contact map of the structure, meaning they are *unsupervised* learners of protein structure (Rao et al. 2020, Vig et al. 2020). This property of unsupervised learning (often called self-supervised in the machine learning literature Hendrycks et al. (2019)) of protein structure allows these models to exploit the sequence-structure gap in proteins. We propose that future work with pocket attention could help us overcome the known actives/ligand structure gap. This could be verified by visualising the attention weights between the atoms and the protein on the protein surface.

## 5 Conclusion

This project introduced a novel supervised approach for generating candidate molecules *de novo* for a target protein. To the best of our knowledge, this is the first time a comprehensive description of the target pocket has been used for molecular design in a VAE-based machine learning model. We show that adding pocket information does improve our results during training, but this does not transfer well when assessed by other benchmarks, notably, docking. This may be due to the limitations of the docking scores (Chen 2015) but is more likely to be either an error in our implementation or a fundamental limitation of our approach.

This throws into doubt whether the current paradigm in VAE based approaches for molecule generation will be sufficient to overcome the significant challenges in ligand design, let alone, drug design and that newer and more innovative methods might be needed. We propose Reinforcement Learning (Kaelbling et al. 1996, Skwark et al. 2020) and improved pocket attention mechanisms for unsupervised learning of protein structure and interactions (Rao et al. 2020) as possible directions of future work.

The addition of libraries for easy use of Geometric Deep Learning methods on small molecules (e.g. DeepChem (Ramsundar 2018), GDL-LifeSci (Li et al. 2021) and TorchDrug (`torchdrug.ai`)) and protein structures (e.g. Graphein (Jamash et al. 2020)) will hopefully accelerate work in the field of machine learning and drug design.

## 6 Acknowledgements

The author work like thank Michael Bronstein and Bruno Correia for their supervision in this project and generous access to resources, Freyr Sverrisson and Helena Andres for useful discussions and Arian Jamash for his guidance throughout the project. Arian also implemented the molecular graph featurisation functions.

## References

- Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K. et al. (2017), 'The rosetta all-atom energy function for macromolecular modeling and design', *Journal of chemical theory and computation* **13**(6), 3031–3048.
- Atance, S. R., Diez, J. V., Engkvist, O., Olsson, S. & Mercado, R. (2021), 'De novo drug design using reinforcement learning with graph-based deep generative models'.
- Atz, K., Grisoni, F. & Schneider, G. (2021), 'Geometric deep learning on molecular representations', *arXiv preprint arXiv:2107.12375* .

- Bailey, R. J. & Hay, D. L. (2006), 'Pharmacology of the human cgrp1 receptor in cos 7 cells', *Peptides* **27**(6), 1367–1375.
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R. et al. (2018), 'Relational inductive biases, deep learning, and graph networks', *arXiv preprint arXiv:1806.01261*.
- Beamer, S., Asanovic, K. & Patterson, D. (2012), Direction-optimizing breadth-first search, in 'SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis', IEEE, pp. 1–10.
- Bemis, G. W. & Murcko, M. A. (1996), 'The properties of known drugs. 1. molecular frameworks', *Journal of medicinal chemistry* **39**(15), 2887–2893.
- Bertrand, N., Grenier, P., Mahmoudi, M., Lima, E. M., Appel, E. A., Dormont, F., Lim, J.-M., Karnik, R., Langer, R. & Farokhzad, O. C. (2017), 'Mechanistic understanding of in vivo protein corona formation on polymeric nanoparticles and impact on pharmacokinetics', *Nature communications* **8**(1), 1–8.
- Besnard, J., Ruda, G. F., Setola, V., Abecassis, K., Rodriguez, R. M., Huang, X.-P., Norval, S., Sassano, M. F., Shin, A. I., Webster, L. A., Simeons, F. R. C., Stojanovski, L., Prat, A., Seidah, N. G., Constam, D. B., Bickerton, G. R., Read, K. D., Wetsel, W. C., Gilbert, I. H., Roth, B. L. & Hopkins, A. L. (2012), 'Automated design of ligands to polypharmacological profiles', *Nature* **492**(7428), 215–220. Bandiera\_abtest: a Cg\_type: Nature Research Journals Number: 7428 Primary\_atype: Research Publisher: Nature Publishing Group Subject\_term: Drug discovery;Drug discovery and development;G protein-coupled receptors;Lead optimization;Pharmacology Subject\_term\_id: drug-discovery;drug-discovery-and-development;g-protein-coupled-receptors;lead-optimization;pharmacology.
- Blaschke, T., Arús-Pous, J., Chen, H., Margreitter, C., Tyrchan, C., Engkvist, O., Papadopoulos, K. & Patronov, A. (2020), 'Reinvent 2.0: an ai tool for de novo drug design', *Journal of Chemical Information and Modeling* **60**(12), 5918–5922.
- Bodnar, C., Frasca, F., Wang, Y. G., Otter, N., Montufar, G., Liò, P. & Bronstein, M. M. (2021), Weisfeiler and lehman go topological: Message passing simplicial networks, in 'ICLR 2021 Workshop on Geometrical and Topological Representation Learning'.
- Bronstein, M. M., Bruna, J., Cohen, T. & Veličković, P. (2021), 'Geometric deep learning: Grids, groups, graphs, geodesics, and gauges', *arXiv preprint arXiv:2104.13478*.
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. & Vandergheynst, P. (2017), 'Geometric deep learning: going beyond euclidean data', *IEEE Signal Processing Magazine* **34**(4), 18–42.
- Butina, D. (1999), 'Unsupervised data base clustering based on daylight's fingerprint and tanimoto similarity: A fast and automated way to cluster small and large data sets', *Journal of Chemical Information and Computer Sciences* **39**(4), 747–750.
- Chen, Y.-C. (2015), 'Beware of docking!', *Trends in pharmacological sciences* **36**(2), 78–95.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H. & Bengio, Y. (2014), Learning phrase representations using rnn encoder-decoder for statistical machine translation, in 'Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)'.
- Chowdhury, R., Bouatta, N., Biswas, S., Rochereau, C., Church, G. M., Sorger, P. K. & AlQuraishi, M. N. (2021), 'Single-sequence protein structure prediction using language models from deep learning', *bioRxiv*.
- Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014), Empirical evaluation of gated recurrent neural networks on sequence modeling, in 'NIPS 2014 Workshop on Deep Learning, December 2014'.
- Cieplinski, T., Danel, T., Podlewska, S. & Jastrzebski, S. (2020), 'We should at least be able to design molecules that dock well', *arXiv preprint arXiv:2006.16955*.
- Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. et al. (2009), 'Biopython: freely available python tools for computational molecular biology and bioinformatics', *Bioinformatics* **25**(11), 1422–1423.

- Corso, G., Cavalleri, L., Beaini, D., Liò, P. & Veličković, P. (2020), ‘Principal neighbourhood aggregation for graph nets’, *Advances in Neural Information Processing Systems* **33**.
- de Boer, S. F. & Koolhaas, J. M. (2005), ‘5-HT1a and 5-HT1b receptor agonists and aggression: a pharmacological challenge of the serotonin deficiency hypothesis’, *European journal of pharmacology* **526**(1-3), 125–139.
- DeLano, W. L. et al. (2002), ‘Pymol: An open-source molecular graphics tool’, *CCP4 Newsletter on protein crystallography* **40**(1), 82–92.
- Engkvist, O., Arús-Pous, J., Bjerrum, E. J. & Chen, H. (2020), ‘Molecular de novo design through deep generative models’, *Artificial Intelligence in Drug Discovery* **75**, 272.
- Erhan, D., Courville, A., Bengio, Y. & Vincent, P. (2010), Why does unsupervised pre-training help deep learning?, in ‘Proceedings of the thirteenth international conference on artificial intelligence and statistics’, JMLR Workshop and Conference Proceedings, pp. 201–208.
- Fey, M. & Lenssen, J. E. (2019), Fast graph representation learning with PyTorch Geometric, in ‘ICLR Workshop on Representation Learning on Graphs and Manifolds’.
- Fuchs, F., Worrall, D., Fischer, V. & Welling, M. (2020), ‘Se (3)-transformers: 3d roto-translation equivariant attention networks’, *Advances in Neural Information Processing Systems* **33**.
- Gainza, P., Svärdsson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M. & Correia, B. (2020), ‘Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning’, *Nature Methods* **17**(2), 184–192.
- Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B. R., Vickers, R., Roberts, C., Tang, J., Roblin, D., Blundell, T. L., Bronstein, M. M. & Taylor-King, J. P. (2021), ‘Utilising Graph Machine Learning within Drug Discovery and Development’, *arXiv:2012.05716 [cs, q-bio]*. arXiv: 2012.05716.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), ‘Generative adversarial nets’, *Advances in neural information processing systems* **27**.
- Guo, X. & Zhao, L. (2020), ‘A Systematic Survey on Deep Generative Models for Graph Generation’, *arXiv:2007.06686 [cs, stat]*. arXiv: 2007.06686.
- Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P. & Aspuru-Guzik, A. (2018), ‘Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules’, *ACS Central Science* **4**(2), 268–276. Publisher: American Chemical Society.
- Haga, K., Kruse, A. C., Asada, H., Yurugi-Kobayashi, T., Shiroishi, M., Zhang, C., Weis, W. I., Okada, T., Kobilka, B. K., Haga, T. et al. (2012), ‘Structure of the human m2 muscarinic acetylcholine receptor bound to an antagonist’, *Nature* **482**(7386), 547–551.
- Hendrycks, D., Mazeika, M., Kadavath, S. & Song, D. (2019), ‘Using self-supervised learning can improve model robustness and uncertainty’, *Advances in Neural Information Processing Systems* **32**, 15663–15674.
- Hunter, J. D. (2007), ‘Matplotlib: A 2d graphics environment’, *Computing in science & engineering* **9**(03), 90–95.
- Hussain, J. & Rea, C. (2010), ‘Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets’, *Journal of Chemical Information and Modeling* **50**(3), 339–348. Publisher: American Chemical Society.
- Imrie, F., Bradley, A. R., van der Schaar, M. & Deane, C. M. (2020), ‘Deep generative models for 3d linker design’, *Journal of chemical information and modeling* **60**(4), 1983–1995.
- Imrie, F., Hadfield, T. E., Bradley, A. R. & Deane, C. M. (2021), ‘Deep generative design with 3d pharmacophoric constraints’, *bioRxiv*.

- Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J. & Sayle, R. A. (2020), ‘Zinc20—a free ultralarge-scale chemical database for ligand discovery’, *Journal of chemical information and modeling* **60**(12), 6065–6073.
- Jamasb, A. R., Lió, P. & Blundell, T. (2020), ‘Graphein-a python library for geometric deep learning and network analysis on protein structures’, *bioRxiv*.
- Jumpur, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A. et al. (2021), ‘Highly accurate protein structure prediction with alphafold’, *Nature* pp. 1–11.
- Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996), ‘Reinforcement learning: A survey’, *Journal of artificial intelligence research* **4**, 237–285.
- Kingma, D. P. & Ba, J. L. (2015), Adam: A method for stochastic gradient descent, in ‘ICLR: International Conference on Learning Representations’, pp. 1–15.
- Kingma, D. P. & Welling, M. (2013), ‘Auto-encoding variational bayes’, *arXiv preprint arXiv:1312.6114*.
- Kipf, T. N. & Welling, M. (2016), ‘Variational graph auto-encoders’, *arXiv preprint arXiv:1611.07308*.
- Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. (2017), Grammar variational autoencoder, in ‘International Conference on Machine Learning’, PMLR, pp. 1945–1954.
- Landrum, G. (2013), ‘Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling’.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015), ‘Deep learning’, *nature* **521**(7553), 436–444.
- Li, M., Zhou, J., Hu, J., Fan, W., Zhang, Y., Gu, Y. & Karypis, G. (2021), ‘Dgl-lifesci: An open-source toolkit for deep learning on graphs in life science’, *arXiv preprint arXiv:2106.14232*.
- Li, Y., Vinyals, O., Dyer, C., Pascanu, R. & Battaglia, P. (2018), ‘Learning Deep Generative Models of Graphs’, *arXiv:1803.03324 [cs, stat]*. arXiv: 1803.03324.
- Liao, R., Li, Y., Song, Y., Wang, S., Hamilton, W. L., Duvenaud, D., Urtasun, R. & Zemel, R. (2019), Efficient graph generation with graph recurrent attention networks, in ‘Proceedings of the 33rd International Conference on Neural Information Processing Systems’, pp. 4255–4265.
- Liu, Q., Allamanis, M., Brockschmidt, M. & Gaunt, A. L. (2018), Constrained graph variational autoencoders for molecule design, in ‘Proceedings of the 32nd International Conference on Neural Information Processing Systems’, pp. 7806–7815.
- Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y. & Wang, R. (2015), ‘Pdb-wide collection of binding data: current status of the pdbbind database’, *Bioinformatics* **31**(3), 405–412.
- Mikolov, T., Karafiát, M., Burget, L., Černocký, J. & Khudanpur, S. (2010), Recurrent neural network based language model, in ‘Eleventh annual conference of the international speech communication association’.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M. (2013), ‘Playing atari with deep reinforcement learning’, *arXiv preprint arXiv:1312.5602*.
- Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J. & Bronstein, M. M. (2017), Geometric deep learning on graphs and manifolds using mixture model cnns, in ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 5115–5124.
- Muegge, I. & Oloff, S. (2006), ‘Advances in virtual screening’, *Drug discovery today: technologies* **3**(4), 405–411.
- Mysinger, M. M., Carchia, M., Irwin, J. J. & Shoichet, B. K. (2012), ‘Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking’, *Journal of medicinal chemistry* **55**(14), 6582–6594.

- Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. (2017), ‘Molecular de-novo design through deep reinforcement learning’, *Journal of cheminformatics* **9**(1), 1–14.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. (2019), ‘Pytorch: An imperative style, high-performance deep learning library’, *Advances in neural information processing systems* **32**, 8026–8037.
- Polishchuk, P. G., Madzhidov, T. I. & Varnek, A. (2013), ‘Estimation of the size of drug-like chemical space based on GDB-17 data’, *Journal of Computer-Aided Molecular Design* **27**(8), 675–679.
- Prykhodko, O., Johansson, S. V., Kotsias, P.-C., Arús-Pous, J., Bjerrum, E. J., Engkvist, O. & Chen, H. (2019), ‘A de novo molecular generation method using latent vector based generative adversarial network’, *Journal of Cheminformatics* **11**(1), 1–13.
- Ramsundar, B. (2018), Molecular machine learning with DeepChem, PhD thesis, Stanford University.
- Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. & Rives, A. (2020), Transformer protein language models are unsupervised structure learners, in ‘International Conference on Learning Representations’.
- Richards, F. M. (1977), ‘Areas, volumes, packing, and protein structure’, *Annual review of biophysics and bioengineering* **6**(1), 151–176.
- Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I. & Welling, M. (2021), ‘E (n) equivariant normalizing flows for molecule generation in 3d’, *arXiv preprint arXiv:2105.09016*.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. et al. (2017), ‘Mastering the game of go without human knowledge’, *nature* **550**(7676), 354–359.
- Simm, G., Pinsler, R. & Hernández-Lobato, J. M. (2020), Reinforcement learning for molecular design guided by quantum mechanics, in ‘International Conference on Machine Learning’, PMLR, pp. 8959–8969.
- Skwark, M. J., Carranza, N. L., Pierrot, T., Phillips, J., Said, S., Laterre, A., Kerkeni, A., Şahin, U. & Beguir, K. (2020), ‘Designing a prospective covid-19 therapeutic with reinforcement learning’, *Machine Learning for Structural Biology Workshop, NeurIPS 2020*.
- Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U. & Sutton, C. (2017), Veegan: Reducing mode collapse in gans using implicit variational learning, in ‘Proceedings of the 31st International Conference on Neural Information Processing Systems’, pp. 3310–3320.
- Stumpfe, D. & Bajorath, J. (2012), ‘Exploring Activity Cliffs in Medicinal Chemistry’, *Journal of Medicinal Chemistry* **55**(7), 2932–2942. Publisher: American Chemical Society.
- Sun, C., Shrivastava, A., Singh, S. & Gupta, A. (2017), Revisiting unreasonable effectiveness of data in deep learning era, in ‘Proceedings of the IEEE international conference on computer vision’, pp. 843–852.
- Sverrisson, F., Feydy, J., Correia, B. E. & Bronstein, M. M. (2021), Fast end-to-end learning on protein surfaces, in ‘Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition’, pp. 15272–15281.
- Townshend, R. J., Eismann, S., Watkins, A. M., Rangan, R., Karelina, M., Das, R. & Dror, R. O. (2021), ‘Geometric deep learning of rna structure’, *Science* **373**(6558), 1047–1051.
- Trott, O. & Olson, A. J. (2010), ‘Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading’, *Journal of computational chemistry* **31**(2), 455–461.
- Van der Maaten, L. & Hinton, G. (2008), ‘Visualizing data using t-sne.’, *Journal of machine learning research* **9**(11).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), Attention is all you need, in ‘Advances in neural information processing systems’, pp. 5998–6008.

- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018), Graph attention networks, *in* ‘International Conference on Learning Representations’.
- Venkatasubramanian, V., Chan, K. & Caruthers, J. M. (1994), ‘Computer-aided molecular design using genetic algorithms’, *Computers & Chemical Engineering* **18**(9), 833–844.
- Vig, J., Madani, A., Varshney, L. R., Xiong, C., Rajani, N. et al. (2020), Bertology meets biology: Interpreting attention in protein language models, *in* ‘International Conference on Learning Representations’.
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P. et al. (2019), ‘Grandmaster level in starcraft ii using multi-agent reinforcement learning’, *Nature* **575**(7782), 350–354.
- Wang, C., Jiang, Y., Ma, J., Wu, H., Wacker, D., Katritch, V., Han, G. W., Liu, W., Huang, X.-P., Vardy, E. et al. (2013), ‘Structural basis for molecular recognition at serotonin receptors’, *Science* **340**(6132), 610–614.
- Wang, Z., Sun, H., Yao, X., Li, D., Xu, L., Li, Y., Tian, S. & Hou, T. (2016), ‘Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power’, *Physical Chemistry Chemical Physics* **18**(18), 12964–12975.
- Waskom, M. L. (2021), ‘Seaborn: statistical data visualization’, *Journal of Open Source Software* **6**(60), 3021.
- Xia, X., Hu, J., Wang, Y., Zhang, L. & Liu, Z. (2020), ‘Graph-based generative models for de novo drug design’, *Drug Discovery Today: Technologies* .
- Yang, H., Sun, L., Li, W., Liu, G. & Tang, Y. (2018), ‘In silico prediction of chemical toxicity for drug design using machine learning methods and structural alerts’, *Frontiers in chemistry* **6**, 30.
- Yang, J., Shen, C. & Huang, N. (2020), ‘Predicting or pretending: artificial intelligence for protein-ligand interactions lack of sufficiently large and unbiased datasets’, *Frontiers in pharmacology* **11**, 69.
- Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y. & Liu, T.-Y. (2021), ‘Do transformers really perform bad for graph representation?’, *arXiv preprint arXiv:2106.05234* .
- You, J., Liu, B., Ying, R., Pande, V. & Leskovec, J. (2018), Graph convolutional policy network for goal-directed molecular graph generation, *in* ‘Proceedings of the 32nd International Conference on Neural Information Processing Systems’, pp. 6412–6422.
- Zang, C. & Wang, F. (2020), Moflow: an invertible flow model for generating molecular graphs, *in* ‘Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining’, pp. 617–626.

# Appendices

## A Calculating Attention

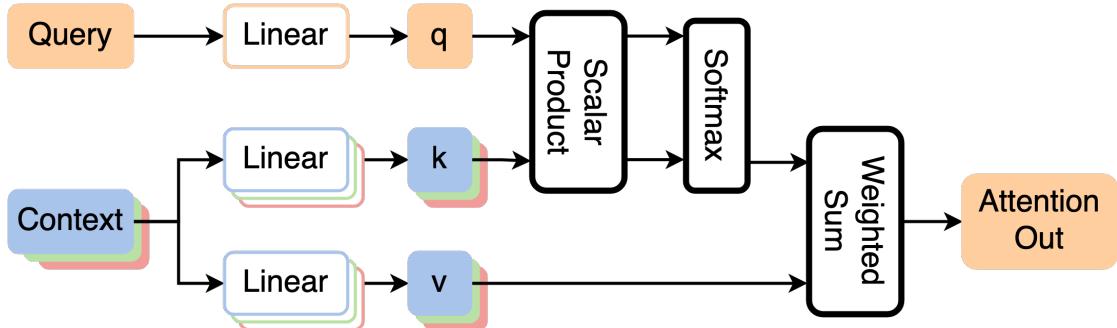


Figure 9: An overview of the attention mechanism.

To calculate attention (Figure 9), we first transform the query and context using one and two linear layers respectively. This produces the transformed query ( $q$ ) and the keys ( $k$ ) and values ( $v$ ) of the context. The scalar product between  $q$  and  $k$  is calculated (i.e. comparing the query features with context keys) and passed to a softmax function, which calculates the attention weights. Finally, a weighted sum is performed between the attention weights and context values  $v$  to return the attention output which is used for some downstream task (Veličković et al. 2018).

## B Training and model implementation

For training, we use the ADAM optimiser (Kingma & Ba 2015) with a learning rate of 0.001 and batch size of 64. All training was performed on either Nvidia RTX 2080 TIs or Tesla V100s. Our approach is implemented in PyTorch (Paszke et al. 2019). Molecule and protein files are parsed with RDKit (Landrum 2013) and BioPython (Cock et al. 2009) respectively and converted into PyTorch Geometric (Fey & Lenssen 2019) Data objects. Drug-likeness properties were calculated with RDKit and figures were produced with Matplotlib (Hunter 2007), Seaborn (Waskom 2021) and PyMol (DeLano et al. 2002).