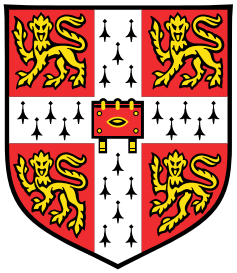
Diffusion Models for Peptide Binding



John Boom

Supervisor: Prof. Pietro Liò
Dr. Pietro Sormanni

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Churchill College August 2023

Dedicated to my family, whose unwavering love and support have made me who I am.

Declaration

I, John D. Boom of Churchill College, being a candidate for the Master of Philosophy in Machine Learning and Machine Intelligence, hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains 14544 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

This dissertation utilizes commonplace Python packages (PyTorch, NumPy, Scikit-Learn, Seaborn, and Pandas) to build the diffusion models and evaluations. Specifically, PyTorch-Geometric underpinned the implementation of the graph neural networks in this work. Our entire implementation of the diffusion process was written from scratch for this dissertation, although the basic implementation of diffusion for protein frames is modelled after Yim et al. [42]. While we referenced their code at times, none of it was directly utilized, and we made major modifications the underlying GNN, changed the translation reverse process, and dramatically modified the final model prediction. The AbAg and SAbDab datasets were curated by Matt Greenig, a fantastic PhD student in the Sormanni Lab, before this dissertation began. Additionally, Matt had already implemented GVP with this library, so that implementation is not original to this work.

John Boom
August 2023

Acknowledgements

Thank you so much, Prof. Liò and Dr. Sormanni. I am so grateful that you agreed to let me join both your labs. I am so grateful for your generous support and time as I conducted this project, and I also really appreciated the way that you let me work independently and waited for me to come for help or with questions. I always felt like I had the right level of support and independence. You both have cultivated a thriving lab culture that is filled with brilliant and collaborative students. It was a joy to be able to join your groups for these past few months, and I hope to have the opportunity to collaborate again in the future.

I would also like to acknowledge Matt Greenig, who helped advise me on this project. Thank you for letting me ask a half-dozen questions a day and never being too busy to help me find the bug or to remind me to 'pip install .' I simply could not have done this dissertation without you, and I look forward to working together again going forward.

Thank you as well to Chaitanya Joshi, Simon Mathis, and Charlie Harris for providing initial guidance as I structured this project. I had so much fun chatting everything GNNs and diffusion over lunch with y'all.

I must also give a heartfelt thanks to my family and my partner Megan. Living overseas has not always been easy, nor were the Cambridge all-nighters... your love and support has kept me going, and I am so lucky to have all of you in my life.

Finally, I would like to thank my high school teachers and my Duke professors. I feel incredibly blessed and privileged to be on this academic journey. I suspect I would not be where I am today if those teachers hadn't nurtured and pushed me to explore my curiosity and attack subjects that I do not understand. Throughout this project and this MPhil, I found myself continually going back to the fundamentals I built in 11th grade calculus. My 17-year-old self would not believe how essential those lessons have been...

Abstract

Rationally designing proteins *in silico* has the potential to unlock treatments for myriad diseases, hasten scientific discovery, and enable a green revolution in manufacturing. While this dream has existed for decades, the last few years have seen unprecedented progress towards this goal as powerful machine learning techniques have been applied to large, high-quality datasets. Within the last year, diffusion models, a particular kind of generative machine learning model, have enabled great advances in designing novel proteins [39].

Here, we build off the recent advancements in diffusion models to introduce a new diffusion model specifically built to generate fragments of proteins that bind to a particular part of a molecule called an epitope [42]. We explore how this model performs in generating the complementarity-determining regions (CDRs) of antibodies, important biomolecules that have many uses in healthcare and scientific technologies. This is an especially difficult task due to the flexibility and diversity of CDR loops, coupled with the relatively small amount of structural data. We demonstrate that our diffusion model is capable of generating novel CDRs that obey biophysical constraints. We also develop a suite of metrics, including a new proof-of-concept using the confidence of an auxiliary masked sequence model. These metrics show that these CDRs appear to be conditioned on the epitope to some degree. Having demonstrated the utility of this model, we noticed a lack of rigorous assessment of which components impact model performance the most. Therefore, we sought to fill this gap by breaking apart our model piece by piece to determine which components most strongly contribute to the model's accuracy. We observed key components in both the construction of the model as well as in the design of the diffusion process that impact model performance. We compare our model on SAbDab, the most common benchmark in the field, and demonstrate that certain common metrics may not be good indicators of structural quality. Finally, we introduce the AbAg dataset of protein fragments and show that this additional data can dramatically boost performance compared to SAbDab alone. In this way, we hope this work can serve as a guide for future scientists developing diffusion models for protein design.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation and Overview

Proteins play a role in almost every function of life. Being able to quickly and accurately design new proteins to carryout different tasks would unlock a vast array of new medical treatments and enable previously unimaginable technologies through synthetic biology. In particular, designing proteins conditioned to bind to a given epitope has important therapeutic. For example, antibodies are widely used as treatments for myriad diseases and especially cancer, due to their ability to recognize a specific molecule and then recruit the body's immune system.

Diffusion models have recently emerged as powerful tools for designing protein structures [39, 4, 29, 41, 37, 27]; however, conditioning these models on a specific task remains a challenge. Moreover, the explosion of interest in protein design with diffusion has outpaced the systematic analysis of the features that tend to make certain models perform better than others.

Therefore, we were motivated by two main questions:

1. Can we build a diffusion model to design the regions of antibodies that are primarily responsible for binding?

2. In the process, can we empirically demonstrate some best practices for protein diffusion models by systematically benchmarking the effect of various design choices?

1.2 Main Contributions

The primary contributions of this dissertation are the following:

- Development of a unique model to generate structures of novel antibody complementarity-determining regions that are conditioned to bind to an epitope.
- Thorough analysis of the how different components such as positional embeddings might improve performance.
- An analysis of whether diffusion models might be able to find the correct location on an epitope to bind by allowing the process to not be centered.
- Experimentation to determine which noise schedules and range of noise work best for both rotations and translations.
- Analysis of a model trained only on SAbDab and a demonstration that pretraining on a separate dataset of protein fragments can significantly boost performance.
- A proof-of-concept that the confidence of masked sequence model trained on protein structures can, in certain settings, be indicative of the likelihood of the structures binding.

1.3 Outline

The dissertation is structured as follows:

Chapter 2 We introduce the importance of protein design and key biochemical concepts which are necessary to understand what components of a protein structure must be generated. We introduce our datasets as well as our key metrics for both the physical plausibility of the generated structures as well as novel metrics to assess whether these structures are conditioned on the epitope. We introduce a proof-of-concept for a new metric based on the confidence of a deep learning model. We conclude by introducing a high-level overview of graph neural networks, specifically SE(3)-Equivariant Message Passing Neural Networks.

Chapter 3 We derive the fundamentals of diffusion models, explaining and connecting the continuous and discrete formulations. We then build on that framework to introduce the necessary changes that need to be made to carry out diffusion on protein structures.

Chapter 4 We first delve into the performance of our base model, demonstrating the high quality of the generated structures. We then systematically study which components of the model contribute to its performance with the hopes of guiding the generative modelling community on best practices for these kinds of diffusion models.

Chapter 5 We continue the analysis of what makes these models perform well; however, it focuses on the noising process of the diffusion. We describe the connection between translations and rotations, we study how the model performs for both translations and rotations as a function of the amount of noise applied to the structure. Then, we empirically assess different noising schedules and different ranges for the noise, providing insight into the optimal configurations for diffusion models.

Chapter 6 We shed light on a common problem in designing models for biology: a lack of large, high-quality, diverse datasets. We demonstrate that training on SAbDab alone, which currently is common practice in the antibody design community, may generated structures with good RMSD but poor structural quality. We also show that pretraining on the AbAg not only recovers structural quality but improves the metrics for binding probability. In this way, we demonstrate that fragment datasets may be highly beneficial to future studies on generative modelling for biology.

Chapter 7 We discuss exciting ways to build upon the findings of this dissertation as well as the limitations of these experiments.

Chapter 2

Background

Chapter Roadmap We first introduce the importance and challenges of protein design before delving into a high-level review of essential topics in biochemistry that are needed to understand of the rest of this dissertation. We then introduce our specific task, predicting antibody complementarity-determining regions (CDR) loops, and introduce the available datasets. Building on this biochemical knowledge, we introduce metrics to evaluate whether the generated CDRs have plausible structures and might bind to the epitope. Finally, we introduce the formulations of graph neural networks that will be used in this project.

2.1 Fundamentals of Protein Design

Proteins are basic building blocks of all life, workhorses that carry out most cellular functions. Their structural diversity and functional versatility enable them to participate in roles ranging from catalyzing biochemical reactions to transmitting signals within and between cells. Rationally designing proteins would give scientists and physicians incredible tools to develop new medicines such as cancer treatments or vaccines. Meanwhile, synthetic biologists and chemical engineers could use rational protein design to manufacture chemicals without carbon emissions or toxic waste. As we unlock protein design, we unlock the potential to reshape, enhance, and repurpose the fundamental building blocks of life itself.

Despite fifty years of effort, *de novo* protein design has remained difficult because proteins are incredibly complex biomolecules whose functions and molecular interactions are not yet fully understood. Therefore, designing proteins with a given function is both important but also incredibly difficult. In 2020, AlphaFold2 (AF2) marked a tremendous advance in biochemistry by providing a pragmatic solution to the first component of the protein folding problem, a 50-year challenge of predicting a protein's structure given only its sequence [23]. AF2 marked a shift away from physics-based methods towards machine learning-based

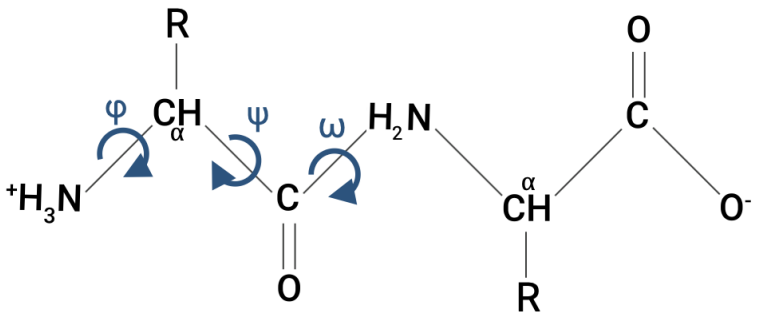


Fig. 2.1 Two amino acids bonded together, with the α-carbons and dihedral angles (ϕ, ψ, ω) labelled.

approaches for solving challenges in biochemistry. In this dissertation, we will describe the use of score-based generative models, a machine learning technique, to design protein fragments conditioned to bind to a particular epitope.

2.1.1 Protein Structure

Understanding the basic elements of protein structure is crucial to building a machine learning model to design novel proteins. Proteins are chains of amino acids (interchangeably called residues). Amino acids contain an amide backbone, and then a side-chain protruding from the C_α (Fig. 2.1). There are twenty different natural amino acids in humans, each with a different sidechain bearing different physiochemical properties.

Therefore, for a protein sequence of length N, there are N^20 possible unique sequences. Proteins have a primary structure: its sequence, secondary structure based on small structural motif, and then tertiary and quaternary structures as combinations of the smaller structural elements. The smallest protein has a length of 11, which could have more than 204 quadrillion sequences [17], but most proteins are between 50 and 500 amino acids [3]. While the potential space of protein sequences is unfathomably large, relatively few sequences will form a stable structure whereby the linear sequence folds into characteristic shapes, such as α-helices, β-sheets, and loops. The hydrophobic effect drives much of protein folding into α-helices and β-sheets, so their sequence composition is constrained [10]. In contrast, loops typically interact with the solvent and are more flexible; therefore, there are fewer constraints on the sequence identity of loops. Protein folding is driven by a few fundamental forces that act on the amino acids in aqueous solutions and include the electrostatic interactions, the formation of hydrogen bonds, Van der Waals interactions and the hydrophobic effect. These same forces governing folding also govern protein-protein interactions.

From the perspective of protein design with machine learning, there are elements of protein structure that any generative model must learn. Amide bonds are quite strong, and the distances between α-carbons lies in a very tight range of roughly 3.7-3.85 Å. Additionally, as shown in Figure 2.1, there are degrees of freedom in the dihedral angles; however, only certain combinations of the backbone angles will result in structures whose sidechains do not physically clash.

2.1.2 Protein-Protein Interactions

Protein-protein interactions (PPIs) are governed by various underlying forces and factors that contribute to the specificity, affinity, and stability of these interactions. Understanding these forces is crucial to evaluating the proposed molecular binders the molecular mechanisms behind PPIs.

Hydrophobic Interactions Hydrophobic amino acid residues tend to cluster together in the interior of protein complexes, away from the surrounding aqueous environment [10].

Electrostatics Like in classic physics, positively and negatively charged, as well as polar, residues, drive many interactions. For biomolecules, hydrogen bonding is a common and important electrostatic interaction.

Van der Waals Interactions Van der Waals forces are weak attractive forces that arise due to fluctuations in electron distribution. These forces contribute to the close packing of amino acids at the protein-protein interface and aid in complex formation [32].

π-π Stacking Aromatic complexes can overlap such that their π molecular orbitals are shared between the two structures, decreasing electron density. While less common at the surface of proteins, it has been documented for antibody binding , and it is highly favorable [6].

Shape Complementarity Protein fragments with complementary shapes and surface features can fit together like puzzle pieces. While proteins can change conformations or a molecule can be induced to fit into a groove, modifying the structure applies an entropic penalty which makes binding less likely.

Overall, these forces contribute to the change in enthalpy (ΔH) of a reaction, which is negative for binding. However, binding also constrains the ability of these molecules to move or adopt different contributions, often causing the change in entropy (ΔS) to be negative. There is a notable exception if the interaction is driven by hydrophobicity, as in this case the binding frees the water molecules around the complex, actually increasing the entropy of the whole system. Therefore, binding reactions must strike a complex balance between forming a highly stable complex (causing a large decrease in enthalpy), and preserving the entropy of the original molecules. For a reaction to proceed, the overall change in Gibb's free energy, $\Delta G = \Delta H - T\Delta S$, must be negative. Therefore, the best binders will have complementary shapes (so that ΔS upon of binding is small), and chemical features in the correct positions such that the complex has many favorable interactions, decreasing ΔH.

2.2 Designing Antibodies

Antibodies play an incredibly important role in the body's immune system, as B-cells can use V/D/J recombination and somatic maturation to make specific antibodies against many different potential antigens [33]. Antibodies have been repurposed by scientists to form cancer therapeutics and other treatments [13]; the global market for therapeutic monoclonal antibodies in 2018 was estimated at $115 billion [28]. Notably, antibodies, or antibody fragments like nanobodies, can usually be safely administered in the body to bind to a target molecule [1].

While novel antibodies for a target of interest can be obtained by immunization or by laboratory-library construction and screening, doing so is time-consuming, expensive, and has no guarantees of success [35]. Rationally designing antibodies *in silico*, therefore, is an important task that will hasten the development of new and possibly better therapies.

Antibodies consist of four protein chains—two heavy chains and two light chains—arranged in a Y-shaped structure, with variable regions at the tips responsible for antigen recognition. Inside each variable region, the binding is mainly dictated by the complementarity determining regions (CDRs), flexible loops structures typically ranging between 6-20 amino acids in length.

Due to its medical significance, this dissertation will focus on improving the binding of antibodies by designing CDRs conditioned on an particular epitope.

2.2.1 Datasets

SAbDab

The Structural Antibody Database (SAbDab) contains experimentally-determined 3D structures of real antibodies, often in complex with a given epitope [16]. All structures from SAbDab were included, and then the CDRs of each antibody and the epitope were extracted. The CDR-epitope complexes were first filtered for structural quality and resolution, and, since many CDRs are highly similar in sequence, we removed CDRs with >90% sequence similarity to other CDRs. After controlling for quality and diversity, 6804 complexes of CDR and epitope remained. Previous work applying machine learning to generate CDRs has focused on SAbDab despite its small size [29, 41]. Notably, to the best of our knowledge, those studies also did not heavily filter SAbDab for quality or unique sequences.

AbAg

While SAbDab is a high-quality dataset of real CDRs, it is very small. The AbAg dataset, introduced by Aguilar Rangel et al. [2], is a much larger dataset of interacting protein fragments that are similar to antibody CDRs. The epitope-like-regions with which the CDR-like-fragments interact are also extracted. Each entry in SAbDab thus contains a CDR-like fragment that is similar to a real CDR, and a corresponding region that the CDR-like fragment interacts with [2]. This dataset was used to rationally design CDRs using a combinatorial approach, obtaining designed antibodies with some affinity to their target, suggesting that the dataset does match key features of antibody CDRs.

To guarantee that the fragments are formed as part of a PPI, we filtered AbAg to include only structures where the CDR-like fragment and epitope are on separate chains. This resulted in 60,658 structures, approximately an order of magnitude larger than SAbDab.

2.3 Evaluation Metrics

Recent works on generative modelling for protein structure have focused on designing whole proteins [39, 42, 37, 4, 27]. Because they focus on whole proteins, they are able to use AF2 to fold their proposed sequence and compare AF2's prediction to the generated structure.

Designing protein fragments, especially fragments conditioned to bind to an epitope, cannot be measured with this metric, or any individual metric. Therefore, we will use the following list of metrics to evaluate the structural quality of the generated CDR structures.

2.3.1 Structure Quality

Certain features of protein structures are conserved, such as the distance between α-carbons, the distribution of the dihedral angles, and the angles between the carbons and nitrogens in the amide backbone. Any atoms on either the CDR or the epitope should not get too close together or they clash. To evaluate these metrics we adopt those used in AlphaFold2 [23]

Table 2.1 Metrics for Evaluating Structure Quality [23]

Metric	Description
Structural Clash	Flags if any atoms are closer together than the sum of their Van der Waals Radii
Bond Angle Violation	Flags if any of the bond angles in the backbone differ from their mean literature values by >12 standard deviations
Bond Length Violation	Flags if N-C_α, C_α-C', or C'-N bonds differ from mean literature value by >12 standard deviations
Clash Between CDR and Epitope	Flags if any α-carbons of the CDR are $\leq$ 3.5 Åto any epitope CDR

2.3.2 Likelihood of Binding

Evaluating whether a CDR is conditioned to a particular epitope is very difficult. While physics-based computational tools like Rosetta can compute a predicted ΔG, these models are known to perform poorly on loops [40]. There are no highly accurate metrics *in silico* tests for to predict whether a loop will bind to a structure. Nonetheless, there are certain characteristics of a good binder. We describe the metrics below and provide a summary in Table 2.2.

Similar to other studies [41, 29], we compute the RMSD of our structures to the structures of real CDRs binding the same epitope. However, we would like to point out that the goal is to generate novel structures, not to just replicate existing ones. There is a lot of degeneracy in how antibodies bind a given epitope. In 2021, Raybould et al. shockingly found over 1131 unique antibodies that bound to COVID-19, often in the same specific epitope [30]. Therefore, the model should give some structures with a low RMSD to the groundtruth, but some predictions are expected and desired to have high RMSD. A better metric may be comparing the RMSD of a CDR compared to a CDR that was generated for a completely different epitope, as that provides a gauge if the model's output is conditioned on the epitope. RMSD is a helpful metric but it should not be the sole focus.

Additionally, the residues of real CDRs are positioned at different distances to the epitope. (Fig. 2.2). Since binding is heavily dependent on distance, the generated CDRs should follow this distribution. To evaluate this numerically, we calculate the Jensen-Shannon Distance (Eqn. 2.1).

$$\sqrt{\frac{D(P||M)+D(Q||M)}{2}} \tag{2.1}$$

where D is the Kullback-Leibler Divergence and M is the mean of the distributions: $M = \frac{P+Q}{2}$.

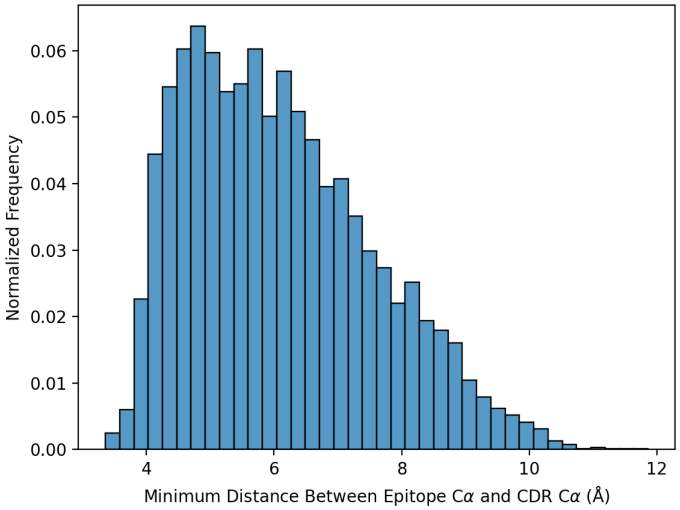


Fig. 2.2 The Minimum Distance Between any C_α in the epitope and the CDR.

Typically there is often a correct orientation in how the CDR should bind an epitope; therefore, we compute the SVD of α-carbons to find the major axis of the CDR, and then we compute the magnitude of the cosine similarity to the groundtruth CDR. Since CDRs are roughly cylindrical, we compute only the similarity to the largest eigenvectors since the 2nd and 3rd do not always align well.

Finally, motivated by the complete lack of models to evaluate whether a structure is plausible or not, we hypothesized that the confidence of an auxiliary, an masked structure model (MSM) trained to identify CDR sequence given their backbone, might be a suitable metric. We propose measuring the confidence with the mean Shannon entropy of the predictions for each residue of the CDR. Specifically we expect the entropy to increase when worse CDR backbones are provided. Since this dissertation is focused on diffusion models, we provide the empirical evaluations of this metric in A. Notably, the metric appears to be sensitive to many corruptions of the CDR, but only for AbAg. Nonetheless, this metric is an

interesting proof-of-concept, and we incorporate it as a metric of our generated structures on the AbAg dataset.

Table 2.2 Metrics for Evaluating Likelihood of CDR Binding

Metric	Description
RMSD	RMSD to the Groundtruth CDR.
Shannon-Jensen Distance	Measures whether the residues are a reasonable distance from the epitope
Magnitude of Cosine Similarity	Checks whether the predicted CDR has the right orientation to the epitope by computing the cosine similarity between the principal components of the predicted and groundtruth CDR
MSM Confidence	Uses the Mean Shannon Entropy to Evaluate, in a black-box manner, whether the generated structures match the original distribution. Only used on AbAg.

2.4 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as a powerful framework for analyzing and learning from data with complex relational structures that do not follow repetitive patterns, such as social networks or molecular structures. In this way, GNNs are uniquely designed to handle information like in protein structures, which themselves can be described as a graph by using the underlying bond networks. By aggregating information from neighboring nodes and edges, GNNs enable the propagation of features and interactions across the entire graph, making them well-suited for tasks like node classification, link prediction, and graph-level analysis. In general, graph neural networks can be grouped into three broad categories: convolutional, attentional, and message-passing. Message-passing neural networks (MPNNs), while not necessarily the most efficient, offer the most flexibility since arbitrary messages can be constructed and passed along edges; therefore, in this dissertation we will focus on MPNNs.

2.4.1 Message Passing Neural Networks

Message-Passing Neural Networks are the most generic form of a GNN layer whereby messages are constructed between each node and its neighbors. The message ($\mathbf{m}_{ij}$ is typically constructed by passing the node features and other information into a neural network.

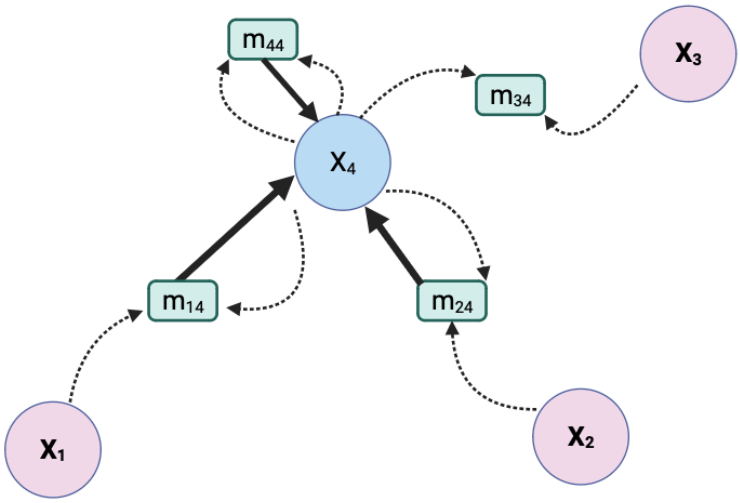


Fig. 2.3 A diagram of a generic MPNN depicting the incoming messages to X_4 from its neighbors. Note that the each node also passes a message to itself.

$$\mathbf{m}_{ij} = \psi(\mathbf{x}_i, \mathbf{x}_j) \tag{2.2}$$

Then, the messages from neighbors ($\mathcal{N}_i$) are aggregated in a permutation-invariant manner. This aggregation is typically concatenated with the original value of the node and then passed into a neural network to return the updated node $\mathbf{h}_i$ (Eqn. 2.3).

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi\left(\mathbf{x}_i, \mathbf{x}_j\right)\right) \tag{2.3}$$

MPNNs are hard to scale because the aggregation step cannot be described as matrix multiplications, meaning that MPNNs are not particularly efficient on modern GPUs and TPUs. Despite this computational cost, MPNNs have been highly successful on a wide very of tasks in computational chemistry [14, 18]; therefore, they are a good choice for handling 3D protein structures.

2.4.2 SE(3)-Equivariance

So far, we have described a generic MPNN for dealing with a graph. However, protein structures are not just a graph, but a geometric graph since each node has a position in 3D space. Biochemically, the distance and angle between residues is important to whether and how they might interact. Therefore, the GNN should take positional information into account.

Additionally, even if it a protein complex is rigidly translated or rotated, it remains the same complex. Therefore, the model must incorporate also be equivariant to the Lie group SE(3), which describes all rigid rotations and translations.

Following the notation of [34], let $T : X \to X$ be a transformation on X and $\phi : X \to Y$. The transformation is invariant to transformation T if $\phi(T(x)) = \phi(x)$ and equivariant if $\phi(T(x)) = T(\phi(x))$. In the case of biomolecules, the transformation T is SE(3).

Invariance and equivariance to rotations and translations has been shown to improve GNN performance [22]. In particular, equivariant GNN layers are more expressive than invariant layers as they can propagate positional information and geometric patterns to future layers. There are a number of SE(3) equivariant GNNs, notably E(n) Equivariant GNNs (EGNNs) [34] and Geometric Vector Perceptron (GVP) [21].

In conclusion, in this section we have introduced the relevant biochemistry, applied that knowledge to develop metrics both for the physical quality of generated structures as well as the likelihood of being a binder, and we introduced SE(3)-Equivariant Graph Neural Networks. In the next section we will derive the fundamentals of diffusion.

Chapter 3

Diffusion Models for Protein Structure

Inspired by physics, diffusion models are a powerful and emerging class of generative machine learning models. Diffusion models have recently attracted a surge of attention as they have been shown to excel at a number of tasks ranging from image generation [36], audio synthesis [25], and even protein design [39]. Diffusion models gradually apply small amounts of noise to the data, eventually completely masking the underlying data. At each step the model is trained to undo the noise and thereby take a step towards the real data distribution X_0. Intuitively, by applying noise to the underlying distribution, the density of the data gradually spreads out at the different timesteps. This means that the model sees lots of intermediate, bad structures in training, but it learns to undo them and progress towards the real data. Additionally, by taking many "steps" repeatedly, the model does not need to make a perfect prediction every time, but rather the model simply needs to trend towards the mean of the distribution. In this sense, the formulation as a diffusion process simplifies the generative task.

There has been an explosion of papers applying diffusion to protein design, and, notably, Watson et al. [39] demonstrated that diffusion generated *de novo* proteins with an unprecedented success rate. Therefore, diffusion models are an excellent choice for designing antibody CDR loops to bind to a given epitope.

Chapter Roadmap We first introduce the empirical utility and underlying intuition of diffusion models. We then derive the fundamental equations underlying diffusion models for both the continuous and discrete formulations; we describe how to map between the two. Having covered the general principles of diffusion, we then describe the specific adaptations for handling protein structure: We derive the relevant equations for the SE(3) manifold and introduce the crucial Lie algebra necessary for incorporating diffusion over rotations. We briefly highlight some of the excellent prior literature in this space to introduce how our

experiments complement existing works. Finally, we introduce the full architecture and training regimen for our model to do diffusion over CDR structure.

3.1 The Discrete Formulation

Diffusion models can be described as first order Markov chains where, in the forward process, noise is gradually applied to the data:

$$\begin{aligned} p(\mathbf{x}_T|\mathbf{x}_0) &= p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_2|\mathbf{x}_1)...p(\mathbf{x}_T|\mathbf{x}_{T-1}) \\ p(\mathbf{x}_T|\mathbf{x}_0) &= \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}) \end{aligned} \tag{3.1}$$

where $p(x_T|x_0)$ is some noising process that converges to a known and tractable stationary distribution.

The model is trained to predict the reverse process $p_\theta(x_{0:T})$, which, as described in Ho et al. [19], can done by optimizing the variational bound on the negative log likelihood.

$$\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right] \tag{3.2}$$

where $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)$ is the real probability of the the sequence $\mathbf{x}_{1:T}$.

Typically, the noising process is defined to converge on the standard Gaussian, and to accomplish this, the noising process must both decrease the mean and apply a variance that eventually converges to be 1. The most common formulation of this is known as a denoising diffusion probabilistic model (DDPM). Ho et al. [19] defines the noising procedure as:

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \tag{3.3}$$

where β_t is a noise schedule. Because each transition is Gaussian, the posterior at any time $\mathbf{x}_t$ can be rewritten in closed form in terms of $\mathbf{x}_0$:

$$p\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{3.4}$$

where $\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{t=1}^{T} \alpha_t$.

There are many methods of reparameterizing the negative log likelihood (Eqn. 3.2). Ho et al. shows that this can be re-parameterized such that the model can simply be trained to predict the isotropic noise applied at a timestep (ε_t). This simple objective allows for easy sampling of $(\mathbf{x}_t, t)$, the reverse process, $\mathbf{x_{t-1}} \sim p_\theta(\mathbf{x}_{t-1}|t)$, which can be defined as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\left(\mathbf{x}_t, t\right)\right) + \sigma_t \mathbf{z} \tag{3.5}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and ε_θ is the output of a model trained to predict ε_t using this simple training objective:

$$L = \mathbb{E}_{x_t, t}\left[\left\|\varepsilon_t - \varepsilon_\theta\left(x_t, t\right)\right\|_2^2\right] \tag{3.6}$$

In summary, we have now established that this discrete formulation of diffusion converges to a known stationary distribution, the isotropic Gaussian. We have shown that we can efficiently sample the noised distribution of any timestep in closed form without traversing the Markov chain. Finally, we have demonstrated that a function approximator trained to predict the isotropic noise applied at each timestep will optimize the variational bound of the negative log likelihood, and that this can be used to define a reverse process.

3.2 Derivation of the Continuous Formulation

While the DDPM approach based on Markov chains is intuitive, this can also be written as a continuous time process because it is an example of an Ornstein-Uhlenbeck (OU) process, which can be rewritten in continuous time or as a Markov chain [11]. The continuous time version can be more naturally related to diffusion and Langevin dynamics, and understanding the continuous time process is crucial to deriving score-based generative models (SGBMs), which will be used to noise the protein residue orientations. The following math is based on the work of Song et al., which also contains a more formal derivation [36].

Let the diffusion process be represented by an Itô SDE:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{3.7}$$

where $\mathbf{f}$ is the drift coefficient that depends on $\mathbf{x}(t)$ and $g(t)$ is a function known as the diffusion coefficient.

The solution to this SDE is given in Anderson [5].

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}} \tag{3.8}$$

where $\bar{\mathbf{w}}$ is the standard Wiener process (Brownian motion).

Therefore, to reverse this diffusion process, a model must learn the Stein score: $\nabla_{\mathbf{x}} \log p(\mathbf{x}_t)$. This can be done using score-matching [20]:

$$L = \mathbb{E}_t \left[\lambda(t) \mathbb{E}_{x_0} \mathbb{E}_{x_t|x_0} \left[\| s_\theta(x_t, t) - \nabla_x \log p(x_t \mid x_0) \|_2^2 \right] \right] \tag{3.9}$$

where $\lambda(t)$ is a weighting function.

Using the score in this manner is highly efficient because the score does not rely on estimating the normalization over all distributions because the gradient of the model parameters is constant with respect to x [12].

The following sections will demonstrate DDPMs are just a special case of SGBMs when the drift and diffusion coefficients are chosen to be certain values. In this sense, SGBMs are more powerful and better represent the underlying Langevin dynamics, motivating their use.

3.2.1 Connecting the Discrete and Continuous Formulations

An OU process is written as the following stochastic differential equation (SDE).

$$dx_t = -\theta x_t dt + \sigma dW_t \tag{3.10}$$

where θ and σ sigma are some function, and W_t represents Brownian Motion, the Wiener process. [1]. Note that noise is not differentiable, and therefore standard calculus cannot be used for SDEs. Itô calculus must be used instead. The trajectory of the x_t can be rewritten in terms of a probability density function, $P(x,t)$, which satisfies the Fokker-Plank equation:

$$\frac{\partial P(x,t)}{\partial t} = \theta \frac{\partial}{\partial x}(xP(x,t)) + \frac{\sigma^2}{2} \frac{\partial^2 P(x,t)}{\partial x^2} \tag{3.11}$$

Assuming that the particle begins at single point, we can apply the initial condition $p(x,t') = \delta(x - x')$ and solve the SDE.

$$P(x,t \mid x',t') = \mathcal{N}\left(x; x' e^{-\int_{t'}^{t} \theta(s)}, \frac{\sigma^2}{2\theta} \left(1 - e^{-2\int_{t'}^{t} \theta(s)} \right) \right) \tag{3.12}$$

Let $\theta = \frac{\beta(s)}{2}$ and $\sigma = \sqrt{\beta(s)}$, where $\beta(s)$ is some noise schedule, and assigning $x' = x_0$ and $t' = 0$, then this can be written as

$$P(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0 e^{-\frac{1}{2}\int_0^t \beta(s)}, (1 - e^{-\int_0^t \beta(s)})\mathbf{I}) \tag{3.13}$$

By analogy to Eqn. 3.4, the discrete and continuous time formulations can be related as the following equations:

[1] This function can be rewritten as a Langevin equation: $\frac{dx_t}{dt} = -\theta x_t + \sigma \eta(t)$, where $\eta(t)$ represents the application of noise

$$\begin{aligned} \sqrt{\bar{\alpha}_t} &= {}^{-\frac{1}{2}\int_0^t \beta(s)} \\ (1-\bar{\alpha}_t) &= 1 - e^{-\int_0^t \beta(s)} \end{aligned} \tag{3.14}$$

3.3 Adapting Diffusion To Protein Structures

Having derived the discrete and continuous formulations of diffusion processes, we will now apply them to designing a diffusion process specifically for protein structures.

3.3.1 SE(3)

Proteins are different from other "data" like images or audio because proteins are real 3D objects and therefore live on the SE(3)-manifold, the manifold of rigid translations and rotations. Therefore, standard SGBMs are not enough; the scores must be defined the manifold. Recent work has extended SGBMs to Riemannian manifolds [15], and in the last six months Yim et al. proposed a mathematically principled framework for the diffusion of protein structures [42]. We chose to replicate Yim et al.'s approach, and the rest of Section 3.3 is based on their excellent work.

To deal with SE(3), Yim et al. carefully define a choice of inner product to identify SE(3) as SO(3) X $\mathbb{R}^3$ from a Riemannian point of view [42]. Crucially, their choice of inner product allows translations ($\mathbb{R}^3$) and rotations (SO(3)) rotations to be handled separately:

Let $\mathbf{T}^t = (\mathbb{R}^t, \mathbf{X}^t)$, where $\mathbf{X}^t$ represents the position and $\mathbb{R}^t$ represents the rotation. Let $\mathbf{B}^t_{\mathscr{M}}$ represent Brownian motion on manifold $\mathscr{M}$ Then,

$$\mathbf{T}^t = [0, -\frac{1}{2}\mathbf{X}^t]dt + \left[\mathrm{d}\mathbf{B}^{(t)}_{\mathrm{SO}(3)}, \mathrm{d}\mathbf{B}^{(t)}_{\mathbb{R}^3}\right] \tag{3.15}$$

Using Yim et al.'s approach, we can now address each $\mathbb{R}^3$ and SO(3) separately.

3.3.2 Translations: $\mathbb{R}^3$

Diffusion on $\mathbb{R}^3$ can be defined in by diffusing towards an isotropic Gaussian according to the standard equations presented in Sections 3.1 and 3.2.

We initially applied the continuous time variation; however, we empirically observed that the scores increased by three orders of magnitude at early timesteps due to numerical instability. This worsened the model's ability to learn the correct loss, and has been observed in other settings [42, 24]. Therefore, noting that the SGBM and DDPM versions are related,

we reverted to the standard DDPM formulation for translations, even as we maintain the continuous time version for SO(3), which we describe below.

Notably, the drift term depends on the magnitude of $\mathbf{x}$; in this way, diffusion is not E(3)-invariant, which can present challenges. To handle this, we ensured that the noised CDR loops were always centered at the origin during training and during generation, as re-centering practically achieves the E(3)-invariance.

3.3.3 Rotations: SO(3)

SO(3) corresponds to all 3D rigid rotations. Leach et al. describe how to sample from the Isotropic Gaussian on SO(3), the IGSO(3) distribution [26]. First, the axis-angle parameterization is used whereby an axis is defined with uniform probability on the sphere. Then, the angle ω is sampled using this probability density:

$$f(\omega) = \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\varepsilon^2} \frac{\sin\left(\left(l+\frac{1}{2}\right)\omega\right)}{\sin(\omega/2)} \tag{3.16}$$

where the variance is ε^2. Note that an additional $\frac{1-\cos\omega}{\pi}$ must be used as a scaling factor when sampling from the distribution. Figure 3.1 shows an example of the IGSO(3) distribution sampled using this approach.

Sampling is down via inverse transform sampling using the CDF. The score can be computed by analytically differentiating f with respect to ω. While this scalar gradient is easy to compute with simple calculus, operating on the SO(3) manifold requires careful use of Lie algebra to ensure that the gradient points in the correct direction. Please see Yim et al. for more details on the relevant Lie algebra[42].

Finding the CDF and analytically computing the score requires approximating that infinite sum. This is computationally intensive to do in a continuous way; therefore, in practice we do not sample random timesteps, but rather pre-compute the scores from a discrete set of angles for which we have already cached the CDF. Inversion can then be done using linear interpolation.

Since the drift term is set to 0, finding the variance of the process is easier and can be found through Itô's lemma, which states that

$$Var[x_t] = \int_0^t g(s)^2 ds, \tag{3.17}$$

where $g(s)$ is the diffusion coefficient.

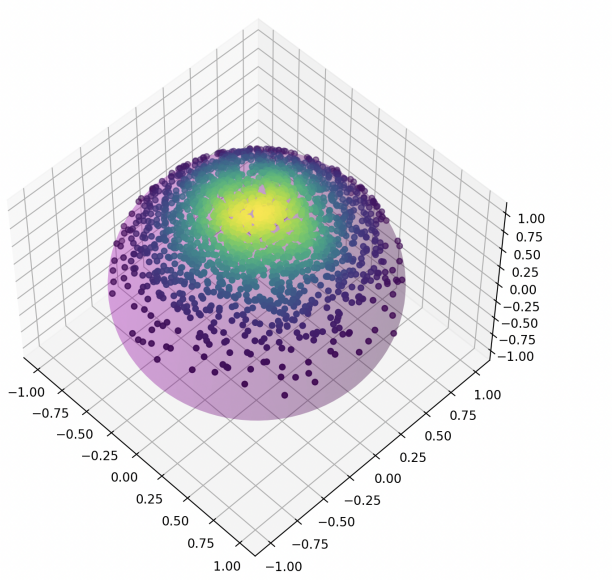


Fig. 3.1 An example of the IGSO(3) distribution plotted on the unit sphere, colored by the density of the distribution.

We chose to follow Yim et al.'s [42] approach and define

$$g(t) = \sqrt{\frac{dVar[x_t]}{dt}} \tag{3.18}$$

and we set the variance schedule $Var[x_t]$ to any function of our choosing.

Finally, SGBMs require a weighting schedule $\lambda(t)$. As in Yim et al. [42], we chose to normalize by the expectation at that timestep so that the model would equally weight the losses at each step regardless of the magnitude of the score. This weighting schedule is very important because the scores increase dramatically at early timesteps.

$$\lambda_t^{\mathrm{r}} = 1/\mathbb{E}\left[\left\|\nabla \log p_{t|0}\left(\mathbf{R}_n^{(t)} \mid \mathbf{R}^{(0)}\right)\right\|_{\mathrm{SO}(3)}^2\right. \tag{3.19}$$

We use Mean Squared Error (MSE) between the predicted score and the groundtruth analytically computed score. To evaluate the expectation in our code, we simply divide by the square of the analytically computed scores.

At this point, we also choose to significantly deviate from Yim et al and Watson et al [42, 39]. In both of those models the output of the model is the final predicted frame $\hat{T}^0$. Then, to get the score, they analytically compute

$$\nabla \log(p(T^t|\hat{T}^0)) \tag{3.20}$$

This means that their network output must live on SE(3), which is why they choose to use the Invariant Point Attention (IPA) network introduced by Jumper et al. for AlphaFold2 [23]. They also have to compute the scores at every step. In contrast, we chose to simply do direct score matching to avoid recomputing the score for every training example. This allow provided greater flexibility in selecting network architectures as there are many more that are SE(3)-equivariant than can predict full frames like IPA.

3.4 Putting the Pieces Together: Combining into a Full Model

We define the diffusion processes as described in the sections above, using the DDPM formulation for translations and the SGBM implementation of SO(3), as we empirically found that switching away from SGBMs for translations avoided loss instabilities.

Since it is computationally expensive to compute the rotations, we arbitrarily chose to include 100 timesteps and cached the IGSO(3) distribution for each the variance at each timestep.

To represent the CDR-epitope we generated a heterogeneous graph to represent the CDR, what we diffuse, and the epitope, which remains constant. Defining the heterogeneous graph in this way maximizes the expressiveness of the GNN because a different message functions is used for each unique combination of node types: CDR-CDR, CDR-epitope, epitope-epitope. To develop the graph, we used a k-Nearest neighbours algorithm with k=6 for each edge type. In other words, each node was connected to up to its 6 closest neighbours of the same type, and 6 neighbours of the opposite type. For our GNN, we decided to use Geometric Vector Perceptron (GVP), as it is a well-characterized SE(3)-equivariant GNN [21].

GVP contains both scalar and vector features for its nodes. To construct the scalar features, we used sinusoidal encodings to encode the timestep, which we concatenated to a sinusoidal embedding of the residue's position in the sequence of the CDR [38]. These encodings were kept quite small, only 5-dimensions each, in order to keep the network lightweight. To construct the vector features, we passed in the orientations of the residue using the axis-angle representation of the vectors. We ablated the sequence information from the CDR as our current model only generates structures, not sequences and structures. However, for the epitope we included sequence information through a one-hot encoding of the amino acid identity as a scalar feature. The epitope vector features were initialized as the

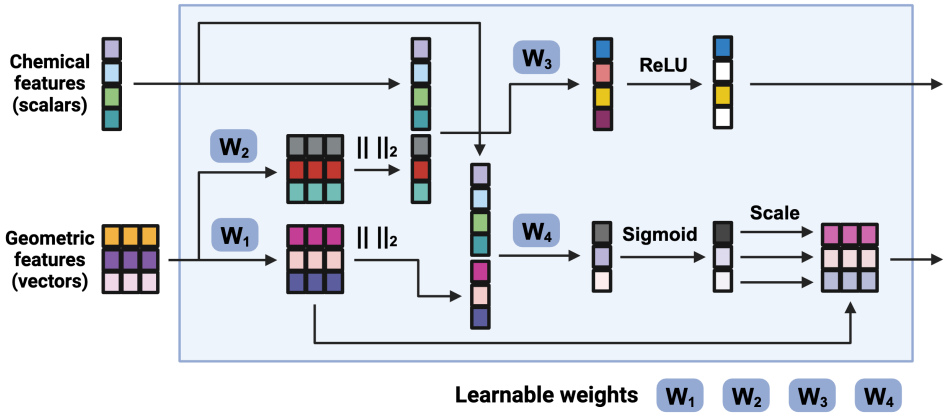


Fig. 3.2 Diagram of how the scalar and vector features are passed through one layer of GVP. Figure credit to my advisor, Matt Greenig.

three bond vectors (C-C_α,N-C_α, C_β-C_α). Figure 3.2 depicts one layer of GVP and how the vector and scalar features are passed through the model.

We also chose to implement self-conditioning as is done in RFDiffusion [39]. Self-conditioning provides the model's previous prediction as a feature to the model, breaking the 1st-order Markov approximation. This attribute is based on the success of "recycling" in AlphaFold2 [23], and we hypothesized that it might improve performance. Similar to Yim et al., we provide self-conditioned features 50% of the time during training [42]. We concatenate the self-conditioned features to the other node features.

We chose to use three message passing layers, and then the final output for each scalar channel is set to 3, to predict the SO(3) score, and the output vector channel returns a single SE(3)-equivariant vector feature, the applied translation noise ε_t.

Chapter 4

Exploring Key Choices in Model Structure and Training Regimen

Chapter Roadmap Although many studies have now been conducted to design proteins, there has been limited exploration of which components are most important. We first describe the results of our model and analyze the quality of the generated structures both in terms of their physical plausibility and in whether they might bind the epitope. We then provide insight into an unanswered debate about the importance of incorporating residue orientations, and we examine how removing the positional encoding affects the generated structures. We explore the impact of self-conditioning both during training and evaluation, and we attempt to allow the model to "dock" the CDR to the correct location.

4.1 Evaluation of Base Model

We trained the mode described in Section 3.4 on AbAg, and we generated 2000 structures on AbAg's test set as well as SAbDab using a noise scale of 0.2. Encouragingly, we find that it generates structures that are highly plausible both on AbAg and on SAbDab (Figure 4.1). When evaluated on both SAbDab and AbAg, the majority of the generated structures are free of any flaws (Table 4.1), although the rate of flawed structures increase 5 times between AbAg and SAbDab. The dihedral angles, as demonstrated by the Ramachandran distribution (Fig. 4.3) precisely match the overall distribution on AbAg. In fact, it appears the model might be slightly overfit as the clusters of angles have lower variance than the groundtruth. Conversely, on SAbDab, it misses the 3rd largest cluster because that pattern (corresponding to α-helices is not present in the AbAg dataset and was not seen during

training. Fascinatingly, it predicts a wider distribution of angles on SAbDab that matches the underlying data, showing remarkably good generalization.

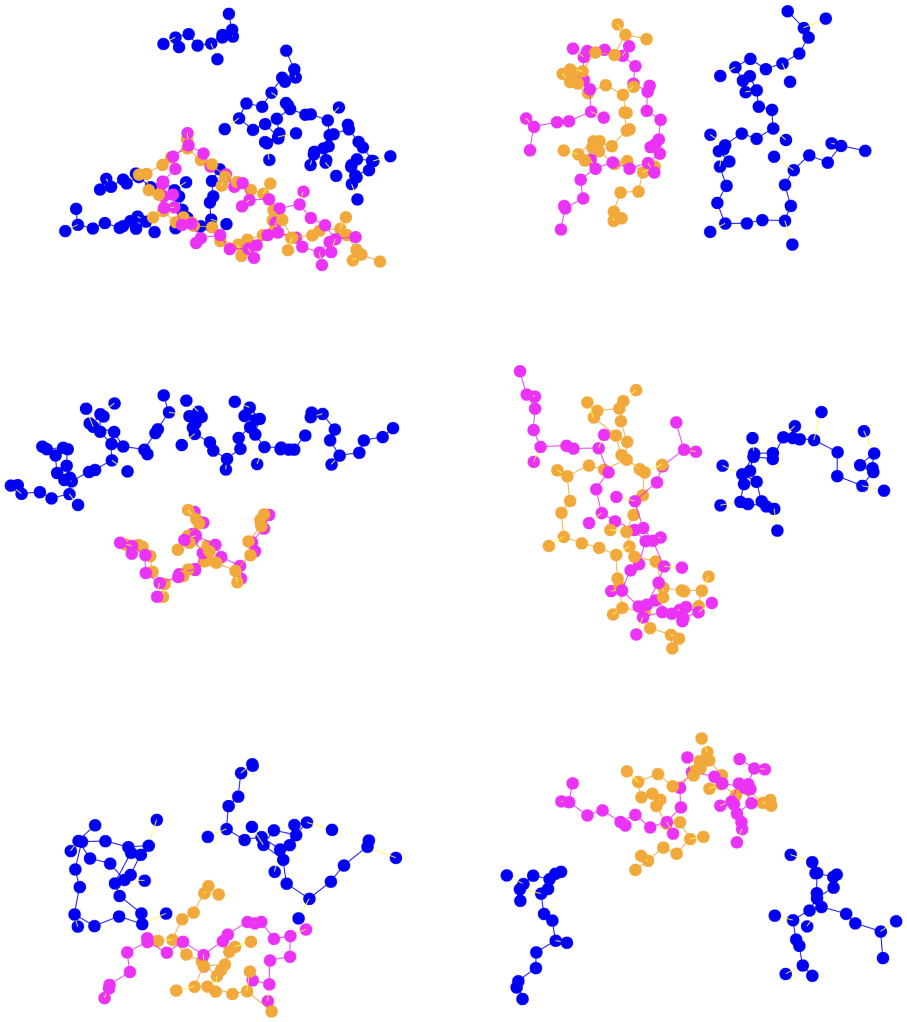


Fig. 4.1 Images of the generated CDR (orange) compared to the groundtruth (magenta) next to their respective epitope (blue). The left column are all structure from the AbAg Dataset. The right column are real CDRs from SAbDab, which the model never saw during training. The model never saw any of these epitopes in training.

We also observe that the model generates structures that are realistic distances from the epitope. The distributions of distances precisely match (Figure 4.2, although a few structures are slightly too close to the epitope.

In conclusion, these results suggest that the model generates remarkably good structures, particularly on Abag, but that the quality of the physical structures generalizes to real CDRs in SAbDab.

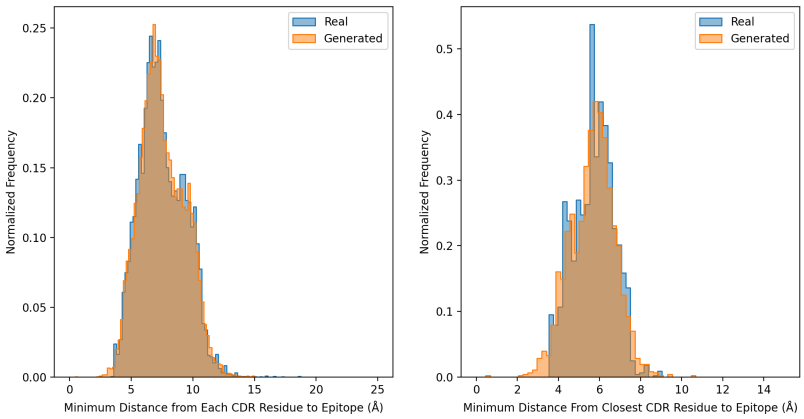


Fig. 4.2 Distances between residues in the generated CDRs to the nearest epitope epitope α-carbon. The distance from all residues to the nearest epitope is on the left, and on the right only the closest residue is plotted, as that is the most important to driving the binding. This plot only shows evaluation on AbAg, but the distribution is nearly identical on SAbDab.

Table 4.1 Comparison of Model's Predictions on SAbDab and AbAg

Metric	Dataset	
	AbAg	SAbDab
Internal Clashes (%) ↓	0.1	2.0
Bond Length Violations (%)	4.0	32.2
Bond Angle Violations (%)	0.9	8.5
Epitope-CDR Clashes (%)	2.6	1.0
Any Structural Flaw (%)	6.1	32.8
RMSD Average (Å)	1.75 ± 0.90	3.46 ± 1.02
All Residue Jensen-Shannon Distance	0.026	0.054
Closest Residue Jensen-Shannon Distance	0.036	0.072
Mean Magnitude of Cosine Similarity	$0.84 \pm$	0.54 ± 0.30

However, when evaluating the likelihood of binding, the generalization of the model on SAbDab is called into question. Most of the structures in AbAg align quite nicely with the epitope as measured by the cosine similarity with the groundtruth CDR. In contrast, many of generated CDRs on SAbDab do not align with the groundtruth CDR orientations. When

analyzing the RMSD, many of the structures have an RMSD < 1.0 Angstroms, the highest possible resolution of the underlying structures. In contrast, the RMSD on SAbDab is no better than a random guess (Fig. 4.4), suggesting that the model may generate good structures that align, to some extent, but that the structures may not be close enough to a real CDR.

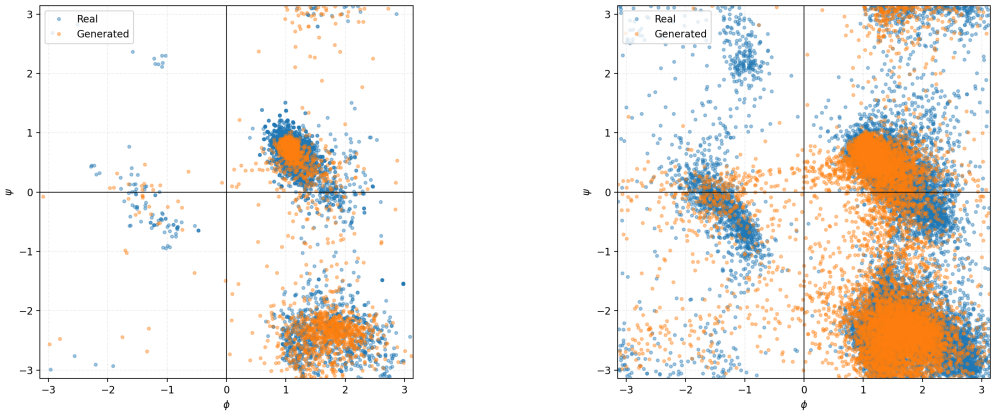


Fig. 4.3 Ramachandran Distributions on Abag (left) and SAbDab (right).

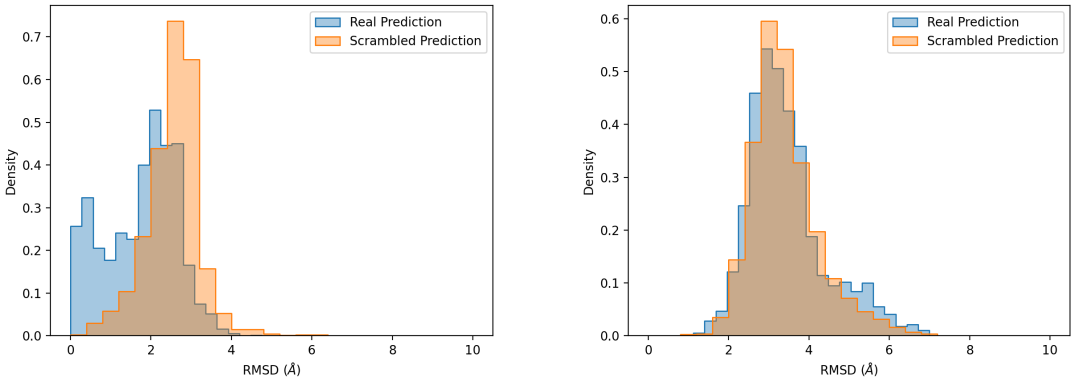


Fig. 4.4 RMSD of the generated CDRs to the groundtruth. AbAg (left) and SAbDab (right). The scrambled prediction is generating by swapping the generated CDR with another generation from a different epitope. Overlap between the real and scrambled predictions on SAbDab suggests that the model struggles to predict a real binder.

Overall, these findings suggest that the base model is remarkably capable of generating CDRs that obey biochemical constraints; however, that the model may not be conditioning on the epitope strongly enough and it may not be generalizing to SAbDab.

Having validated the performance of the base model, we then sought to analyze how different choices in designing the model and diffusion process impact the model's performance.

4.2 Are Orientations Truly Necessary?

In the last eighteen months, there has been an explosion in the number of papers applying diffusion models to design proteins, but no consensus has emerged on how to incorporate rotational information into each residue. FrameDiff [42], RFDiffusion [39], DiffAb [29], and Anand et al. al [4] incorporate specific orientations of each frame. In contrast, ProtDiff [37], presented at ICLR 2023, achieves reasonable results without incorporating frame information at all. Genie also only adds noise in translation, but it gives it's model access to rotational information through constructing frames using the Frenet-Serret frames in the manner of [27]. Surprisingly, no one has evaluated their models with and without rotational information. We hoped to fill that gap.

To rebuild our model without orientations, we simply removed the orientation information entirely in the underlying graph. Specifically, the vector features of the GVP, which previously contained orientation information, was set to a tensor of zeros. We found that the model, even without orientations, distributed the α-carbons into shapes that visually resembled CDR loops and appeared to match the orientation of the epitope. However, the distance between α-carbons no longer matched the groundtruth. Without orientations, the α-carbons tended to be too close together when evaluating on SAbDab and AbAg (Fig. 4.5). Interestingly, ablating orientations not only led to increased variance in the α-carbon distances, it also led the model to miss the mean α-carbon distance by a small amount. While most of the α-carbon distances were feasible, few were as large as the correct average distance. This deviance was so surprising that we trained a second model, which replicated the findings. This finding suggests that orientation information dramatically improves the quality of output structures.

4.3 Analyzing Positional Embeddings

Proteins are formed from linear chains of amino acids; therefore, we initially incorporated this information into the model by concatenating a short sinusoidal embedding of the sequence position (1,2,3,...) to the features of each node in a similar manner to Vaswani et al. [38]. We note that Trippe et al. similarly incorporates positional information [37]; however, the chose to add the difference in sequence position to the edge embedding of their GNN, whereas we only provide it to the node features. Empirically, we observed that our residues always seemed to end up in the correct ordering, suggesting that this approach may be enough for small protein fragments like CDRs. In training, the real node sequence positions were used

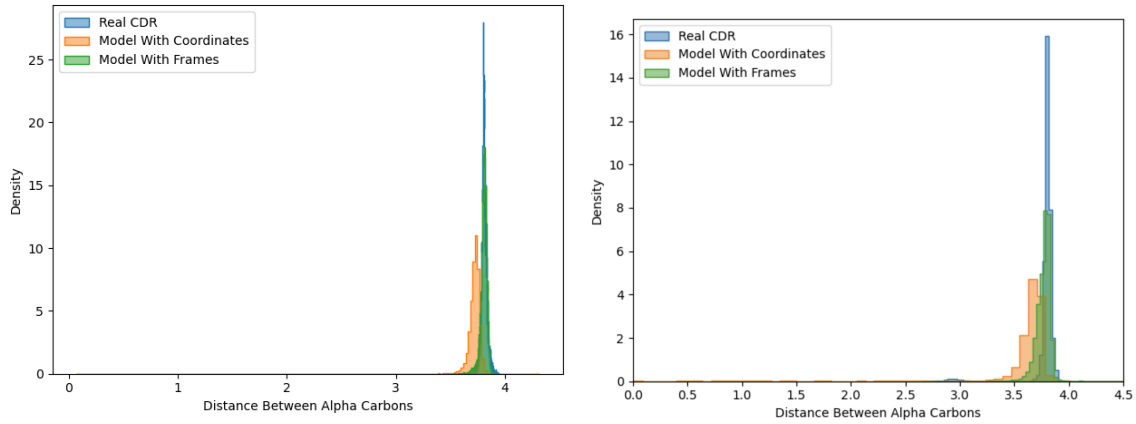


Fig. 4.5 Distance between α-carbons in the generated structures for AbAg (left) and SAbDab (right). The SAbDab plot range has been limited to [0,4.5] to remove one outlier at around 20 Å.

as features. In the reverse generative process, each residue was arbitrarily assigned a position when the noised graph was generated.

To evaluate the importance of this positional embedding, we ablated the positional information. Then, because structures must be organized into a linear chain, we sorted the residues into the correct ordering using a nearest neighbour approach based on their coordinates, and then, to organize the chain from N-terminus to C-terminus, we applied cutoffs to select the orientation where the dihedral angles ω were most similar to the highly conserved values in real structures. Unsurprisingly, we found that the model performed worse; however, we were surprised by the ways in which it decreased. Specifically, the loss for translations increased from 0.23 to 0.26, a relatively small increase. However, the loss for rotations more than doubled from 0.24 to 0.50. Note that the rotational loss is normalized to the size of the groundtruth score, so this increase represents a signficant drop in the model's performance. Interestingly, 84% of the generated structures on AbAg were biochemically plausible (Table 4.2), but the performance dropped sharply on SAbDab, where only 22% of structures were free of major flaws. This analysis reveals that the positional information improves model performance and has a particularly strong impact on the model's ability to generalize.

Table 4.2 Comparison of Structural Flaws when Sequence Position is Ablated

	AbAg		SAbDab	
	Base Model	Position Ablated	Base Model	Position Ablated
Internal Clashes (%)	0.1	6.2	2.0	46.0
Bond Length Violations (%)	4.0	14.2	32.3	77.4
Bond Angle Violations (%)	0.9	6.9	8.5	45.2
Epitope-CDR Clashes (%)	2.6	2.9	1.1	1.4
Any Structural Flaw (%)	6.1	16.2	32.8	77.9

Table 4.3 Impact of Self-Conditioning on Structural Quality and Binding Metrics

	AbAg		SAbDab	
	With	Without	With	Without
Internal Clashes (%)	0.1	0.30	1.95	10.0
Bond Length Violations (%)	4.0	5.25	32.25	42.0
Bond Angle Violations (%)	0.9	1.90	8.50	13.0
Epitope-CDR Clashes (%)	2.6	3.55	1.05	2.0
Any Structural Flaw (%)	6.1	7.75	32.75	43.0
RMSD Average Å	1.75 ± 0.90	1.79 ± 0.90	3.46 ± 1.02	3.48 ± 1.31
All Residue JS Distance	0.026	0.022	0.054	0.056
Closest Residue JS Distance	0.036	0.033	0.073	0.075
Magnitude of Cosine Sim.	0.84 ± 0.21	0.83 ± 0.22	0.54 ± 0.30	0.54 ± 0.30
MSM Prediction Mean	1.61 ± 0.72	1.64 ± 0.70	NA	NA

4.4 How Important is Self-Conditioning?

4.4.1 Ablating Self-Conditioning During Generation

Self-conditioning, as described in Section 3.4, provides the model with access to its prediction at the previous time step. Empirically, Yim et al has shown that it tends to boost performance [42]; therefore, we wanted to validate their findings on our different model and different task.

In Table 4.3, we present the structural quality and binding metrics when self-conditioning is used during the generative process. Similar to Yim et al., we find self-conditioning to increase the overall quality of the structures, although, according to our metrics, there appears to be limited to no improvement in whether such structures might bind the epitope.

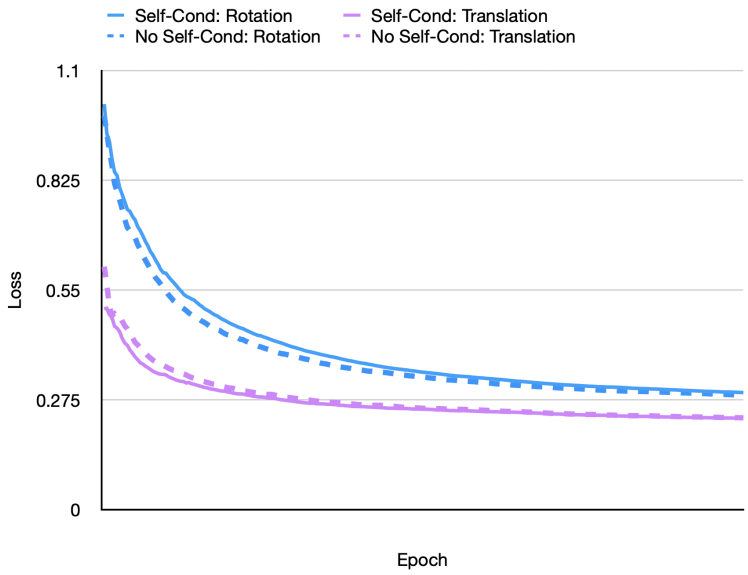


Fig. 4.6 Validation Loss Curves for when training with and without self-conditioning. The self-conditioning rate

4.4.2 Does Self-Conditioning Improve Training?

When training, we provided self-conditioned predictions 50% of the time. However, early in training the model's predictions are effectively noise, so we hypothesized that self-conditioning, while beneficial to the final generative process, might destabilize the training procedure.

We empirically tested this hypothesis by training two models: one without self-conditioning, and the other self-conditioning 50% of the time. The performance of the model without self-conditioning was nearly identical to the results in Table 4.3. Moreover, although we hypothesized that self-conditioning would slow down training, empirically we saw no difference in the validation loss curves (Fig. 4.6). This suggests that more complicated schedules for the self-conditioning rate may not be necessary to achieve optimal performance.

4.5 Docking + Generating CDR

One of the major limitations of our model in its current formulation is that the center of mass of the CDR must be provided so that the complex can be centered such that the CDR center of mass is at the origin. To address this limitation, we wanted to explore whether we could center the CDR-epitope complex at the epitope's center of mass and instead let the model determine where CDR should translate too over time.

This model worked surprisingly well in the sense that the generated CDR structures were all highly plausible, equivalent to when it was centered on the CDR. However, the model often failed to place the CDR at the correct distance from the epitope. Although some predicted structures appeared to be a reasonable distance from the epitope, often the structures would be translated tens or hundreds of angstroms away from the epitope. Sometimes, they were too close. Qualitatively, we also observed that the CDRs tended to drift towards regions of the epitope that were nearby the groundtruth CDR, suggesting that the model may have learned some information about which portions of the epitope would make the best binding site.

This failure mode is not unexpected given that diffusion equation (3.7, which is not E(3)-invariant. The drift term depends on the coordinates of the CDR. As the CDR coordinates increase, the rate of drift correspondingly increases. If small errors in the model's predictions translate the CDR by the wrong amount, these errors can quickly cause the drift term to be too large or small. Interestingly, since the MPNN is SE(3)-equivariant, the model can update the coordinates of the structure to ensure all the nodes move together, thereby maintaining the structural quality of the generated CDRs. In this way, the lack of E(3)-invariance, coupled with an SE(3)-equivariant GNN, neatly explain the behavior seen in Figure 4.7. In the literature [39, 29], the reverse process is made E(3)-invariant by constantly re-centering the structure, as we did in our baseline model.

Overall, these results suggest that centering the CDR is necessary to have a model with a high success rate. However, we would like to point out that the structures themselves are reasonable, and some of them do appear to be positioned at reasonable distances relative to the epitope. Knowing the center of mass of the CDR is a challenge in practice, as the right part of the epitope needed to be selected *and* the center of mass must be placed at the correct distance to the epitope. The distances between epitope and CDR vary a lot in practice, making this problem highly nontrivial. Therefore, "docking" approach, coupled with filters to remove structures in the wrong location, merits further study.

4.6 Takeaways

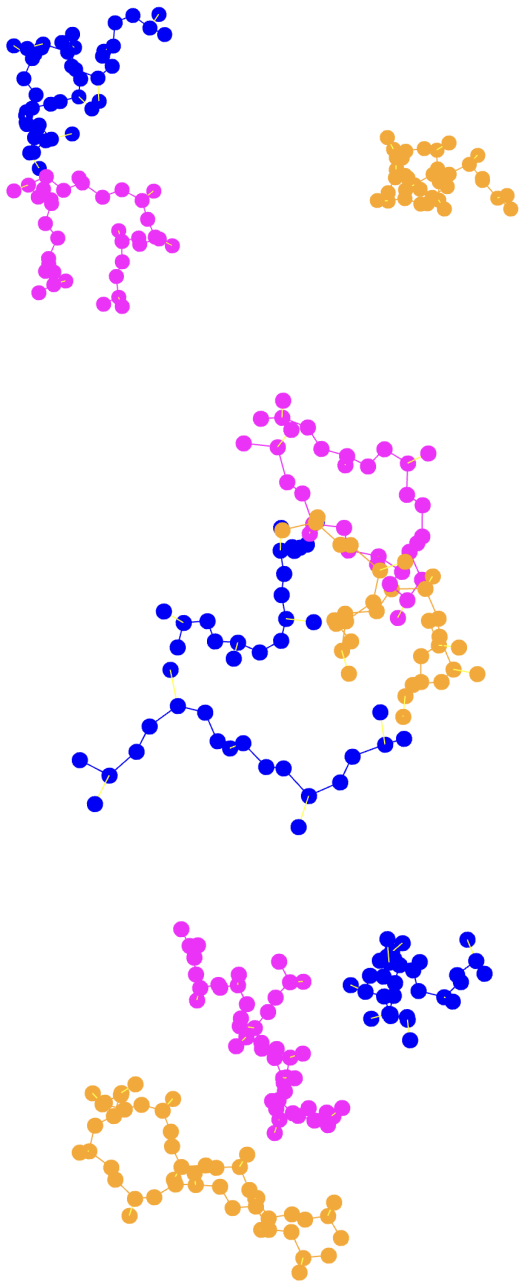


Fig. 4.7 Three generated CDRs (orange) compared to their groundtruth CDRs (magenta) and the epitope (blue). The generated CDRs tend to be at unrealistic distances to the epitope.

Chapter 5

The Impact of Variance Schedules in the Forward and Reverse Diffusion Processes

In the previous section, we analyzed what information improves the model: the orientation of residues, encoding the sequence position of the residues, self-conditioning, and how the timestep is discretized. Now, we turn our attention to the diffusion process itself, specifically the choice of variance schedules for translations and rotations and the amount of noise in the forward and reverse processes.

Chapter Roadmap We first investigate how different variance schedules affect the quality of the generated structures. We then evaluate the quality of structures when the amount of noise in the reverse process is scaled down, and we allude to changes in the structure diversity. Finally we analyze common choices for the lower bound of the noise and demonstrate how this choice may not be optimal.

5.1 Selecting a Variance Schedule

Variance schedules $\beta(s)$ are used to apply different noise to the data at different points in training, and, in the reverse process, steps are taken with respect to the amount of noise that was applied at a given time step.

Variance schedules have been shown to play a large role in the model quality when diffusion models are applied to image. However, to our knowledge no one has explored which variance schedules might be best suited for protein design. Notably, diffusion models for proteins incorporate two different processes over translations and rotations, and the two are biochemically linked and interdependent; the relative position between a residue and its neighbors inform what orientations are possible and vice versa.

When evaluating the model's predictions, we found that the model's performance had a strong dependence on which timestep was sampled (Figure 5.1), and that the pattern differed for rotations and translations. Since certain timesteps appears easier to learn than others, the choice of variance schedule is thus very important and may be different for translations and rotations.

Notably, for translations, the model performs best at the late timesteps. We suspect that this is because the residues have converged on the stationary distribution and are out of order. Therefore, the positional sequence encoding provides a strong signal for how each residue should move. Strikingly, the model does quite poorly at the earliest timesteps. We hypothesize that this is because those timesteps apply a noise threshold that is below the minimum resolution of the data, a hypothesis we will explore in detail in Section 5.3.

The accuracy of the model's predictions for rotations strongly differs from the pattern for translations. Rotations do worst at the late timestep; however, this makes sense because the are out of order and on top of one another. There is less signal dictating how the rotations should change. Similarly to translations, the loss increases again at low timesteps, but it is not so similar. This likely occurs because the average Stein score for rotations increases by an order of magnitude at the lowest timesteps. The model tends to make predictions that are too small at these timesteps, causing the increase in loss.

The complex relationship between the rotation variance schedule, translation schedule, and timestep suggest that optimizing these settings could strongly improve model performance.

5.1.1 Translations

For translations, we investigated four common variance schedules: linear, quadratic, sigmoid, and logarithmic. We held the minimum and maximum β as a constants (1e-4, 20), choosing these values to match Yim et al. [42]. These schedules simply fit a function to start and end at those points. However, for diffusion models we most often care about the cumulative variance $\bar{\alpha}_t$ as this indicates how much of the underlying signal is corrupted at different timesteps.

The cumulative variances of the different schedules are shown in Figure 5.2, where it can be seen that quadratic preserves the most information the longest whereas the logarithmic quickly loses most of the information.

We find that the choice of translation variance schedule strongly impacts the model's performance (Table 5.1). All of the results were generated using a constant rotation variance

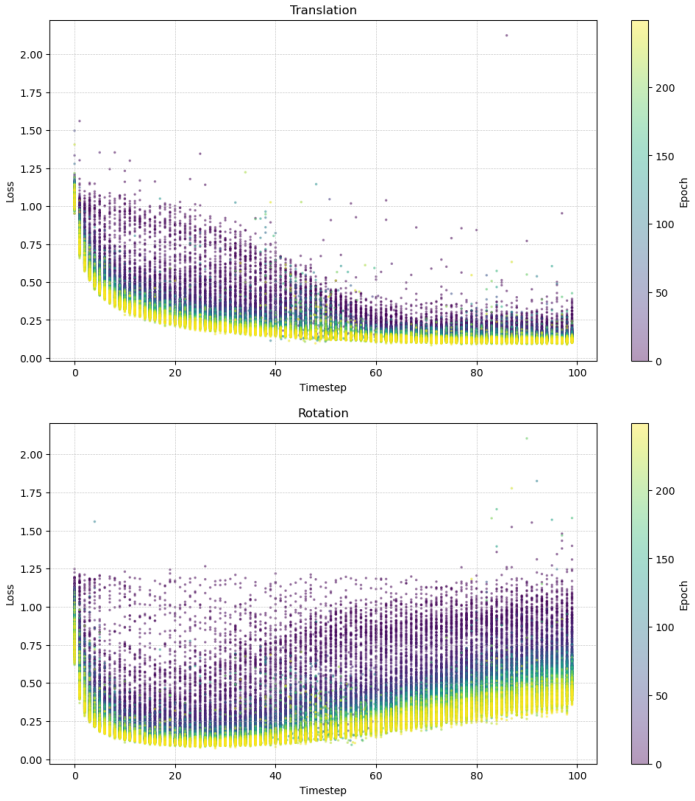


Fig. 5.1 The model's loss for translations (top) and rotations (bottom) on the training set as a function of the sampled timestep and the training time, as measured by the epoch

schedule, the logarithmic schedule for rotations.[1]. Notably, the logarithmic scale performs horribly, with almost all of its structures being flawed. The quadratic noising scale works significantly better than any other others, with more than 80% of its structures being free of flaws. While Table 5.1 shows the results only on SAbDab, similar but less pronounced trends were observed on AbAg. The general trend is that the schedules which apply noise more gradually appear to boost the quality of the generated structures.

5.1.2 Rotations

For rotations, we performed a similar analysis as the translations. The variance schedule is defined slightly differently for rotations because the noising process is not Brownian motion, like for translations. There is no drift term for rotations, simply the noising process. To select a variance schedule, we chose to follow the manner of Yim et al. [42]. Figure 5.3 demonstrates the cumulative variance of these schedules as a function of the timestep. In other words, the variance when sampling $p(x_t|x_0)$, not the step-wise variance of $p(x_t|x_{t-1})$.

[1]The rotation variance schedule, as a reminder, is defined differently than the translation

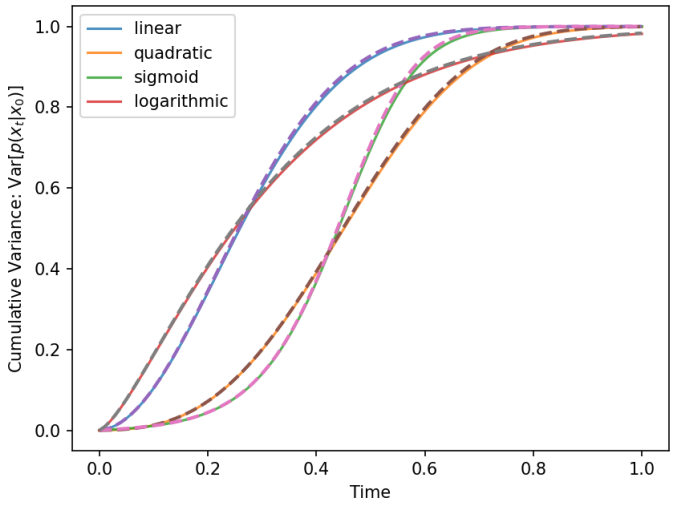


Fig. 5.2 Cumulative variance ($\bar{\alpha}_t$) for the different variance schedules. The continuous time formulation is plotted as the solid line, and the dashed line is the equivalent version using the discrete approximation. The overlap demonstrates that the variance schedules are aligned.

Table 5.1 The Effect of Translation Variance Schedule on the Quality of Generated Structures. SAbDab

	Linear	Quadratic	Logarithmic	Sigmoid
Internal Clashes (%)	2.0	0.6	98.4	9.0
Bond Length Violations (%)	32.3	18.7	99.7	28.2
Bond Angle Violations (%)	8.5	5.3	79.6	13.5
Epitope-CDR Clashes (%)	1.1	1.2	0.8	1.1
Any Structural Flaw (%)	32.8	**19.5**	99.9	28.9

As described in Table 5.2, we evaluated the three different noising schedules with the same range of variances: $\beta_{min} = 0.1$ and $\beta_{max} = 1.5$. Like for translations, the choice of variance schedule has a considerable impact on the model's performance. Specifically, choosing the logarithmic schedule resulted in an $\approx 20\%$ improvement in the number of structures without any flaw.

For translations, there was a general pattern where the more gradual variance schedules seemed to yield better results. We once again notice a trend whereby the logarithmic schedule, the best, has a cumulative variance that increases roughly linearly and relatively gradually. The other schedules apply relatively little noise at first, but then quickly escalate. We theorize that this may have to do with the timing of translations, since most of the translations of the residue occur in the early timesteps because the size drift term depends on the magnitude

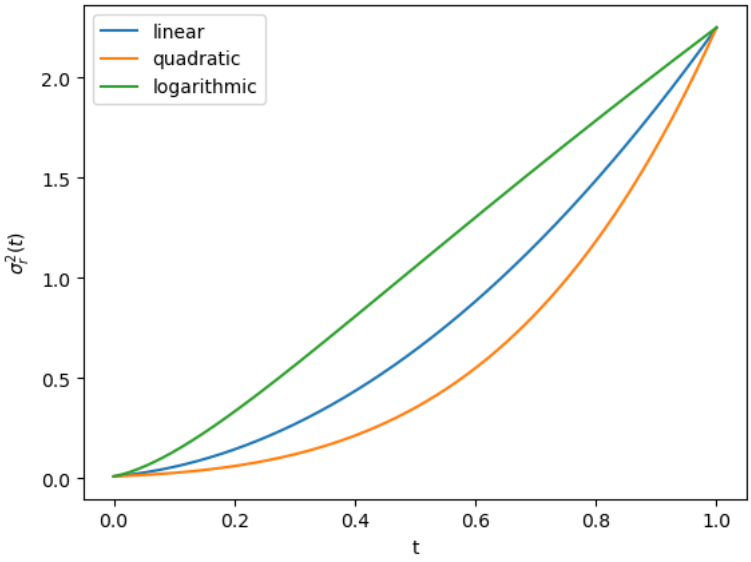


Fig. 5.3 Cumulative variance ($\bar{\alpha}_t$) for the different variance schedules over rotations.

Table 5.2 Impact of the Rotation Variance Schedule on CDR Structure Quality (Evaluated on SAbDab)

	Logarithmic	Linear	Quadratic
Internal Clashes (%)	1.95	5.25	6.4
Bond Length Violations (%)	32.25	38.75	41.2
Bond Angle Violations (%)	8.5	7.4	11.5
Epitope-CDR Clashes (%)	1.05	1.45	1.9
Any Structural Flaw (%)	32.75	39.6	42.6

of the position. Rotations of the frames are predominantly relevant when the structure is partially assembled; therefore, this suggests the model needs to be able to apply sufficiently large perturbations at early timesteps. Since the size of the update is dictated by the variance schedule, the quadratic and linear schedules, which have significantly smaller variance at early timesteps, perform worse. We note that this is only a theory, and there are many hypothetical justifications for its behavior, and there are likely far more complex interactions with the translation noising process that have not been explored in this analysis.

5.2 Scaling Noise in the Reverse Process

The formal reverse process according to Langevin dynamics applies the same amount of noise as the forward process at all timesteps (Eqn. 3.8). However, previous studies have

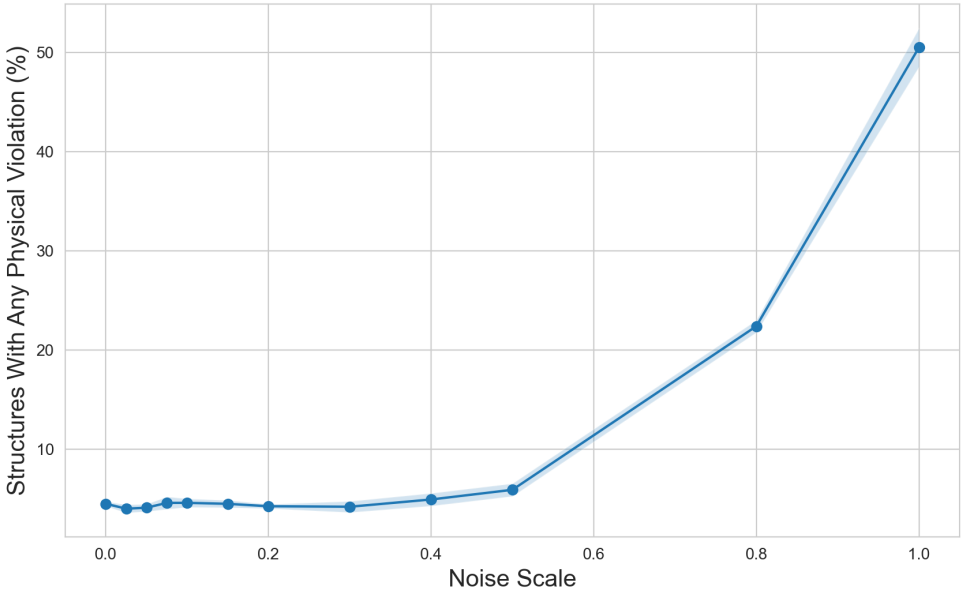


Fig. 5.4 The percentage of generated CDRs that violate any physical constraint in terms of bond length, angle, or steric clash with itself or the epitope. Shading indicates ± 1 standard deviation.

shown that decreasing the amount of noise by a factor $0 \leq \zeta \leq 1$ can improve the final output quality [42].

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x},t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + \zeta g(t) \mathrm{d}\overline{\mathbf{w}} \tag{5.1}$$

We extend the previous analysis by more rigorously analyzing the quality of output structures as a function of ζ. We note that our model and task of protein fragments make this analysis much easier since it takes only 1-200 ms to generate a single structure, far, far faster than comparable models [39, 42].

We find that lowering the noise scale is very important to generating high-quality structures that are free from flaws (Fig. 5.4); however, it appears that the performance plateaus around 0.5. This is encouraging, as lowering the amount of noise in the reverse process undesirably decreases the diversity of the generated structures. This finding suggests that a noise scale between 0.4 and 0.5 will generate high quality structures while hopefully not dramatically decreasing the structural diversity of its predictions. Future experiments are needed to develop reliable metrics to characterize the structural diversity of these small fragments.

5.3 Identifying the Correct Lower Bound of Noise For Translations

Figure 5.1 shows that the model makes no accurate predictions at the lowest timesteps. Because we use the DDPM formulation for translations, the sampled noise comes from the standard Gaussian and does not vary as a function of timestep in the way that the score does. Initially we speculated that these timesteps might be the hardest for the model to learn from; however, the loss for rotations decreased, suggesting that there could be meaningful predictions even at this small timestep. Since the model makes no meaningful predictions at this timestep, we next hypothesized that applied noise is simply random relative to the underlying data, and that therefore there is no signal for the model to learn.

Figure 5.1 was generated using the base model described in 3.4. In it, we applied a lower bound on the noise threshold to be 1e-4 to match the implementation in FrameDiff, a similar model [42]. They chose their variance schedule to range from 1e-4 to 20. [2] This lower bound is incredibly small, especially since the resolution of structures in the PDB typically ranges from 1.5-3 Å[9].

To demonstrate how small this is, Figure 5.5 shows the standard deviation of the noise in Åas a function of the timesteps. At timesteps below the red line, the amount of noise is far smaller than the resolution of the data. This cutoff roughly matches when the model's accuracy becomes to drop, suggesting that the lower bound of the noise should be increased to boost performance. However, it is also possible that this small lower bound might improve the model if it contributes to a variance schedule that is somehow easier to learn.

Based on these findings, we decided to empirically test the model's performance as a function of the lower bound of the noise. Specifically, we tested at every order of magnitude between $1e-4$, that of the original model, and $1e0$, which would never apply noise smaller than the natural resolution of the data.

As shown in Table 5.3, we find that there is no change in the model performance from when the lower bound is increased 100-fold from 1e-4 to 1e-2. However, the number of structural flaws does increase dramatically as the minimum noise threshold of 1e0. Qualitatively, we observe that the model continues to be unable to make meaningful predictions at the smallest timesteps except when $\beta_m in = 1e0$. Therefore, we expect that the optimal threshold likely is somewhere around $1e-1$.

[2] The bound of 20 might seem absurdly large for the variance. However, these bounds are reported using the continuous time notation; they cannot be mistaken for the variances of the Markov process. To convert between them, we use Volterra product integrals. As a quick rule of thumb, to get the variances that would be used with the Markov process, simply divide by the number of steps in the Markov chain. In this case, we use 100, so the variance at the final timestep would be 0.2.

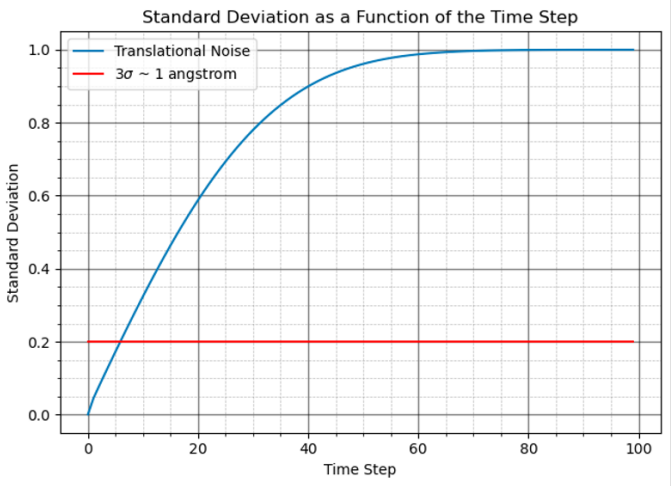


Fig. 5.5 The standard deviation of the noise (Å) for the variance schedule described in Yim et al [42]. The red line denotes when two adjacent when 3σ of noise, applied to both, would cause an expected difference of 1Å. In this sense, at timesteps before that line, the applied noise is so small that it is well below the real resolution of the data.

Table 5.3 Analyzing the quality of the generated CDRs on SAbDab as a function of the β_{min} of the translation variance schedule.

	β_{min}			
	1e0	1e-2	1e-3	1e-4
Internal Clashes (%)	41.4	1.1	4.1	2.0
Bond Length Violations (%)	50.7	33.2	36.2	32.3
Bond Angle Violations (%)	28.8	10.9	10.0	8.5
Epitope-CDR Clashes (%)	1.3	1.2	1.3	1.1
Any Structural Flaw (%)	52.1	33.8	36.6	32.8

Ultimately, these empirical results suggest that there is an optimal minimum of the variance schedule such that the model makes meaningful predictions at all timesteps and that result in a good structure. We expect that this finding would likely generalize for the many different diffusion models for protein design, as the resolution of the underlying protein structures is generally the same across the dataset. Therefore, this analysis is a useful contribution to the field in that it can guide the optimal settings of the variance schedules; however, more detailed analysis is still needed to find the optimal range of variances.

Chapter 6

Impact of Data Availability

In previous sections, we approached the problem of designing antibody CDRs from the perspective of better ML models. However, dataset size and quality remain key concerns in machine learning for science. Therefore, in this section we briefly explore whether this task might benefit from larger datasets.

Chapter Roadmap To date, most models analyzing CDR loops have exclusively trained on SAbDab and similar datasets of real CDRs, to our knowledge. Therefore, we first analyze how training on SAbDab alone compares to the models trained on AbAg for the task of generating CDRs. Then, we investigate whether larger datasets are necessary by analyzing the quality of generated structures as models are trained on fewer and fewer examples. Finally, we pretrain a model on AbAg and then fine-tune on SAbDab, demonstrating that this approach has the highest accuracy of any other method and that the generated CDRs neatly align with groundtruth CDRs.

6.1 Training on SAbDab Alone

SAbDab is one of the largest and most complete datasets of antibodies. However, after filtering to include only antibody-epitope complexes, not just antibody structures alone, and removing any CDRs with >90% sequence similarity, only 6800 structures remained. While a few other works have applied generative models to designing CDRs (Xie et al, Luo et al), these works have been limited to training on SAbDab and a few other very small datasets [41, 29]. An early pre-print of Luo et al. reported that they trained on a similar in-house fragment dataset to AbAg; however, they reported that the dataset did not improve their model's performance compared to training on SAbDab alone [29]. Because of the importance

Table 6.1 Comparing Model Performance When Training Exclusively on SAbDab or AbAg

Evaluation Dataset	AbAg		SAbDab	
Training Dataset	AbAg	SAbDab	AbAg	SAbDab
Internal Clashes	0.05	32.12	1.95	51.52
Bond Length Violations (%)	4.00	45.91	32.25	58.94
Bond Angle Violations (%)	0.90	33.03	8.50	50.91
Epitope-CDR Clashes (%)	2.55	1.21	1.05	0.91
Any Structural Flaw (%)	6.05	46.67	32.75	59.70
RMSD Average (Å)	1.75 ± 0.9	2.54 ± 0.39	3.46 ± 1.02	2.66 ± 0.81
All Residue JS Distance	0.026	0.101	0.054	0.117
Closest Residue JS Distance	0.036	0.211	0.073	0.187
Mag. Cosine Sim.	0.84 ± 0.21	0.634 ± 0.29	0.54 ± 0.30	0.65 ± 0.30
MSM Prediction	1.61 ± 0.72	2.03 ± 0.47	NA	NA

of SAbDab in the antibody design literature, we wanted to explore how our model performed when trained on SAbDab alone.

There are some striking patterns in how the models before. For example, training on AbAg leads to structures that do not have good RMSDs against the known structures, as an RMSD of 3.4-3.6 corresponds to that of a random guess (Fig. 4.4). Shockingly, training on SAbDab, despite its being significantly smaller, leads to RMSD that are better than a random guess on AbAg (2.54 ± 0.39). While this is of course not as good as the model trained on AbAg, it is very interesting that training on SAbDab seems to lead to better RMSDs on AbAg but not vice versa. This may be due to the differences in how the datasets are constructed, as structures in AbAg often are only fragments of CDRs, and they typically contain larger epitopes which constrain the possible solution space. In this way, AbAg is almost certainly an easier task, which is confirmed by the fact that the model trained on SAbDab has a higher RMSD on SAbDab than AbAg (2.66 ± 0.81 vs 2.54 ± 0.39, respectively).

Training on SAbDab meant that the generated structures tended to have the same major axis as the underlying groundtruth, another confirmation that training on SAbDab offers a large performance boost. Similarly, the distribution of distances of the closest residue is slightly better when trained on SAbDab.

While SAbDab appears to generate structures that are closer to the underlying groundtruth, its structures fail at a dramatically higher rate than the model trained on AbAg. 60% of structures generated on SAbDab have a major flaw; the model trained on AbAg is only 33%. In this way, it appears that training on SAbDab incorporates important and unique information about CDR loops. However, likely due to its small size, training on SAbDab

Table 6.2 Comparing Fine-Tuning on SAbDab to Baseline Models. Evaluated on Withheld Test Set of 330 SAbDab structures

	Training Dataset		
	AbAg	SAbDab	Fine-tuned on SAbDab
Internal Clashes (%)	1.95	51.52	**0.15**
Bond Length Violations (%)	32.25	58.94	**14.85**
Bond Angle Violations (%)	8.50	50.91	**2.12**
Epitope-CDR Clashes (%)	1.05	0.91	2.73
Any Structural Flaw (%)	32.75	59.70	**16.52**
RMSD Average (Å)	3.46 ± 1.02	2.66 ± 0.81	$\mathbf{2.45 \pm 1.05}$
All Residue JS Distance	0.054	0.117	**0.030**
Closest Residue JS Distance	0.073	0.187	**0.062**
Mag.of Cosine Sim.	0.54 ± 0.30	0.65 ± 0.30	$\mathbf{0.70 \pm 0.29}$

does not generate as many structures that are physically plausible. This is an important observation as previous papers in this space have mainly focused on RMSD and amino acid sequence recovery rate. However, these results show that structures with a decent RMSD may have major structural flaws.

6.2 Fine-tuning on SAbDab

Training on AbAg alone led to CDRs that generally were very high quality in terms of obeying physical constraints, but the alignment with real CDRs was poor. Training on SAbDab alone created structures that more closely matched the groundtruth, but the generated structures often had major structural flaws. Therefore, we decided to see if fine-tuning a model pre-trained on AbAg might be able to achieve the best of both worlds.

Excitingly, we find that the pre-trained model manages to generate structures that are fairly well-aligned with the epitope and that are mostly free from any major structural flaws (Table 6.2). The fine-tuned model outperforms all other settings, and it incorporates some of the secondary structures that were missing in AbAg, as shown by the Ramachandran distribution in Figure 6.1.

6.3 Impact of Decreasing AbAg Size

Larger datasets almost always improve the power of ML models; however, biological datasets often are extremely expensive and difficult to gather. After filtering, SAbDab contains <7000

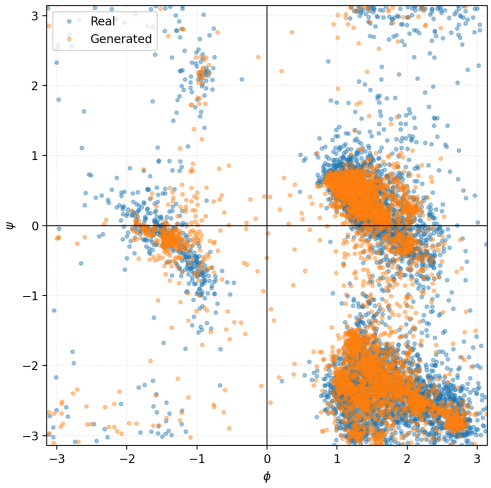


Fig. 6.1 The Ramachandran distribution of generated CDRs compared to the groundtruth improves significantly when using the pre-trained model.

Table 6.3 Analyzing the Quality of 2000 Structures When Trained on Smaller Versions of AbAg

Evaluation Dataset	AbAg			SAbDab		
Training Set Size	100%	50%	25%	10%	50%	25%
Internal Clashes (%)	0.05	0.8	2.65	1.95	17.7	33.15
Bond Length Violations (%)	4	4.6	5.1	32.25	47.85	40.05
Bond Angle Violations (%)	0.9	0.9	1.95	8.5	15.2	26.05
Epitope-CDR Clashes (%)	2.55	1.3	2.75	1.05	1.45	1.45
Any Structural Flaw (%)	6.05	5.5	6.9	32.75	48.2	41.15

complexes, and AbAg <70,000. Compared to the corpus of images for computer vision or speech and text for natural language processing, these datasets are minuscule. Therefore, we were motivated to explore how the model performs when trained on smaller and smaller versions of AbAg.

Surprisingly, we find that the dataset size does not appear to have a particularly large impact on performance (Table 6.3). The model trained on on 100% of AbAg does have the lowest amount of structural flaws on SAbDab by a margin; however, the model trained on only 25% of the dataset outperformed the model trained on 50%. Therefore, the improved performance of the full model may not be fully reproducible, and more experiments are needed to validate these findings. We would also note that we are training with a relatively small model of only 3.42 million parameters. Improving performance often requires simultaneously increasing the number of parameters, amount of data, and training time. It is likely that training a larger model would benefit from the larger dataset.

Chapter 7

Discussion

Overall, the CDRs generated by our models appear to be of very high structural quality, which is incredibly exciting. However, we still appear to have relatively high RMSD against known CDRs. While RMSD is not a perfect metric, our results suggest that the model's predictions could be much more strongly conditioned on the epitope, and future work should examine ways of enforcing the CDRs to not just form reasonable shapes but to match the epitope. Additional *in silico* metrics of binding quality would greatly improve this dissertation and provide a valuable resource to the protein design and machine learning communities.

Another challenge that this dissertation neglects is the fact that epitopes often deform upon binding, adopting a new conformation. This is incredibly important because our model always trains on bound structures in these higher-energy conformations. In real use, however, the model would be provided an epitope alone or perhaps even a predicted structure from AlphaFold2. It would be very exciting to explore whether diffusion models might be able simultaneously generate a CDR and sample different conformations of the epitope, alleviating this problem.

In addition, there are a number of ways that the methodology in this dissertation was limited. While we evaluate a number of the architecture choices for the diffusion model, many papers in the literature utilize different GNNS [37, 42, 29]. A more complete analysis would have additionally tested some of these other major SE(3)-equivariant GNN architectures to determine the optimal configuration. We also set out to explore how the choice of variable schedule for both rotations and translations affects the model performance, and we generated post-hoc explanations for which schedule seemed to be better. This analysis would strongly benefit from future experiments to confirm or refute the explanations. Furthermore, although we point out that the optimal translation and rotation schedules depend on each other, in our experiments we hold rotations constant when we varied the translations and vice versa. A more complete analysis of the schedules is needed.

Additionally, although this dissertation attempts to make general conclusions about designing peptide binders with diffusion models, we exclusively focus on CDR loops. This focus is warranted because Watson et al. has recently demonstrated that RFDiffusion excels at generating binders *de novo*; however, these binders would likely be highly immunogenic and could not be incorporated into antibody CDRs [39]. Designing binding loops is thus a particularly important and difficult prediction task that needs new approaches. Therefore, the focus on CDR loops is justified, but this dissertation cannot make broad claims about designing general peptide binders.

Finally, the largest limitation of this model is that it does not incorporate sequence information. The side chains of amino acids have a wide variety of shapes and sizes, and interactions between these side chains and the epitope are driving the binding reaction. Although Watson et al. [39] found that using diffusing both structure and sequence is unnecessary, that finding might not hold for binding loops, which are very different from the rigid proteins that RFDiffusion generates. Small changes in the position of a residue can have an incredibly strong effect on whether a molecule binds tightly or poorly. Consequently, jointly diffusion over both sequence and structure is a natural and necessary extension of this work.

7.1 Future Directions

Beyond the many additional experiments proposed above, the main next step will be to incorporate sequence information. The initial plan for this dissertation was to examine novel methods of doing sequence diffusion; however, the current experiments easily filled the short 3 months of the dissertation. Nevertheless, I spent over a week reading about different methods of doing sequence diffusion, and I am very excited to explore these approaches.

7.1.1 Discrete Diffusion over Sequences

Rotations and translations exist in a continuous state space; however, amino acids belong to a categorical distribution. Therefore, noise must be applied in a different way, and the model must also learn something different in the reverse process.

Discrete diffusion is a method of noising on the categorical distribution. Specifically, the forward transition probabilities can be expressed as a transition matrix $\mathbf{Q}$, and then the change in the a categorical distribution $\mathbf{s}_t$ can be written as:

$$q(\mathbf{s}_t|\mathbf{s}_{t-1}) = Cat(\mathbf{s}_t; \mathbf{p} = \mathbf{s}_{t-1}\mathbf{Q}_t) \tag{7.1}$$

as expressed in Austin et al. [7]. By definition, $\mathbf{Q_t}$ represents the probability of any one category becoming a different one, and there it is square. So, the transition probability from $\mathbf{s}_0$ to $\mathbf{s}_t$ can be written in closed form as

$$q(s_t \mid s_0) = \text{Cat}\left(s_t; p = s_0 \overline{Q}_t\right), \quad \text{with} \quad \overline{Q}_t = Q_1 Q_2 \ldots Q_t \tag{7.2}$$

Then, the model can be trained to predict the previous distribution of the data by optimizing against the KL-divergence. There are a number of different potential transition matrices; however, to our knowledge, only models using uniform transition matrices [29] and an absorbing state [4] have been published in the literature. Neither of these transition probabilities account for the underlying biology, and it would be fascinating to explore how biologically-inspired transition matrices might improve results.

While discrete diffusion is easy to define mathematically, it is also somewhat nonphysical in the sense that at every step the model samples from the categorical distribution. This means that amino acids might jump to become one with very different physio-chemical properties, and the model may not have enough steps in the reverse process to correct itself.

7.1.2 Dirichlet Diffusion over Sequences

Rather than using the hard jumps of discrete diffusion, Dirichlet diffusion instead applies noise to the probability simplex such that the stationary distribution will be a Dirichlet distribution. Advantageously, this allows the residues to not transition sharply but instead to slowly diffuse towards specific amino acids. The theoretical groundwork for diffusion on the probability simplex has only recently been demonstrated in the last year [8, 31]. To my knowledge, no papers have been published that apply Dirichlet diffusion to generative protein modeling, and believe that this is a very promising direction because the hard transitions of discrete diffusion are nonphysical. If the model samples a transition at a late time step, the entire structure might need to be rearranged to make space to avoid a steric clash. Conversely, in Dirichlet diffusion, if a CDR residue is close to the epitope, the model, in theory, will shift the distribution so that more mass is assigned to the vectors where the residue is relatively small, and this information can be preserved through the rest of the diffusion process.

Chapter 8

Conclusion

We developed a new diffusion model to design protein fragments that will bind to a given epitope. Specifically, we focused on a particularly difficult but important task, designing the flexible CDR loops that mediate antibody affinity and specificity. Excitingly, our model was capable of generating structures that obeyed the fundamental physical principles underlying protein structure, and it often gave new structures that appeared dissimilar to the groundtruth CDR, suggesting that these CDRs are both plausible and reasonable. Despite the explosion of recent works developing diffusion models for protein design, there has been little analysis of which features of the models most strongly affect the quality of the generated structures. We thoroughly benchmarked different design choices and highlighted the importance of using positional sequence encodings to enforce the protein to adopt a linear structure. We showed that giving the model access to rotational information is essential to generating plausible structures. We demonstrate a proof-of-concept for allowing the model to predict where a CDR should bind; however, this approach had middling success due to the underlying lack of $\mathbb{R}^3$-invariance. We experimented with different noising schedules for translation and rotation, finding that the choice of schedule and of the maximum bounds on the variance schedule are highly important. Finally, we study whether the availability of data for this task. We show that training on SAbDab alone, the approach of scientists in this field, empirically gives structures that are physically flawed. These flawed structures also had reasonably good RMSD, which is striking because many recent papers have not evaluated the structural quality of the generated CDRs and instead focused on RMSD.

Ultimately, this work is focused on how machine learning can be used to hasten scientific progress and unleash the potential of synthetic biology to improve healthcare. I hope that these findings will spur further research that eventually realizes the dream of *in silico* rational protein design.

References

[1] Ackaert, C., Smiejkowska, N., Xavier, C., Sterckx, Y. G. J., Denies, S., Stijlemans, B., Elkrim, Y., Devoogdt, N., Caveliers, V., Lahoutte, T., Muyldermans, S., Breckpot, K., and Keyaerts, M. (2021). Immunogenicity Risk Profile of Nanobodies. *Frontiers in Immunology*, 12:632687.

[2] Aguilar Rangel, M., Bedwell, A., Costanzi, E., Taylor, R. J., Russo, R., Bernardes, G. J. L., Ricagno, S., Frydman, J., Vendruscolo, M., and Sormanni, P. (2022). Fragment-based computational design of antibodies targeting structured epitopes. *Science Advances*, 8(45):eabp9540. Publisher: American Association for the Advancement of Science.

[3] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). The Shape and Structure of Proteins. In *Molecular Biology of the Cell. 4th edition*. Garland Science.

[4] Anand, N. and Achim, T. (2022). Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models. arXiv:2205.15019 [cs, q-bio].

[5] Anderson, B. D. O. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

[6] Arzhanik, V., Svistunova, D., Koliasnikov, O., and Egorov, A. M. (2010). Interaction of antibodies with aromatic ligands: the role of pi-stacking. *Journal of Bioinformatics and Computational Biology*, 8(3):471–483.

[7] Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Berg, R. v. d. (2023). Structured Denoising Diffusion Models in Discrete State-Spaces. arXiv:2107.03006 [cs].

[8] Avdeyev, P., Shi, C., Tan, Y., Dudnyk, K., and Zhou, J. (2023). Dirichlet Diffusion Score Model for Biological Sequence Generation.

[9] Bank, R. P. D. (2023). PDB Statistics: PDB Data Distribution by Resolution.

[10] Camilloni, C., Bonetti, D., Morrone, A., Giri, R., Dobson, C. M., Brunori, M., Gianni, S., and Vendruscolo, M. (2016). Towards a structural biology of the hydrophobic effect in protein folding. *Scientific Reports*, 6(1):28285. Number: 1 Publisher: Nature Publishing Group.

[11] Chan, K. C., Karolyi, G. A., Longstaff, F. A., and Sanders, A. B. (1992). An Empirical Comparison of Alternative Models of the Short-Term Interest Rate. *The Journal of Finance*, 47(3):1209–1227. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1992.tb04011.x.

[12] Chang, Z., Koulieris, G. A., and Shum, H. P. H. (2023). On the Design Fundamentals of Diffusion Models: A Survey. arXiv:2306.04542 [cs].

[13] Darvin, P., Toor, S. M., Sasidharan Nair, V., and Elkord, E. (2018). Immune checkpoint inhibitors: recent progress and potential biomarkers. *Experimental & Molecular Medicine*, 50(12):165.

[14] Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., Bera, A. K., King, N. P., and Baker, D. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56. Publisher: American Association for the Advancement of Science.

[15] De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. (2022). Riemannian Score-Based Generative Modelling. arXiv:2202.02763 [cs, math, stat].

[16] Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. (2014). SAbDab: the structural antibody database. *Nucleic Acids Research*, 42(D1):D1140–D1146.

[17] Galindo, M. I., Pueyo, J. I., Fouix, S., Bishop, S. A., and Couso, J. P. (2007). Peptides Encoded by Short ORFs Control Development and Define a New Eukaryotic Gene Family. *PLoS Biology*, 5(5):e106.

[18] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. arXiv:1704.01212 [cs].

[19] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs, stat].

[20] Hyvärinen, A. (2005). Estimation of Non-Normalized Statistical Models by Score Matching.

[21] Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. (2021). Equivariant Graph Neural Networks for 3D Macromolecular Structure. arXiv:2106.03843 [cs, q-bio].

[22] Joshi, C. K., Bodnar, C., Mathis, S. V., Cohen, T., and Liò, P. (2023). On the Expressive Power of Geometric Graph Neural Networks. arXiv:2301.09308 [cs, math, stat].

[23] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589. Number: 7873 Publisher: Nature Publishing Group.

[24] Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the Design Space of Diffusion-Based Generative Models. arXiv:2206.00364 [cs, stat].

[25] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2020). DiffWave: A Versatile Diffusion Model for Audio Synthesis.

[26] Leach, A., Schmon, S. M., Degiacomi, M. T., and Willcocks, C. G. (2022). DENOISING DIFFUSION PROBABILISTIC MODELS ON SO(3) FOR ROTATIONAL ALIGNMENT.

[27] Lin, Y. and AlQuraishi, M. (2023). Generating Novel, Designable, and Diverse Protein Structures by Equivariantly Diffusing Oriented Residue Clouds. arXiv:2301.12485 [cs, q-bio].

[28] Lu, R.-M., Hwang, Y.-C., Liu, I.-J., Lee, C.-C., Tsai, H.-Z., Li, H.-J., and Wu, H.-C. (2020). Development of therapeutic antibodies for the treatment of diseases. *Journal of Biomedical Science*, 27(1):1.

[29] Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. (2022). Antigen-Specific Antibody Design and Optimization with Diffusion-Based Generative Models for Protein Structures.

[30] Raybould, M. I. J., Kovaltsuk, A., Marks, C., and Deane, C. M. (2021). CoV-AbDab: the coronavirus antibody database. *Bioinformatics (Oxford, England)*, 37(5):734–735.

[31] Richemond, P. H., Dieleman, S., and Doucet, A. (2022). Categorical SDEs with Simplex Diffusion. arXiv:2210.14784 [cs].

[32] Roth, C. M., Neal, B. L., and Lenhoff, A. M. (1996). Van der Waals interactions involving proteins. *Biophysical Journal*, 70(2):977–987.

[33] Roth, D. B. (2014). V(D)J Recombination: Mechanism, Errors, and Fidelity. *Microbiology spectrum*, 2(6):10.1128/microbiolspec.MDNA3–0041–2014.

[34] Satorras, V. G., Hoogeboom, E., and Welling, M. (2022). E(n) Equivariant Graph Neural Networks. arXiv:2102.09844 [cs, stat].

[35] Sifniotis, V., Cruz, E., Eroglu, B., and Kayser, V. (2019). Current Advancements in Addressing Key Challenges of Therapeutic Antibody Design, Manufacture, and Formulation. *Antibodies*, 8(2):36.

[36] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. arXiv:2011.13456 [cs, stat].

[37] Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. (2023). Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. arXiv:2206.04119 [cs, q-bio, stat].

[38] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention Is All You Need. arXiv:1706.03762 [cs].

[39] Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., Bortoli, V. D., Mathieu, E., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. (2022). Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. Pages: 2022.12.09.519842 Section: New Results.

[40] Weitzner, B. D., Jeliazkov, J. R., Lyskov, S., Marze, N., Kuroda, D., Frick, R., Adolf-Bryfogle, J., Biswas, N., Dunbrack, R. L., and Gray, J. J. (2017). Modeling and docking antibody structures with Rosetta. *Nature protocols*, 12(2):401–416.

[41] Xie, X., Lee, J. S., Kim, D., Jo, J., Kim, J., and Kim, P. M. (2023). Antibody-SGM: Antigen-Specific Joint Design of Antibody Sequence and Structure using Diffusion Models.

[42] Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. (2023). SE(3) diffusion model with application to protein backbone generation. arXiv:2302.02277 [cs, q-bio, stat].

Appendix A

Evaluating Structure Quality Using Confidence of Auxiliary Model

Generative modelling for protein structures has exploded in recent years; however, while good metrics exist for evaluating whether structures are biochemically plausible, there is a dearth of *in silico* metrics to evaluate whether a protein structure might carry out a specific function. While some metrics exist for the surfaces of whole proteins CITATION: MASSIF, and there are physics-based tools such as Rosetta CITATION: Rosetta, these tools are limited in accuracy, require information about the sequence of the residue, and perform better on larger structures than the loops in our dataset. It is also notoriously difficult to calculate the ΔG of binding for loops as they are much more flexible than alpha helices or beta sheets, which increases the entropic penalty of binding and allows for more degrees of freedom in the shape of the binder. Furthermore, physics-based computational software such as Rosetta are quite expensive; it would be very computationally intense to evaluate the tens of thousands of novel structures generated in this dissertation. Therefore, we were motivated to develop a novel, fast metric to evaluate the likelihood of a CDR binding to a particular epitope.

We hypothesized that the confidence of an auxiliary model, trained on real CDRs, might be a suitable, rapid metric. We theorized that the model would have lower confidence in its predictions if it was given a structure where the CDR was less likely to bind the epitope, perhaps due to being too close or far away to the epitope, having a poor orientation to the epitope, or simply having a shape that is dissimilar to real CDRs. In this way, the out-of-distribution (OOD) data could be detected by the confidence of the model. However, deep learning models typically have no guarantees of performance on OOD examples. In fact, some OOD examples might elicit an extraordinarily confident prediction from the model, despite being completely wrong CITATION: Hendrycks all through here.

To use a deep learning model's confidence, therefore, we would need to empirically demonstrate that it becomes less confident as the quality of the underlying data decreases. To do this, we first trained a masked sequence model (MSM) to predict the protein sequence of the loop (inverse folding). The model was based on the same architecture as LoopDiff, a GVP, but the MLP in the last layer was modified to predict which amino acids were most likely for a given residue. Notably, unlike other inverse folding models like ProteinMPNN CITATION: daupaurus, this model's task is not just to find a sequence that folds into the correct shape, but also to identify the right individual amino acids so that the loop tightly binds the epitope. We aimed to simulate CDRs that are bad binders and test whether the model's confidence would predictably decrease as the CDR structure worsened.

Biochemically, a CDR's binding would strongly depend on its geometric relationship to the epitope. Being even slightly too far away can drastically decrease the strong of protein interactions; conversely, being too close would introduce steric strain on the loop (an entropic penalty) or cause repulsion if truly too close. Therefore, we determined that the model's confidence should predictably increase if any of the following occurred:

1. The CDR is translated away from the epitope
2. The CDR is translated towards the epitope
3. The CDR is translated parallel to the epitope
4. The CDR is rotated parallel to the epitope
5. The CDR is rotated and translated

To measure the model's confidence, we compute the mean Shannon Entropy over all of the residues in the CDR (Eq. A.1).

$$H(X) = \frac{1}{N}\sum_{n}^{N}\sum_{i}^{I} -P(x_{i,n})\log P(x_{i,n}) \tag{A.1}$$

As shown in Figure A.1, the mean entropy of the model's predictions increases linearly when the CDR is rigidly translated away from the center of mass of the epitope, before plateauing at a large amount of entropy. Similarly, when translated parallel to the epitope, the entropy increases before plateauing. Notably, there is a wide variance in the entropy without any corruption, and the variance is roughly constant as the CDR is modified and the entropy increases.This large variability means that this method cannot judge the quality of any one structure; however, it is still applicable to evaluating a large group of predictions, because in that case the mean of the predictions is meaningful.

Notably, the entropy does not monotonically increase when the CDR is moved closer. It first increases, and the rapidly falls when the structures are overlapping. This failure mode would be detected as a clash by other metrics; therefore, it's not a complete failure. Interestingly, this occurs because the MSM predicts that the residues should be small amino acids to avoid clashes. There was a 60% chance that the model predicted glycine, threonine, or alanine. Therefore, although the entropy does not predictably increase as they get closer, in practice this failure mode is both rare, would be captured by other metrics, and actually reflects the model making a prediction that matches biochemical intuition.

Having established that translations cause predictable changes in the entropy, we then hoped to evaluate rotations. The CDR was rotated around its center of mass and the vector between the center of masses of the CDR and the epitope (Fig. XXY). In this way, the CDR is rotated "parallel" to the epitope. Encouragingly, we find that as the CDR is rotated even slightly, the entropy dramatically increases. Since most CDR loops are roughly cylindrical, not spherical, the worst alignment would likely be at 90º angle rotations, which we empirically observe. Moreover, the prediction is symmetrical. In this way, the MSM's mean confidence is clearly dependent on the CDR having the correct orientation relative to the epitope.

Finally we demonstrate that translating and rotating and the same time both independently increase the entropy, suggesting that the MSM metric is robust (Fig. A.2).

So far, all of the results have been on the AbAg dataset with a model trained on AbAg. Unfortunately, we were unable to train any MSM on SAbDab, which is both smaller and more complex, that had any meaningful confidence in its predictions. All of the predictions on SAbDab had relatively high entropy. As shown in Figure A.3, this means that this metric fails on predictions of real CDRs and can only be used to evaluate predictions on AbAg.

Nonetheless, it is remarkable that the confidence of the MSM appears to be sensitive to the CDR adopting the correct binding pose. In this way, the MSM metric, if only on AbAg, provides an additional metric to evaluate the generated structures from the diffusion model. The model is also very lightweight, evaluating thousands of structures in seconds on a Quadro RTX 8000 GPU.

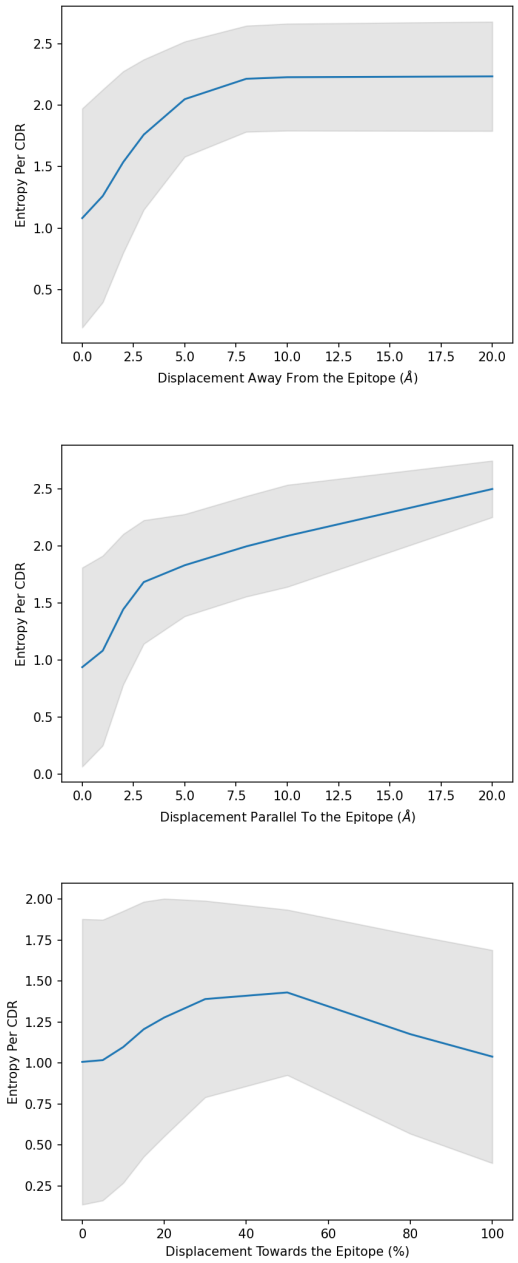


Fig. A.1 The Mean Shannon entropy as the CDR is rigidly translated in different directions relative to the epitope. From top to bottom, the CDR is translated away along the line connecting their center of masses, then towards along that line, and finally perpendicularly to that line, in the plane containing the CDR's center of mass.

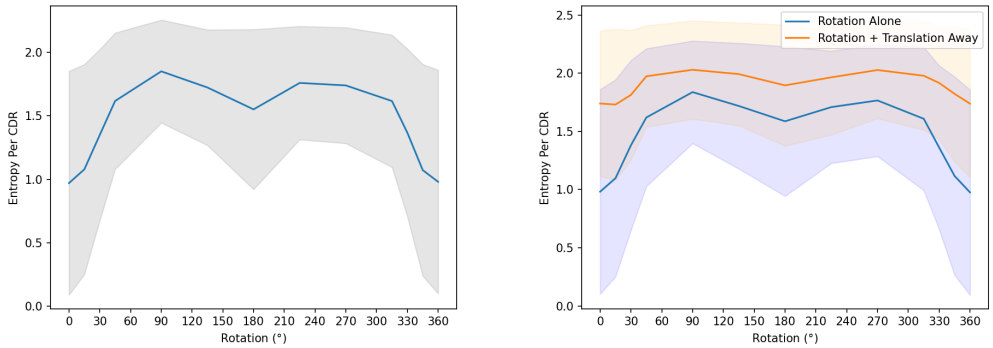


Fig. A.2 Left: The Mean Shannon entropy as the CDR is rotated about its center of mass, in the plane perpendicular to the vector connecting the center of mass of the CDR. Right: The same as left, but the CDR has now been additionally displaced 3 angstroms away from the epitope, a reasonable range for our generated structures to miss by.

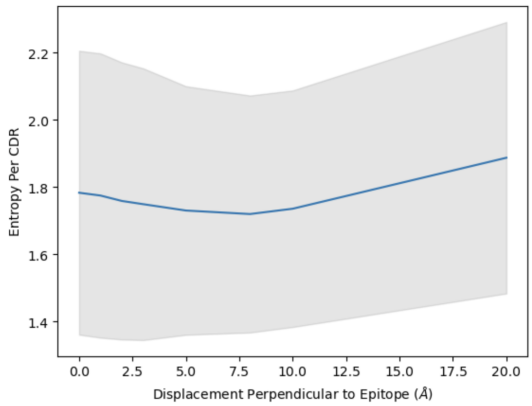


Fig. A.3 No MSM trained on SAbDab gives predictions that change when the CDR's location is corrupted.