# Triforce: Using Diffusion to Engineer New Protein-Ligand Binding Pockets

Pranav Konda
Glasgow Lab, Columbia University
Mathematics and Computer Science, pvk2108@columbia.edu

## Introduction

Proteins have many essential biological functions and often have their activity regulated by small binding molecules called ligands. The pockets, or zones, which these ligands bind are of particular structural interest in protein engineering. Furthermore, water molecules have a functional impact on stabilizing these pocket structures. We seek to develop an engineering pipeline to facilitate binding arbitrary ligands to proteins through engineering new pockets, leveraging artificial intelligence techniques that have revolutionized the field in the last decade.

## Dataset

The working dataset is restricted to proteins with binding pockets large enough to have interesting water networks whilst not being too large. Data is sourced from the Protein Data Bank, with an appropriate filtering process to select for proteins with ligands present such that their binding pockets are large enough to have such a network. This is calculated by checking for the distance from the ligand body to the nearest protein residues being under a threshold of four Angstroms. Such filtering is required due to the nature of water molecule behavior in proteins, as hydrostatic interactions vary based on the number of molecules and whether they have contact with the greater outside surface. Hence calculations of the solvent accessible surface area as a filter.

## Architecture

A diffusion model was chosen for this project due its recent successes in image generation over GAN (generative adversarial network) models [HJA20], and for protein structure design [Chu+23]. To generate sequence data for the protein, a language model is leveraged at the beginning to analyze the proposed sequence data, specifically `FAIR ESM2` [Riv+19]. This step is not done during the training process in order to ensure biologically accurate proteins. Such a diffusion model works by learning to reverse a diffusion process on a schedule: the mathematical formulation is described below.

## Formulation

The initial analysis is performed by a language model to semantically analyze the primary structure (that is, the sequence of amino acids) proposed for engineering and generate its 3D coordinates in space for further analysis, which is done via the following process of diffusion.
A diffusion model has two processes: a forward process and a backward process. The parameterization of the forward process can be described in relatively simple terms: given a variance schedule $\beta_i$, we have

$$q(\mathbf{x}_{1:T}|\mathbf{x_0}) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t, \mathbf{I}).$$

The reverse procecess is what the model learns, more specifically the joint distribution

$$p_\theta(\mathbf{x}_0 : T) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := p(\mathbf{x}_T)\prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t; t), \sigma_\theta(\mathbf{x}_t; t)),$$

where pure Gaussian noise is denoted by $p(\mathbf{x}_T) \equiv \mathcal{N}(\mathbf{x}_T, \mathbf{0}, \mathbf{I})$. [HJA20]
During the training process, we optimize the log likelihood: we seek to minimize

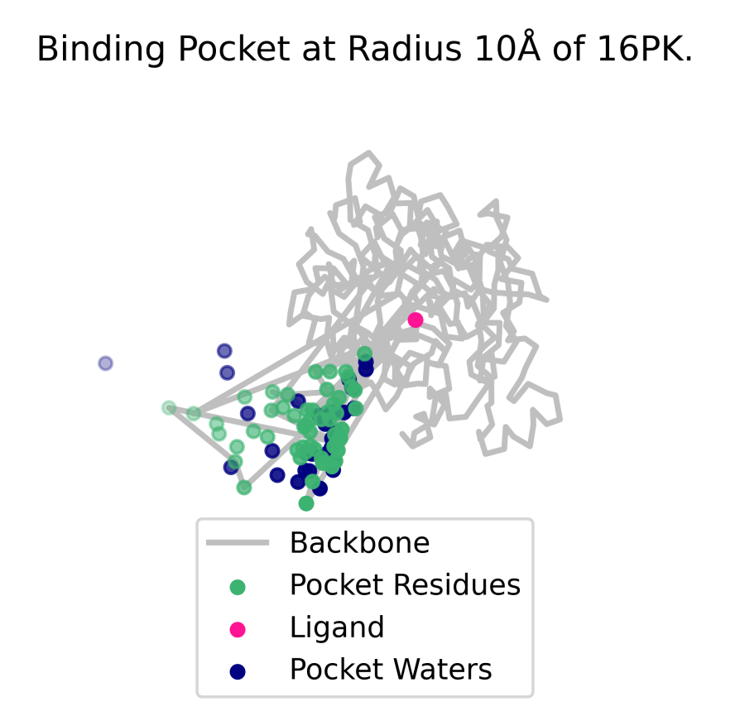
$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right].$$

The end result is a model that can properly reverse the diffusion stage processes described here. These models have shown that they are more effective than Generative Adversarial Networks (GANs) for image data and have been featured in protein design pipelines.

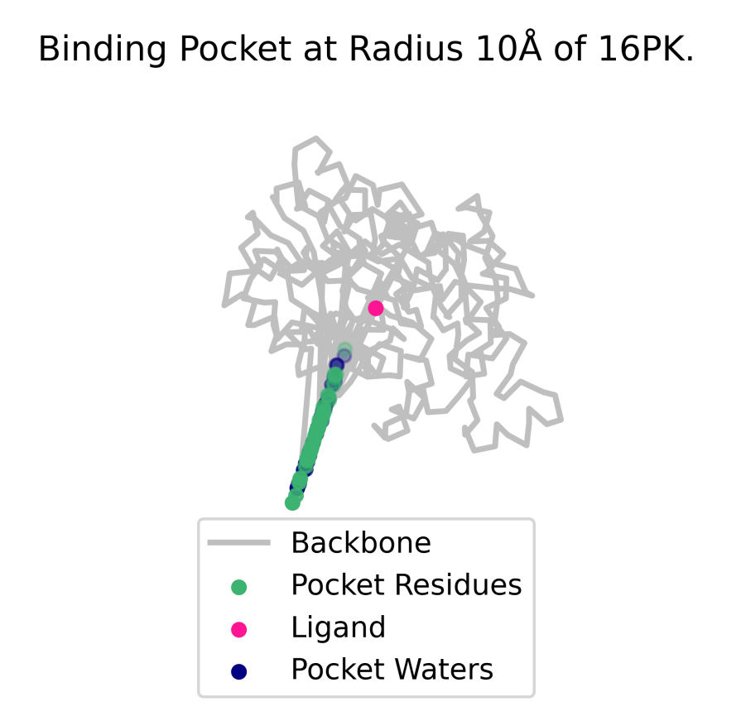
## Neural Network

In order to actually learn the reverse process we must train a neural network to approximate the score (the gradient of the log density of the distribution $p(\mathbf{x})$). Leaning on previous implementations [Chu+23], we use an U-Net like architecture as the input and output dimensionality must be identical. When working with protein data, we represent the proteins themselves as point clouds in $\mathbb{R}^3$.

## Example Represenatation and Noising



(a) Partially Noised Protein



(b) Noised Protein

## Further Direction

Further modifications will be made to the model in order to integrate it into design pipelines, such as integrating it into an iterative design process to view stages of water networks in the process. This may be accomplished by leveraging existing sidechain design processes such as `ProteinMPNN`[Dau+22]. Furthermore this pipeline must be verified on experimental data to ensure real-world applications.

## References

[Riv+19] Alexander Rives et al. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences". In: *PNAS* (2019). DOI: 10.1101/622803. URL: https://www.biorxiv.org/content/10.1101/622803v4.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models.* 2020. arXiv: 2006.11239 [cs.LG].

[Dau+22] J. Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56. DOI: 10.1126/science.add2187. eprint: https://www.science.org/doi/pdf/10.1126/science.add2187. URL: https://www.science.org/doi/abs/10.1126/science.add2187.

[Chu+23] Alexander E. Chu et al. "An all-atom protein generative model". In: *bioRxiv* (2023). DOI: 10.1101/2023.05.24.542194. eprint: https://www.biorxiv.org/content/early/2023/05/25/2023.05.24.542194.full.pdf. URL: https://www.biorxiv.org/content/early/2023/05/25/2023.05.24.542194.