Designing Novel DNA-Binding Proteins with Generative Deep Learning

by

Ido Calman

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by .....
Ido Calman
Program in Media Arts and Sciences
May 19, 2023

Certified by .....
Joseph Jacobson
Associate Professor, Program in Media Arts and Sciences

Accepted by .....
Tod Machover
Academic Head, Program in Media Arts and Sciences

Designing Novel DNA-Binding Proteins with Generative Deep Learning

by

Ido Calman

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Protein-DNA interactions play a critical role in various biological processes, such as gene regulation and genome maintenance. Designing protein backbones specifically tailored for DNA binding remains a challenging task, requiring the exploration of novel computational approaches. This thesis presents a novel framework for generating protein backbones that exhibit affinity for DNA molecules. The proposed methodology leverages Graph Neural Networks (GNNs) for encoding protein structures and diffusion models for conditional sampling. The GNNs capture the intricate relationships between amino acids in the protein backbone, allowing for the effective encoding of structural information relevant to DNA binding. The diffusion models enable the conditional generation of protein backbones, given specific DNA sequences as input. The thesis proposes a Transformer architecture and provides a practical way to diffuse from its protein encoding. The findings from this research have significant implications for the design and engineering of DNA binding proteins, facilitating advancements in fields such as synthetic biology, gene therapy, and drug development.

Thesis Supervisor: Joseph Jacobson
Title: Associate Professor, Program in Media Arts and Sciences

This thesis has been reviewed and approved by the following committee members:

Joseph Jacobson .....
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Kevin Esvelt .....
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Pranam Chatterjee .....
Assistant Professor of Biomedical Engineering
Duke University

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Joseph Jacobson, for his unwavering guidance, invaluable expertise, and constant support throughout the course of this thesis. Their insightful feedback, patience, and encouragement have been instrumental in shaping this research endeavor. I am sincerely grateful for the countless fruitful conversations, stimulating discussions, and the continuous motivation provided by my advisor.

Furthermore, I would like to express a special acknowledgement to Allan S. Costa and Manvitha Ponnapati for their valuable contributions and support. Their insights, technical assistance, and dedication have significantly contributed to the success of this research. I am grateful for their valuable input, which has enriched the depth and quality of the work presented in this thesis.

Transitioning from a distinct field within computer science to the intricate realm of computational protein design presented a genuine and formidable challenge to me and I am fortunate to be surrounded by extraordinary individuals in the Molecular Machines group who made this transition dramatically smoother.

I would also like to extend my gratitude to the entire research group for their collaborative spirit, which fostered a stimulating and dynamic environment. The exchange of ideas, constructive criticisms, and shared knowledge within the group greatly enriched my understanding and enhanced the quality of this thesis. The collective effort and camaraderie within the group have been truly inspiring.

Lastly, I would like to extend my heartfelt appreciation to my family, friends, and loved ones for their unwavering encouragement, understanding, and patience throughout this research journey. Their support has been a constant source of strength and inspiration.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Targeted genome engineering is a broadly applicable tool in therapeutic design, plant and animal research, and many other biomedical purposes. For example, Sickle Cell Disease (SCD) is caused by a single DNA mutation in the β-globin gene HBB [1]. This mutation is associated with an inversion of one base pair in the sixth codon of the β-globin chain such that the translated amino acid, originally Glutamine is replaced by Valine [2]. Recent in-vivo gene therapy solutions which rely on the CRISPR-Cas9 system show promising results in targeting this locus and cure SCD on mice [3]. One of the main challenges is to predetermine the precise region in a genome and guide the cleaving protein to it. Targeting the desired DNA region is normally the task of the binding domain in the synthetically engineered nuclease. Different approaches for genome engineering protein design explore various methods for constructing programmable binding domains, and specificity is reached by binding to longer DNA regions.

1.2 Natural Protein-DNA Binders

Proteins that bind to single-stranded and double-stranded DNA exist in nature and function in various ways. For instance, transcription factors (TFs) are protein molecules

that bind to specific DNA sequences and modulate the initiation or rate of RNA synthesis by recruiting or blocking the RNA polymerase complex, thereby playing a crucial role in the regulation of gene expression and transcriptional control in the cell.

Transcription factor proteins have high binding affinity to specific DNA sequences in the promoter part of the gene, and thus when their concentration in the cell increases, they inhibit more promoter and repress the gene transcription to mRNA. This process is commonly known as *feedback inhibition*. TFs can function as either repressors or operators depending on their structure. The genome engineering field seeks to explore synthetic possibilities that mimic this type of interaction between proteins and DNA in order to design custom nucleases which bind to the desired genetic address.

1.3 Traditional Genome Engineering

1.3.1 Zinc Finger Nuclease

Traditional methods of engineering programmable ZFNs involve designing two main components: the binding domain and the cleavage domain. In most cases, cleaving is performed by FokI, a type II restriction enzyme that can cleave non-specifically a short distance from a bound DNA sequence [4]. Programmability of the binding domain is achieved by assembly of multiple Zinc Finger motifs, each responsible for recognition of three base pairs (Fig 1-1). $Cys_2 - His_2$ domains are the most abundant DNA-binding motif in eukaryotes and consist of $\sim$ 30 residues that fold into a $\beta\beta\alpha$ -structure coordinated by a zinc ion [5].

A key challenge in effectively utilizing ZFN technology is the inherent difficulty and time-consuming nature of ZFN design [6]. This is primarily attributed to the imperfect modular nature of tandem zinc fingers, wherein the assembled ZFNs may not consistently exhibit high affinity for the targeted sequence, which is a composite of the 3-base pair binding sequence of each individual zinc finger.

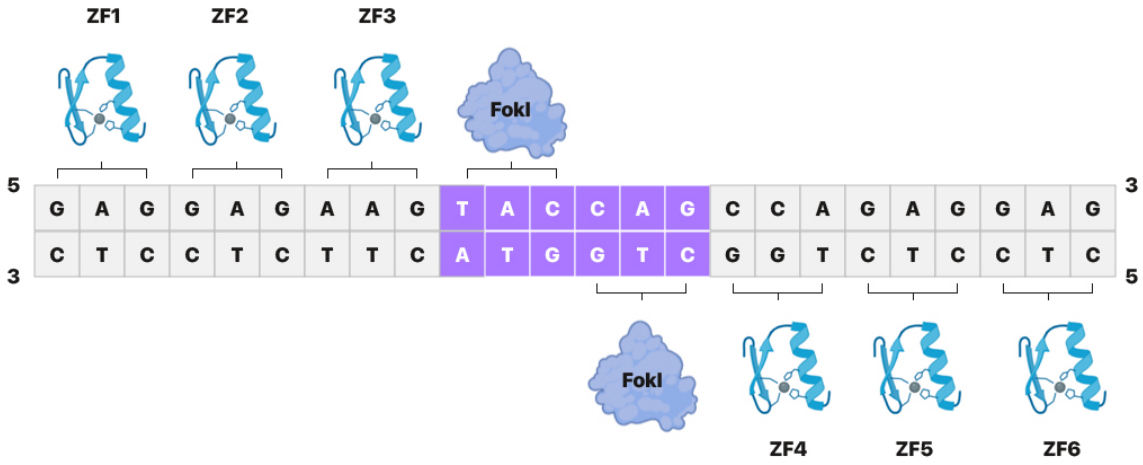

Figure 1-1: An 18-base pair ZFN. Each Zinc Finger motif recognizes 3 bases. 3 fingers are synthesized in a complex with the FokI restriction enzyme which performs a non-specific cleavage. Programmability is achieved by handpicking the 6 different fingers to match the desired sequence from both ends.

1.3.2 Transcription Activator-like Effector Nuclease

Transcription activator-like effector nucleases (TALENs) are yet another type of engineered DNA-binding protein that can be used to introduce targeted changes in the genome of living cells. TALENs are composed of a DNA-binding domain derived from transcription activator-like effectors (TALEs), which are naturally occurring proteins found in some bacteria, fused to a nuclease domain derived from the FokI endonuclease.

The DNA-binding domain of TALENs consists of a repeating unit of 34 amino acids that can be rearranged to create a specific DNA-binding sequence. The DNA-binding specificity of TALENs is determined by the amino acid sequence of a series of tandem repeats within the DNA-binding domain, each of which recognizes a specific nucleotide in the target DNA sequence. By linking these repeating units together in a specific order, TALENs can be designed to recognize virtually any DNA sequence.

Once TALENs have bound to their target DNA sequence, the FokI nuclease domain induces a double-stranded break in the DNA. This break can be repaired by the cell's natural repair mechanisms, either by non-homologous end joining (NHEJ), which often results in the insertion or deletion of a small number of nucleotides at

the site of the break, or by homology-directed repair (HDR), which can be used to introduce specific changes or insertions at the site of the break using a DNA repair template [7].

One notable drawback of TALENs is their considerably larger size in comparison to ZFNs. Typically, the cDNA encoding a TALEN spans around 3 kb, while a ZFN's cDNA is approximately 1 kb. This size disparity presents challenges in delivering and expressing TALENs in cells, especially when compared to ZFNs [8]. Moreover, the larger size of TALENs makes them less favorable for therapeutic applications where delivery is accomplished through viral vectors with limited cargo capacity, such as adeno-associated virus (AAV) with less than 5 kb, or as RNA molecules.

1.3.3 CRISPR-Cas9

CRISPR-Cas9 is a revolutionary genome editing technology that utilizes the bacterial adaptive immune system to target and modify specific DNA sequences in a variety of organisms. The CRISPR system consists of two main components: the CRISPR RNA (crRNA) and the CRISPR-associated protein 9 (Cas9). The crRNA is designed to recognize and bind to a specific target DNA sequence, while Cas9 functions as a molecular scissors that cuts the DNA at the targeted location.

The CRISPR-Cas9 system is guided by a short RNA molecule, which is designed to match the specific DNA sequence to be edited. The RNA molecule is part of a complex with the Cas9 protein, which is responsible for cutting the DNA at the designated location. Once the RNA molecule and Cas9 protein complex find their target DNA sequence, the Cas9 protein cuts the DNA, creating a double-strand break.

This break then triggers the cell's natural DNA repair mechanisms, which can be harnessed to introduce specific changes to the DNA sequence. This process can be used to create specific genetic mutations, insert new genes or remove unwanted ones, and study the function of specific genes.

Although CRISPR-Cas9 has many advantages for genome engineering, there are also some potential limitations and flaws that need to be considered.

Off-target effects: One of the main concerns with CRISPR-Cas9 is the potential for

off-target effects, where the system cuts DNA at unintended sites that have sequence similarities to the intended target. This can lead to unintended mutations and other negative consequences. Despite the presence of off-target effects in all genome editing systems, the significant drawback of CRISPR-Cas9 technology lies in its elevated occurrence rate ($\geq 50\%$) of unpredictable off-target effects, which poses a considerable disadvantage [9].

Delivery challenges: Another limitation is the difficulty of delivering the CRISPR-Cas9 components to specific cells or tissues in vivo. Efficient and targeted delivery is important for clinical applications, but this remains a challenge for many applications. An instance of this can be seen in the extensive utilization of viral vectors in both in vivo and in vitro contexts; however, this approach carries numerous limitations, including immune responses and insertional constraints. [10].

Complexity: While CRISPR-Cas9 is simpler and more efficient than previous genome editing methods, it still requires specialized knowledge and resources to implement effectively, which could limit its accessibility to some researchers and institutions.

1.4 Technological Background

Deep learning-driven approaches to protein design are becoming more and more evident as ways to generate de novo amino acid sequences for a predetermined function. Deep neural networks turned into a transformative biotechnology approach increasingly following DeepMind's AlphaFold [11] results in the CASP13 [12] and CASP14 [13] competition which have beaten decades-long benchmarks in protein folding. Later in 2021 they published an extensive folding inference on the entire human proteome [14].

Diaz et al. showed capabilities of 3D convolutional neural networks (3DCNN) in predicting neighboring amino acids based on the atomic backbone structure of a protein in what they refer to as "microenvironment" [15]. These machine-learning methods were examined on enzymatic depolymerization of Polyethylene terephthalate

(PET), which resulted in generation of newly designed PET hydrolases.

Developments in deep neural networks for protein design gave rise to the research of de novo generation, a field which were mostly dominated by physics-based models before [16]. Anishchenko et. al [17] used gradient descent on pre-trained sequence-to-structure networks (AlphaFold and RoseTTAFold) in order to "hallucinate" proteins from noise. Some of these generated proteins were synthesized and their X-ray crystallography showed that they indeed fold in nature to the predicted structure. Later on, ProteinMPNN [18] was introduced to address the problem of Motif Scaffolding, that is, generation of a rigid structure which incorporates some motif of interest. With a Graph Neural Network (GNN) architecture, the authors trained a model that goes from structure to sequence. They used some tradition methods to produce a template scaffold structure and then made the GNN suggest novel sequences. This problem is of great significance since it would demonstrate a way to produce artificial proteins with some known function.

The astonishing results of the aforementioned generation of text and images inspired the research of solving protein interaction problems using Diffusion models. DiffDock [19] achieved state of the art results by a large margin for the protein-ligand docking task. They used diffusion to generate translation and rotation of a known binder to determine the exact point of interaction. SCMDiff [20] applied diffusion models to sample from an E3 equivarient GNN to solve the motif scaffolding problem.

Language models are known to play a key role in a variety of Natural Language Processing (NLP) tasks. In 2019 Attention [21] models were introduced and architectures such as BERT, GPT3 and T5 [22, 23, 24] are only a short list of attention-based models that gained popularity and prevailed over traditional machine learning approaches. Similar techniques were followed to learn latent representation of proteins in the form of amino acid sequences. One of the most dominant large models that were trained was ESM [25]. Representations learned from this model produced high accuracy in structural supervised tasks such as mutational effects and secondary structure prediction. Later, researcher showed that a complete sequence to structure

model competitive with AlphaFold can be achieved from the ESM representations only [26]. Language models were utilized not only on amino acid sequences but on genome-scale as well [27] and exhibited promising results in downstream tasks such as prediction of SARS-CoV-2 evolutionary dynamics.

1.5 Related Work

The field of protein backbone generation continues to witness active research and advancements as researchers strive to improve the understanding and modeling of protein structures. Among the various approaches explored, deep learning models combined with diffusion methods have gained significant attention due to their ability to capture complex dependencies and generate diverse and novel protein structures. This approach leverages the power of deep neural networks to learn representations of protein sequences and utilize diffusion-based algorithms to explore the conformational space of protein backbones. By incorporating unsupervised learning techniques, these models can effectively capture the underlying patterns and intricacies of protein structures without relying on labeled data. The use of diffusion methods further facilitates the exploration of new structures by enabling the sampling of diverse conformations and providing a rich set of potential solutions. To date, there is a lack of published methodologies that showcase the conditional generation of protein structures based on DNA, which is the central contribution of this thesis.

Currently, RFDiffusion stands as the prominent diffusion-based algorithm for protein backbone generation, as evidenced by its widespread recognition [28]. This algorithm builds upon the highly acclaimed RoseTTAfold protein folding network [29], which has demonstrated comparable performance to AlphaFold, a state-of-the-art protein structure prediction model. RFDiffusion initializes its model weights based on the pretraining of the folding algorithm and subsequently learns to predict the diffusion noise. This model generates backbone both conditionally unconditionally and it has shown some promising results in designing symmetric oligomers and in the motif scaffolding problem.

In the context of addressing the motif-scaffolding problem using diffusion models, SMCDiff [20] introduces a viable solution. Their proposed approach involves employing an equivariant graph neural network, which shares similarities with the architecture presented in this thesis. The generation process in SMCDiff employs particle filtering based on sequential Monte Carlo sampling, as outlined by Doucet et al. [30]. Notably, this methodology has demonstrated preliminary success in generating scaffolds with sequences spanning up to 80 residues.

SE(3)-Diffusion [31] is a follow-up work by the same authors which is the closest approach to the work done in this thesis. The key observation is the deployment of "frames" instead of pure three-dimensional coordinates. The frames are an alternative parametrization to the $N - C_\alpha - C - O$ obtained by performing the Gram-Schmidt operation on the vectors directed from C_α to the Nitrogen and Carbon. That parameterizes the $N - C_\alpha - C$ placements with respect to the frame translation, set to the C_α coordinates. An additional torsion angle is, thus, required to determine the placement of the Oxygen atom. The neural network architecture used in this paper is the well-known SE(3)-Transformer which is at the heart of many other trending architectures. While this paper demonstrates a clean approach to diffusion sampling of proteins, it does not tackle the conditional problem which is crucial to derive DNA binders.

1.6 Datasets and Methods

1.6.1 Evaluation Metrics

Ideally, generated protein candidates should be evaluated with a binding affinity assay. However, these are considered expensive and require expertise in protein expression, purification and assay control. Instead, we adopt an in-silico approximation to the validity of generated proteins that is performed with protein folding systems. A generated structure is considered valid if it reaches a high *Reconstruction Score* which is described as follows: First, we apply a structure-to-sequence prediction to the

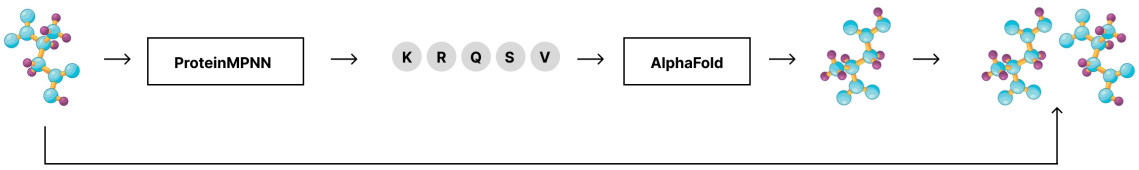

Figure 1-2: Reconstruction score. The pipeline involves two algorithms - inverse folding (ProteinMPNN) and folding (AlphaFold). The final result is compared with the generated backbone.

generated backbone. Then, we fold the resulting sequence using a well-proven folding algorithm (e.g. AlphaFold2). Finally, our score is the TM-alignment of our newly generated backbone and the last step's output. Let $\xi(\mathbf{x}) : \mathbb{R}^{3\times n} \to |V|^n$ be our sequence-to-structure (inverse-folding) algorithm. $|V| = 20$ is the vocabulary of all possible amino acids and n is the protein's sequence length. Let $\psi(\mathbf{t}) : |V|^n \to \mathbb{R}^{3\times n}$ be our folding algorithm. Then our reconstruction metric for a protein generation process $\phi(\mathbf{x})$ is:

$$RS_\phi(\mathbf{x}) := \sqrt{\|\phi(\mathbf{x}) - \psi(\xi(\phi(\mathbf{x})))\|^2} \tag{1.1}$$

1.6.2 Data Mining

In the lack of a standard Protein-DNA dataset, we sought to filter X-ray crystallized structures from the Protein Data Bank (PDB). There exists a trade off between the number of instances and the quality of data we pursue. An accurate X-ray crystallization is considered below 2Å, and the "simplest" instances contain one DNA strand and one protein chain. However, due to the sparsity of such available structures we experiment with multiple different datasets.

In order to obtain a comprehensive and refined dataset of protein-DNA complexes, we employed a clustering and splitting methodology. This approach involved applying a series of systematic procedures to group similar complexes together and subsequently partitioning the dataset into distinct subsets, resulting in an extensive and well-curated collection of protein-DNA complexes (see fig 1-4). Our protocol

involves:

- **Eliminating non-interacting chains**: A criterion was established whereby a minimum of 10 contacts (with distances less than 6Å) were required between the protein's C_α atom and any non-hydrogen atom from the DNA. By implementing this criterion, protein-DNA pairs lacking substantial interaction were effectively filtered out.

- **Hydrogen bonds**: The Baker-Hubbard algorithm was employed to assess the presence of hydrogen bonds within the protein structure. Within the PDB, the representation of DNA often involves storing the two strands as separate chains. In order to identify the complementary DNA strands accurately, a sequence-based search was initially conducted. However, it was observed that certain cases exhibited discrepancies such as variations in DNA lengths. To address such inconsistencies, an additional step was introduced to identify hydrogen bonds between the two strands. Subsequently, the strand exhibiting the highest number of hydrogen bonds was selected as the complementary strand, ensuring a more reliable determination of the complementary DNA strands.

- **Manual error detection**: To ensure the reliability and accuracy of the filtered structures following the aforementioned selection criteria, a manual verification process was conducted. This meticulous examination involved a thorough inspection of the structures to identify and eliminate any erroneous or misleading entries.

The outlined protocol not only facilitated the exclusion of redundant protein complexes as distinct examples but also facilitated the expansion of the dataset by partitioning complexes into interaction motifs. As a result, the processed dataset exclusively comprised monomers featuring a singular double-stranded DNA chain. This approach ensured the generation of a diverse and representative dataset, free from duplications and characterized by distinct interaction motifs, thus enhancing the breadth and richness of the dataset for subsequent analyses and investigations.

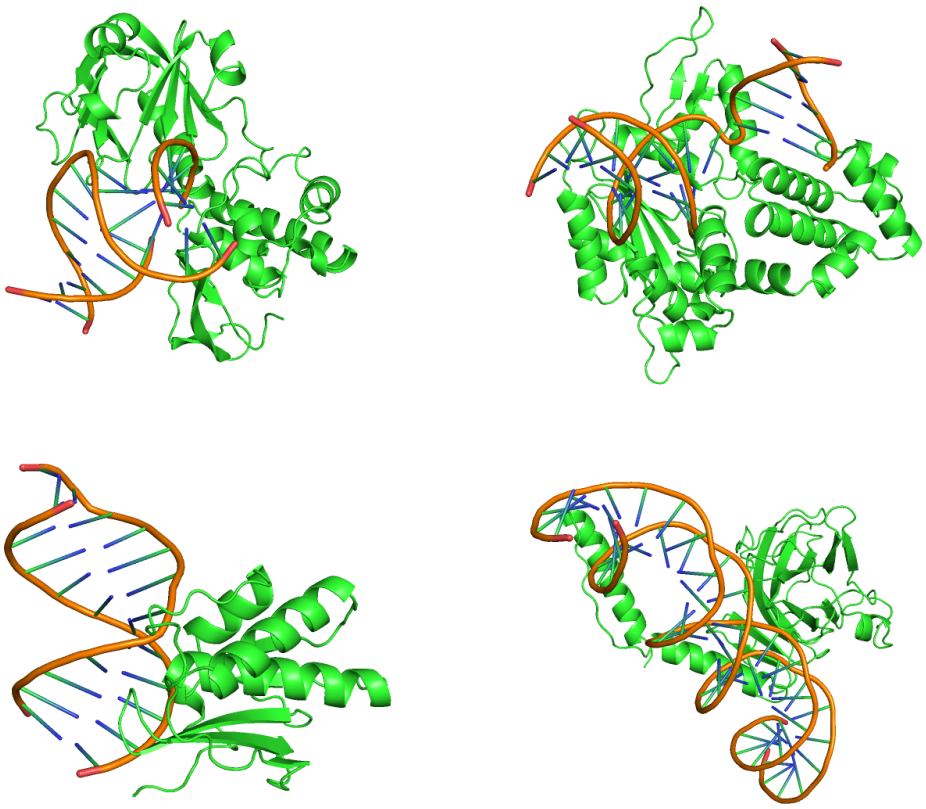

Figure 1-3: Examples of extracted protein-DNA complexes from the PDB.

DNA	Protein	Resolution	Instances
≥ 1	≥ 1	Any	7382
≥ 1	≥ 1	≤ 2.5Å	3452
1	1	Any	1388
1	1	≤ 2.5Å	857

Table 1.1: Available Protein-DNA data instances

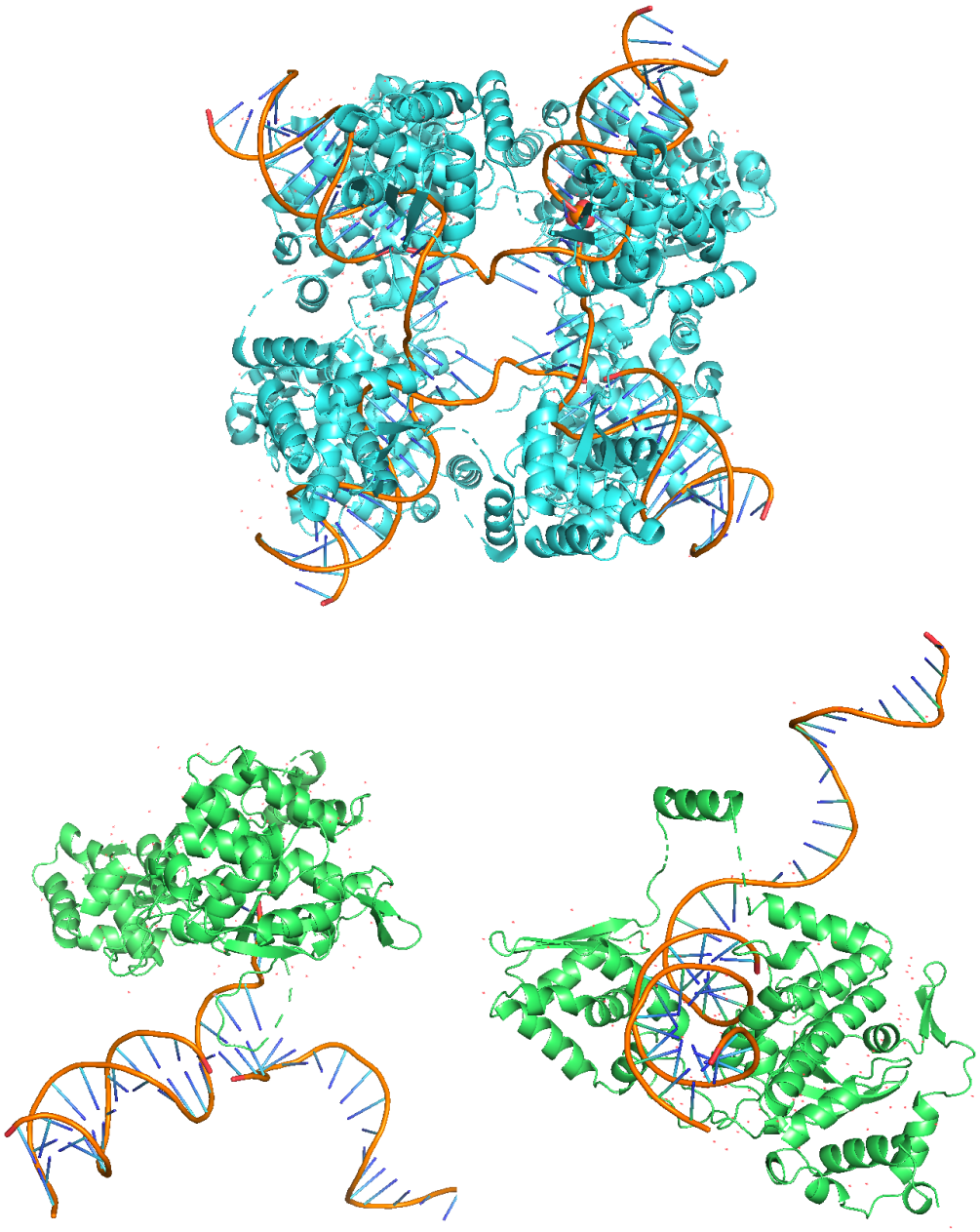

Figure 1-4: Splitting multi-chain protein complexes. Top: the PDB instance with multiple amino acid and nucleic chains. Left: a detected non-interaction which is disregarded. Right: a protein-dna monomer interaction.

Chapter 2

Protein Deep Learning Representation

Protein backbone generation is a fundamental step in computational protein structure prediction, with wide-ranging implications for drug discovery, biotechnology, and molecular biology research. Designing novel proteins in the computational context is the ability to determine the coordinates of amino acid chains in a Euclidean three-dimensional space. The term "backbone" is referred to the common atoms of all twenty different amino acids: $N - C_\alpha - C$ (See figure 2-1). Amino acids differ in a set of atoms which are bound to the C_α atom which are referred as "side chain". These atoms are usually not subject to generation and are considered implied, so generation of the three backbone atoms coordinates suffices in a structure prediction.

Deep learning methods such as AlphaFold, OmegaFold and RoseTTAFold [11] [29] [32] have emerged as a highly effective means of predicting accurate protein backbone structures from primary amino acid sequences. Mathematically, these neural networks approximate the distribution of $\mathbf{P}(structure|sequence)$ For given pairs of amino acid tokens and three-dimensional coordinates for of corresponding backbone atoms. Protein generation, in its unconditional form, strives to learn the distribution: $\mathbf{P}(structure)$ A valid approach to learning such distribution can be thought of

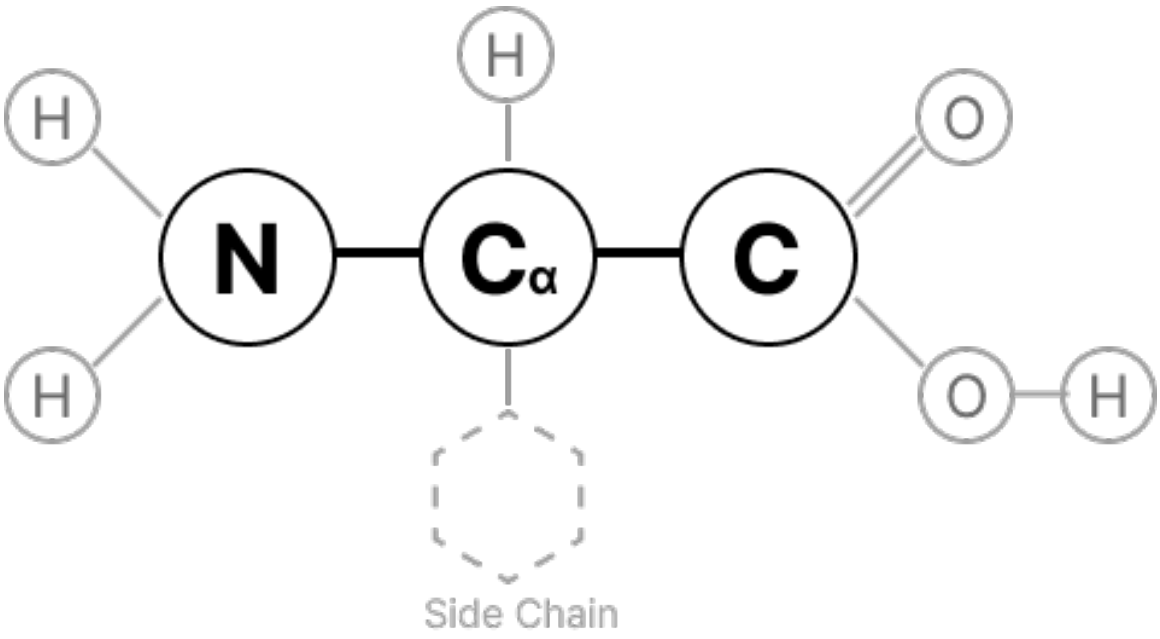

Figure 2-1: Amino acid general structure. In bold are the common Nitrogen and two Carbon atoms which are common to all amino acids. The side chain bound to C_α differs between different types of amino acid.

as learning in a Bayesian fashion with a known protein structure predictor:

$$\mathbf{P}(structure) = \mathbf{P}(structure|sequence) \cdot \mathbf{P}(sequence) \tag{2.1}$$

Where $\mathbf{P}(sequence)$ can be learned, for example, by a language model such as ESM [25] [33]. In that formulation, generation of an arbitrary protein would be achieved by sampling from the language model, followed by prediction of its structure using a sequence-to-structure predictor. However, recent research shows that the structure distribution can be learned unsupervised directly. Modelling structural distribution is beneficial to problems where a sequence is completely or partially missing and when structures are to be generated conditioned on other structural sub-components such as the Motif Scaffolding problem.

2.1 Equivariant Neural Networks

A key aspect to a deep neural network's capability to generalize is being able to capture symmetries within input data. This is particularly true to problems where similar data instances may have varied values, but the relations between segments of the instance are preserved. In molecular structural data, the numerical values of the coordinates in space are less meaningful than the distances between individual atoms. This rapidly growing field is often referred as *geometric deep learning* [34]. Our goal in this field is to construct deep learning architectures that are compatible with a symmetry group G that acts transitively on the input data. We say that a function $\phi : X \to Y$ is *equivariant* [35] to transformations $T_g : X \to X$ and $S_g : Y \to Y$, $g \in G$ if for every input $\mathbf{x}$ it suffices:

$$\phi(T_g(\mathbf{x})) = S_g(\phi(\mathbf{x})) \tag{2.2}$$

In the case of proteins, we are particularly interested in the set of transformations $\mathbf{E}(3)$ which are the rotation, translation and permutation in 3D space.

1. Equivariance to translation of input $\mathbf{x} \in \mathbb{R}^{n\times 3}$ by $g \in \mathbb{R}^3$ where addition is element-wise: $\mathbf{x} + g = (\mathbf{x}_1 + g, ..., \mathbf{x}_n + g)$ is defined as: $\phi(\mathbf{x} + g) = \mathbf{y} + g$.

2. Equivariance to rotation and reflection by some orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{3\times 3}$, $\mathbf{Qx} = (\mathbf{Qx}_1, ..., \mathbf{Qx}_n)$ is defined as: $\phi(\mathbf{Qx}) = \mathbf{Qy}$.

3. Equivariance to permutation simply means that if $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ were to be permuted in some different order $\mathbf{P}(\mathbf{x})$, then: $\phi(\mathbf{P}(\mathbf{x})) = \mathbf{P}(\mathbf{y})$.

These three conditions should suffice for any deep neural network architecture we choose that would capture protein structural data.

2.2 Protein Graph Representation

A natural choice for representing protein data and molecular data in general is the graph structure. Molecules are in essence a collection of atoms and bonds, which

conveniently fits in the $G = (V, E)$ formulation, where $V = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ is a set of vertices assigned to each atom $\mathbf{x}_i \in \mathbb{R}^3$ in the molecule and $E \subseteq V \times V$ is the set of edges that encompasses an atom-to-atom relation. We denote e_{ij} the edge between atom $\mathbf{x}_i$ to $\mathbf{x}_j$. In most cases we are interested in a fully connected graph so $E = \{e_{ij}\}_{i,j=1}^{n}$ and we introduce an adjacency (symmetric) matrix $A \in \mathbb{R}^{n \times n}$ which would store local information, primarily for computational efficiency. Most commonly the adjacency matrix will indicate whether nodes belong to the same neighborhood, meaning:

$$A_{ij} = A_{ji} = \left\{ \begin{array}{ll} 1, & \mathbf{x}_i \leftrightarrow \mathbf{x}_j \\ 0, & else \end{array} \right\}$$

2.3 Graph Neural Networks

With our goal to learn the atomic coordinates distribution in mind, we strive to learn representations in some latent space for our molecules described as a graph. Graph Neural Networks (GNNs) provide solutions to processing the graphical structured data and learn meaningful vector representations for nodes, edges and the graph in its entirety. GNNs gained popularity in recent years thanks to successes in social networks analysis [36], stock market predictions [37], physical system dynamics [38] and many other research fields [39].

With the graph formulation, a protein latent representation is being updated in the following manner: Assume $h_i^k \in \mathbb{R}^d$ is the representation of $\mathbf{x}_i$ within layer k. And let $H_k \in \mathbb{R}^{n \times d}$ be the matrix of all such hidden vectors. We also denote $\tilde{A} = A + \mathbf{I}$ the adjacency matrix with self-reference so every node is adjacent to itself. Then we apply:

$$H_{k+1} = \sigma(\tilde{A} H_k W_k)$$

Where W_k is a learnable linear transformation and σ is some non-linear function. Or equivalently:

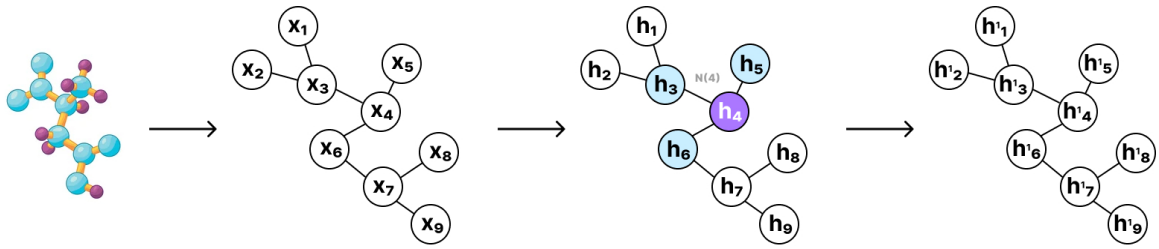

Figure 2-2: Transformation of a plain protein intro a graph formulation. Features h_i are updated at each network step according to neighbor $N(i)$ features.

$$h_i^{k+1} = \sigma\left(\sum_{j \in N(i)} W_k h_j^k\right)$$

Where $N(i)$ is the set of all neighboring atoms to node $\mathbf{x}_i$. Finally, notice that the matrix A may induce scaling issues since nodes with more neighbors would result in a larger value, so the neighborhood normalization $\frac{1}{|N(i)|}$ normalisation brings the layer update rule to:

$$h_i^{k+1} = \sigma\left(\frac{1}{|N(i)|}\sum_{j \in N(i)} W_k h_j^k\right) \tag{2.3}$$

2.3.1 Graph Convolutional Layer

Normalization of neighboring nodes, as simply presented in equation 2.3 can be expressed in a more symmetric way. Kipf & Welling [40] popularized such update rule, namely the Graph Convolutional Layer (GCL) with a slight moderation to the former equation:

$$H_{k+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H_k W_k\right) \tag{2.4}$$

Here the matrix $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the degree matrix of $\tilde{A}$. Or in other words:

$$h_i^{k+1} = \sigma\left(\sum_{j \in N(i)} \frac{1}{|N(i)||N(j)|} W_k h_j^k\right)$$

The GCL method, presented in ICLR 2017 is currently the most cited GNN paper and considered a simple but effective way to process node information in semi-supervised learning problems. One of the problems with GCL for protein representations lies in its lack of explicit edge features, that is, we wish to encourage accumulation of potential atom-to-atom dynamics, which will be represented in latent edge vectors.

2.3.2 Message Passing Neural Networks

As a possible solution to a system that focuses on edge features, a suggested method is the notion of a *message*, which is a bit of information that is passed from one node to its neighbors in each layer of the network. The nodes then aggregate all the messages they receive to produce a more graph-aware representation. The Message Passing Neural Network (MPNN) suggested by Gilmer et al. [41] consists of messages m_{ij} between nodes $i \to j$ computed via some *message function* $M(h_i, h_j, e_{ij})$ and is referred to as the *message passing phase*:

$$m_{ij}^{k+1} = \sum_{j \in N(i)} M^k(h_i^k, h_j^k, e_{ij}^k) \tag{2.5}$$

Note that the sum over neighboring nodes corresponds to the aggregation of all incoming messages. The second step is updating the latent variables h_i and is referred to as the *readoutphase*. Readout is performed via the function $U(h_i, m_i)$:

$$h_i^{k+1} = U^k(h_i^k, m_i^{k+1}) \tag{2.6}$$

The functions M and U are most-commonly being calculated with an MLP. The MPNNs are considered very potent in terms of edge features expressibility however, in practice they suffer from problems with storage when facing a large amount of edges. MPNNs are thus practically applicable mostly to small graphs. So as a middle ground, we step back to our original normalized update rule as described in equation 2.3 and define the normalization term $\frac{1}{|N(i)|}$ in a more general manner:

$$h_i^{k+1} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W_k h_j^k\right) \tag{2.7}$$

2.3.3 Graph Attention Layer

To mitigate on MPNNs scalability issues, instead of applying the message passing phase on each update step with cost $O(d^2n^2)$, Veličkovič et al. [42] presented a method that takes advantage of the recently popular self-attention mechanism.

$$\begin{aligned} e_{ij}^{k+1} &= a^k(W^k h_i^k, W^k h_j^k) \\ \alpha_{ij} &= softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} \end{aligned} \tag{2.8}$$

Where $a : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the self-attention coefficients function over the nodes. This update rule is referred to as Graph Attention Layer (GAT). The GAT formulation lays the foundation for the proposed architecture in this thesis that will be unveiled in the next section.

2.3.4 Equivariant Graph Neural Networks

Combining our two interests: representing proteins as a graph and preserving E(3) equivariance we arrive at a pathway for architectures that address our concerns. The Equivariant Convolutional Layer (EGCL) was proposed [43] for these purposes and is a slight modification from the MPNN layer presented in section 2.3.2. To preserve E(3) equivariance (that is: rotation, translation and reflection) the authors suggest the following update rule:

$$\begin{aligned} m_{ij} &= M(h_i^k, h_j^k, \|\mathbf{x}_i^k - \mathbf{x}_j^k\|^2, e_{ij}^k) \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + C \sum_{j \in N(i)} (\mathbf{x}_i^k - \mathbf{x}_j^k) \phi_x(m_{ij}) \end{aligned} \tag{2.9}$$

Where $\phi_x : \mathbb{R}^d \to \mathbb{R}$ is some function that produces a scalar output from the messages m_{ij}. C is a scalar normalization factor chosen to be $\frac{1}{|N(i)|-1}$. Notice that

the message passing network now receives as an input the (updating) square distance $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ between neighboring inputs $\mathbf{x}_i$ and $\mathbf{x}_j$. The addition of distances between inputs is key to preserve equivariance.

2.4 Transformers

Inspired by the remarkable success of the Transformer architecture [21]. in natural language processing (NLP), this study aims to adapt its principles to the domain of protein design. Proteins play crucial roles in biological processes and exhibit complex structural and functional relationships. By leveraging the Transformer's attention mechanism, which excels at capturing long-range dependencies, we aim to enhance the modeling of protein sequences and structures. We propose a modified version of the Transformer that incorporates domain-specific adaptations, such as utilizing physicochemical features and incorporating protein-specific positional encoding. This allows the model to effectively capture the intricate relationships between amino acids and their spatial arrangements.

The *Self-Attention* mechanism in the Transformer is based on the concept of scaled dot-product attention. Given an input sequence of length N, the self-attention mechanism computes three key components: *query*, *key*, and *value*. In matrix form, denote $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. These components are linearly transformed using learnable weight matrices, and their dot products determine the attention scores. The attention scores are then scaled and softmaxed to obtain the attention weights. The weighted sum of the values, weighted by the attention weights, is computed to produce the attention output.

$$\mathbf{Z} = softmax(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d}})\mathbf{V} \tag{2.10}$$

In the protein formulation, these would correspond to connections between backbone coordinates. The learned weights between the input amino acids refer to how much attention should their representation contribute to the output calculation. All representation matrices are achieved by training the respective $W^{\mathbf{Q}}$, $W^{\mathbf{K}}$ and $W^{\mathbf{V}}$

weights and multiplying by the input $\mathbf{X}$ so for instance: $\mathbf{Q} = \mathbf{X}W^{\mathbf{Q}}$. The normalizing factor d refers to the dimension of $\mathbf{K}$. The hidden output $\mathbf{Z}$ of this component, commonly referred as the attention *head* can be further propagated into following similar layers.

2.5 E(n)-Transformer

With the acquisition of all the necessary components, we are now equipped to construct the ultimate architecture selection for the representation of protein graphs. The architecture we propose is a graph neural network that extends the principles of EGNN (see 2.3.4). The hidden layers are calculated using a Transformer model, incorporating multi-head attention layers. Additionally, the network incorporates an adjacency matrix to confine computations to local neighborhoods, enabling efficient and targeted information propagation within the graph structure. Here we present a protein encoder, a neural network designed to process coordinate representations of proteins and generate output coordinates of the same dimensionality. The protein encoder serves as a foundation for subsequent extensions, enabling the encoding of diverse sets of coordinates that are crucial for protein design purposes. In order to provide a comprehensive description of the architecture, we present each component individually:

2.5.1 Rotary Embeddings

To preserve E(n) equivariance we calculate a relative distance matrix D^r. This matrix is the input for a multi-layer perceptron (MLP) which converts the distances into some higher dimensionality vector.

$$D^r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$$

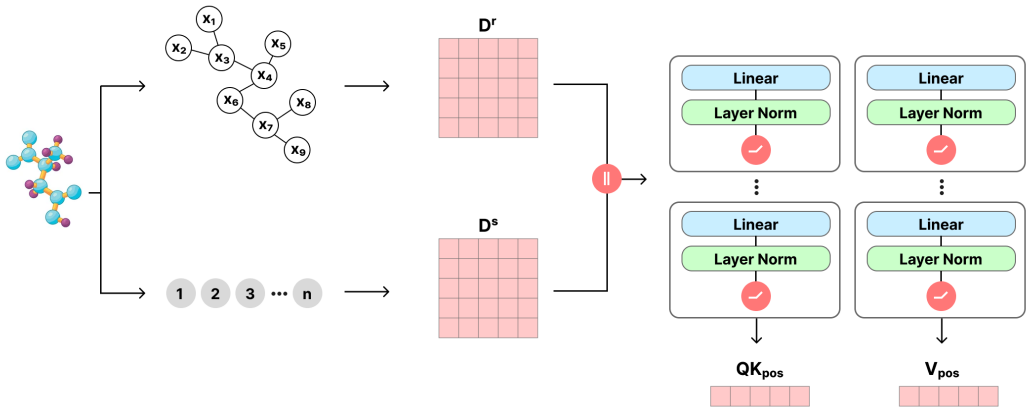

Figure 2-3: Rotary and sequence positional embeddings. on the left, the input is divided to the coordinates and an array of indices. Then, the matrics D^s and D^r are calculated respectively. Finally, a block with two multi-layer perceptrons outputs the positional embeddings for **QK** and **V**. The chosen activation function for each layer is SiLU and the || icon stands for matrix concatenation.

2.5.2 Sequence Position Embeddings

Given that our current input solely consists of a spatial graph comprising coordinates in $\mathbb{R}^3$, our objective was to incorporate positional information regarding each coordinate's location within the protein sequence. By integrating this information, we aim to guide the network towards learning a set of coordinates that accurately corresponds to a chain of amino acids, rather than arbitrary point clouds. The relative sequence position matrix D^s is given by:

$$D^s_{ij} = i - j$$

Both positional embeddings are concatenated and fed through a common position MLP with 2 vector outputs. (See fig 2-3).

2.5.3 Coordinate Update Layer

The output of the cross product **QK** of the query and attention layers is passed through another single linear layer with GELU [44] activation function. This layer assures that we have one unified hidden representation for both the queries and the

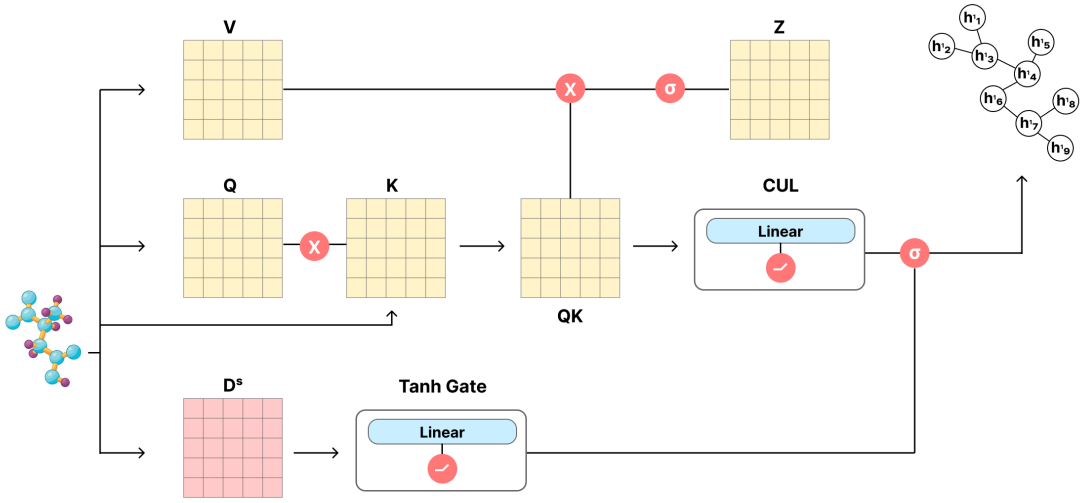

Figure 2-4: Coordinates Update Layer (CUL). We apply a softmax and multiplication for **QK** and **V** to achieve the output hidden state **Z**. In parallel, **QK** is passing through another MLP which is softmaxed and multiplied by the distance matrix to calculate a new coordination set. The network has dual output: hidden state for the next attention layers and an updated coordinate set for downstream task.

keys. This layer will is multiplied by the **V** learned representation to achieve the attention mechanism described in 2.4. In addition to this hidden representation we also calculate the gated relative coordinate differences, this is in order to return as output a set of coordinates in addition to the hidden state.

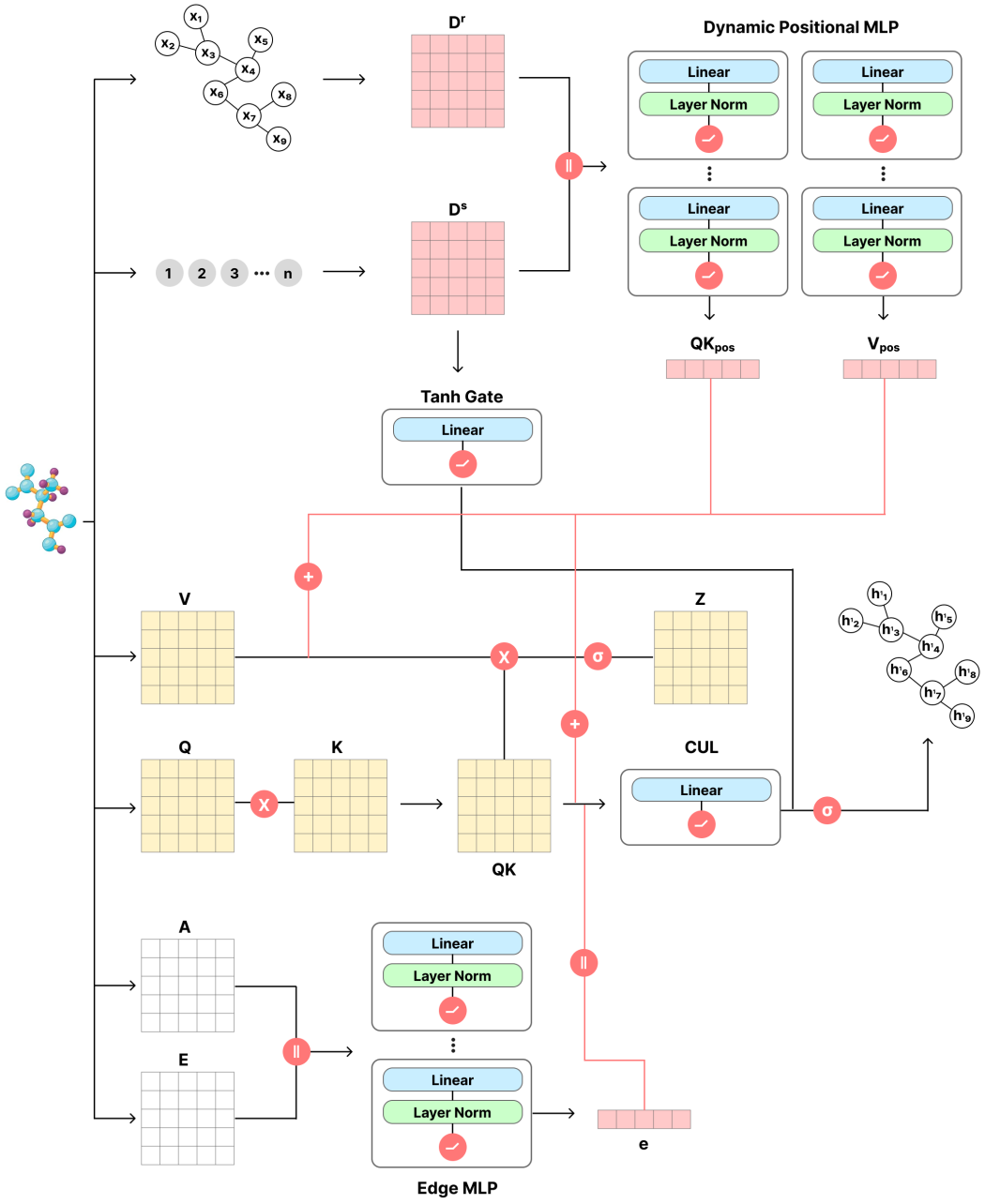

Figure 2-5: The full Equivariant Attention Layer. The architecture incorporates both positional embeddings and self-attention learned weights. In addition, the adjacency matrix **A** and an edge embedding **E** are featured and are concatenated to the **QV** attention representations. The layer has 2 outputs: coordinates h_i and a feature embedding matrix **Z** which are propagated to the next layer.

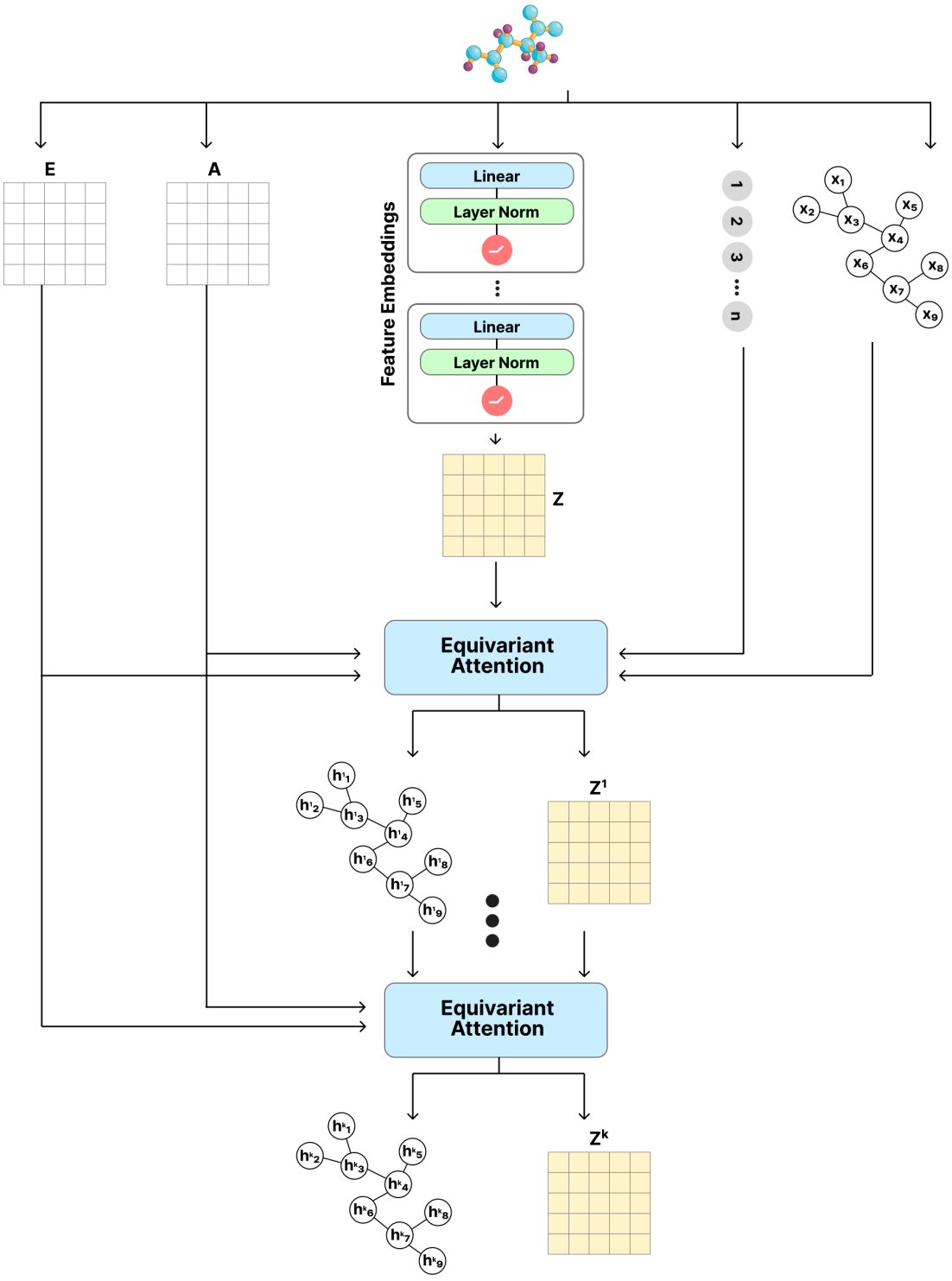

Figure 2-6: Multi Layer Equivariant Attention. This diagram describes how information propagates between attention layers. Note that the edge embeddings and adjacency matrix are constant and do not get interpolated.

Chapter 3

Protein Generation with Diffusion

3.1 Denoising Diffusion Probabilistic Models

Unsupervised generation of data instances from an unknown distribution has been a vast research field in Machine Learning. In this type of tasks the goal is not to find some correlation $X \to Y$ but instead to find an approximation for $P(X)$ and sample instances $\hat{x} \sim P(X)$ which resemble the input data. While not too far in the past highest-scoring models were flavors of Generative Adversarial Networks (GANs) [45], it appears that nowadays Denoising Diffusion Probabilistic Models (DDPMs) [46] constitute the predominant methodology. Diffusion models provide a powerful tool for modeling complex phenomena that are governed by stochastic processes. DDPMs leverage a diffusion process to generate noise samples that can be transformed into data samples using neural networks.

In essence, the DDPM operates over two processes - the *forward process* is fixed so that it gradually adds Gaussian noise to the data:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{3.1}$$

Here $t \in \{1, ..., T\}$ is a timestep and β_t is a variance schedule that can be set to a constant, some function of t, or to be learned by the model. The entire process for all timesteps is defined as the Markov Chain:

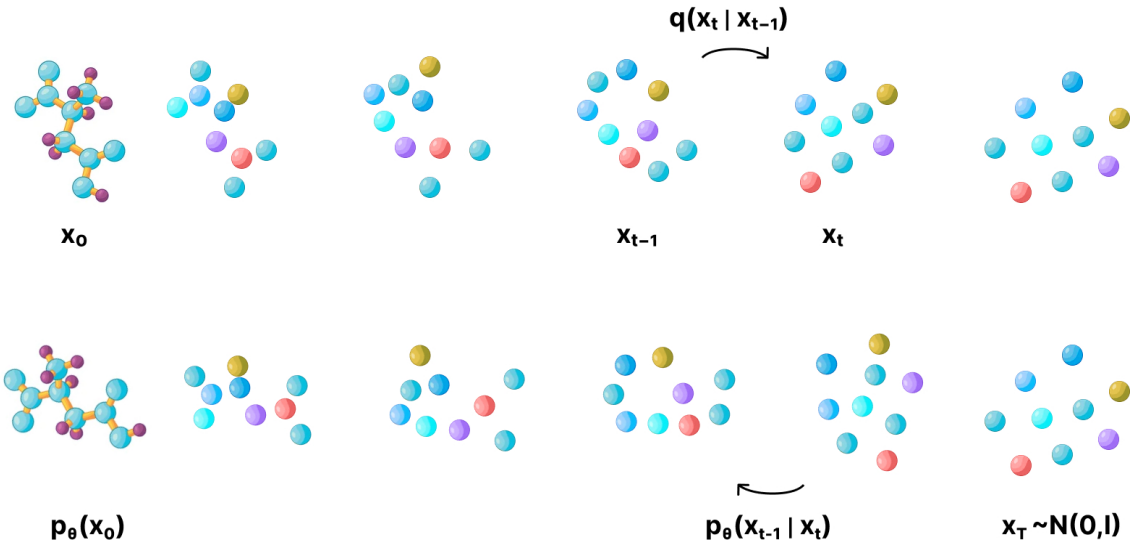

Figure 3-1: Diffusion forward process begins with $\mathbf{x}_0$ as a protein from our dataset, then in step $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ coordinates are perturbed. Then in the backward process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ the model tries to restore the denoised sample.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.2}$$

Calculating $q(\mathbf{x}_t|\mathbf{x}_0)$ can be rewritten using the notation $\alpha_t = 1 - \beta_t$ and the commutative product $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ with the following formula:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\bar{\alpha}_t}\mathbf{x}_{t-1}, (1-\bar{\alpha}_t)\mathbf{I}) \tag{3.3}$$

Now that a fixed forward diffusion process is in place, and when $T \to \infty$ we have a mechanism that gradually turns datum $\mathbf{x}_0 \sim P(X)$ into $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. Denoising the seemingly random noise back to such that comes from the original distribution is the learned *backward process* $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. In the backward process we learn mean μ_θ and variance $\mathbf{\Sigma}_\theta$ such that:

$$\begin{aligned} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &:= \mathcal{N}(\mathbf{x}_t; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)) \\ p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \end{aligned} \tag{3.4}$$

3.1.1 Learning Routine

Recall that ultimately our goal is to approximate $P(X)$ via $p_\theta(\mathbf{x}_0)$, our negative log-likelihood loss can be lower-bound:

$$\mathbb{E}(-\log p_\theta(\mathbf{x}_0)) \leq \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_0) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \tag{3.5}$$

Using Sohl-Dickstein [47] this can be further interpreted in Kullback-Leibler (KL) divergence notation (See Appendix for derivation):

$$\mathbb{E}_q \left[D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \| p(\mathbf{x}_T)) + \sum_{t>1} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]$$

In practice, we use a simpler loss function which proves to be roughly equivalent. Sampling $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, now $\mathbf{x}_T(\mathbf{x}_0, t) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ and we try to find a function $\epsilon_\theta(\mathbf{x}, t)$ that approximates ϵ as close as possible to the generated noise. So the simplified version of the loss is:

$$L(\theta) := \mathbb{E}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2 \right] \tag{3.6}$$

Concretely, training is performed With a simple gradient descent algorithm which optimizes for θ, for example, with a neural network:

Algorithm 1 Train diffusion process

Require: $\theta, \lambda \geq 0$

1: **repeat**
2: $\quad \mathbf{x_0} \sim P(X)$
3: $\quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$
4: $\quad \hat{\epsilon} := \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)$
5: $\quad \theta \leftarrow \theta + \lambda \nabla_\theta \|\epsilon - \hat{\epsilon}\|^2$
6: **until** converged

3.1.2 Unconditional Sampling

The possession of a function ϵ_θ that has been sufficiently trained indicates our capability to carry out the process of sampling. Unconditional sampling refers to simply generating data instances from the similar distribution of our training data without any further constraints. According to Ho et al. [46] we choose:

$$\begin{aligned} \Sigma_\theta(\mathbf{x}_t, t) &:= \sigma_t^2 \mathbf{I} = \beta_t \\ \mu_\theta(\mathbf{x}_t, t) &:= \tfrac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \tfrac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) \end{aligned} \tag{3.7}$$

Applying the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ using algorithm 2 results in denoising $\mathbf{x}_T$ into a dataset-like instance.

Algorithm 2 Sample from diffusion model

Require: ϵ_θ

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = \{T, ..., 1\}$ **do**
  if $t > 1$ **then**
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
  else
    $z = 0$
  end if
  $\mathbf{x}_{t-1} \leftarrow \tfrac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \tfrac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \beta_t \mathbf{z}$
end for
return $\mathbf{x}_0$

3.1.3 Conditional Sampling

Achieving many downstream tasks using sampling may involve some components which remain constant throughout the process. In our case, designing proteins that bind to specific target can be formulated in the context of conditional sampling. Instead of arbitrarily producing samples from the learned distribution we sought to provide additional context. This is done by introducing to the diffusion process another input $\mathbf{u} \in \mathbb{R}^{m \times 3}$ and denote $[\cdot, \cdot]$ the matrix row-wise concatenation operation, then our new input is $[\mathbf{x}, \mathbf{u}] \in \mathbb{R}^{(m+n) \times 3}$.

3.2 Protein Diffusion with E(n)-Transformer

The utilized architecture from 2.5 is employed to compute the Gaussian diffusion noise in the current study. It is important to note that the approximation of the backward diffusion noise $\hat{\epsilon}$ is sufficient, and that it shares the same dimensional characteristics as our input $\mathbf{x} \in \mathbb{R}^{n\times 3}$. Nevertheless, there are still significant components missing from the architecture that are crucial for proper diffusion adoption.

3.2.1 Sinusoidal Time Embeddings

Backward diffusion noise at timestep t is calculated from noisy coordinates, namely: $\hat{\mathbf{x}}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$. This means that our network is required to be aware of the time at which the noise was added to the input. For this purpose, we introduce another layer which takes as input the scalar t and transforms it into some higher dimensional vector. The Sinusoidal embedding function is defined as follows:

$$\xi(t)^{(i)} := \left\{ \begin{array}{ll} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k+1 \end{array} \right\}$$

Where:

$$\omega_k = \frac{1}{10000^{2k/d}}$$

This alternating cosine and sine representation results in vectors with convenient characteristics to represent some notion of order (see fig 3-2).

3.2.2 Sequence Embeddings

In our conditional diffusion model, we have devised a method to incorporate the DNA sequence as input using one-hot encoding vectors. One-hot encoding allows us to represent each nucleotide in the DNA sequence as a binary vector. To integrate the DNA sequential information effectively, we propose attaching it to the initial hidden representation denoted as $\mathbf{Z}$. The hidden representation $\mathbf{Z}$ serves as a starting point for the subsequent modeling and generation process. Typically, the feature embedding

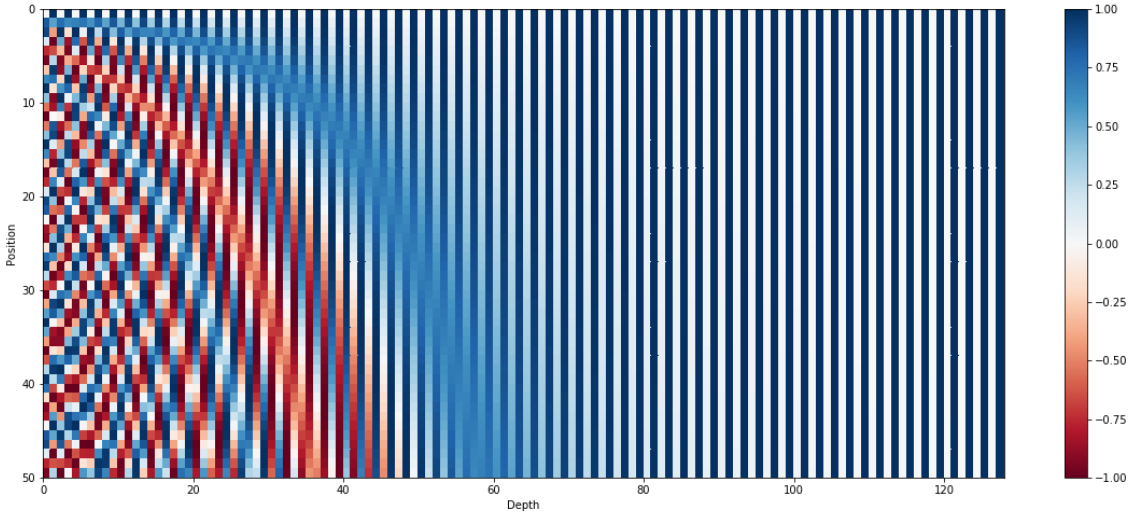

Figure 3-2: Sinusoidal embeddings for dimension $d = 128$. Here the maximum input length is 50. Each row in this image represents the corresponding time embeddings $\xi(t)$

process begins with a constant vector, which acts as a baseline or initial state for the model. However, in our case, to incorporate the DNA sequence, we suggest replacing the constant vector with the corresponding one-hot encoded sequence vector.

By substituting the constant vector with the one-hot encoded sequence vector in the feature embedding step, we ensure that the DNA sequence information is present right from the beginning of the modeling process. This modification allows the model to leverage the specific nucleotide information encoded in the DNA sequence, capturing its unique patterns and correlations. Consequently, this enriched initial hidden representation facilitates more accurate and context-aware generation of the desired outputs based on the DNA input.

3.3 DNA-Conditional Protein Sampling

Adopting the conditional diffusion sampling method we now propose a novel way to generate DNA-specific binders. Let $\delta \in \{A, C, G, T\}^m$ be some DNA sequence comprised of the nucleotides Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). We construct a matrix $\Delta \in \mathbb{R}^{2m \times 3}$ to represent a double-stranded coordinates

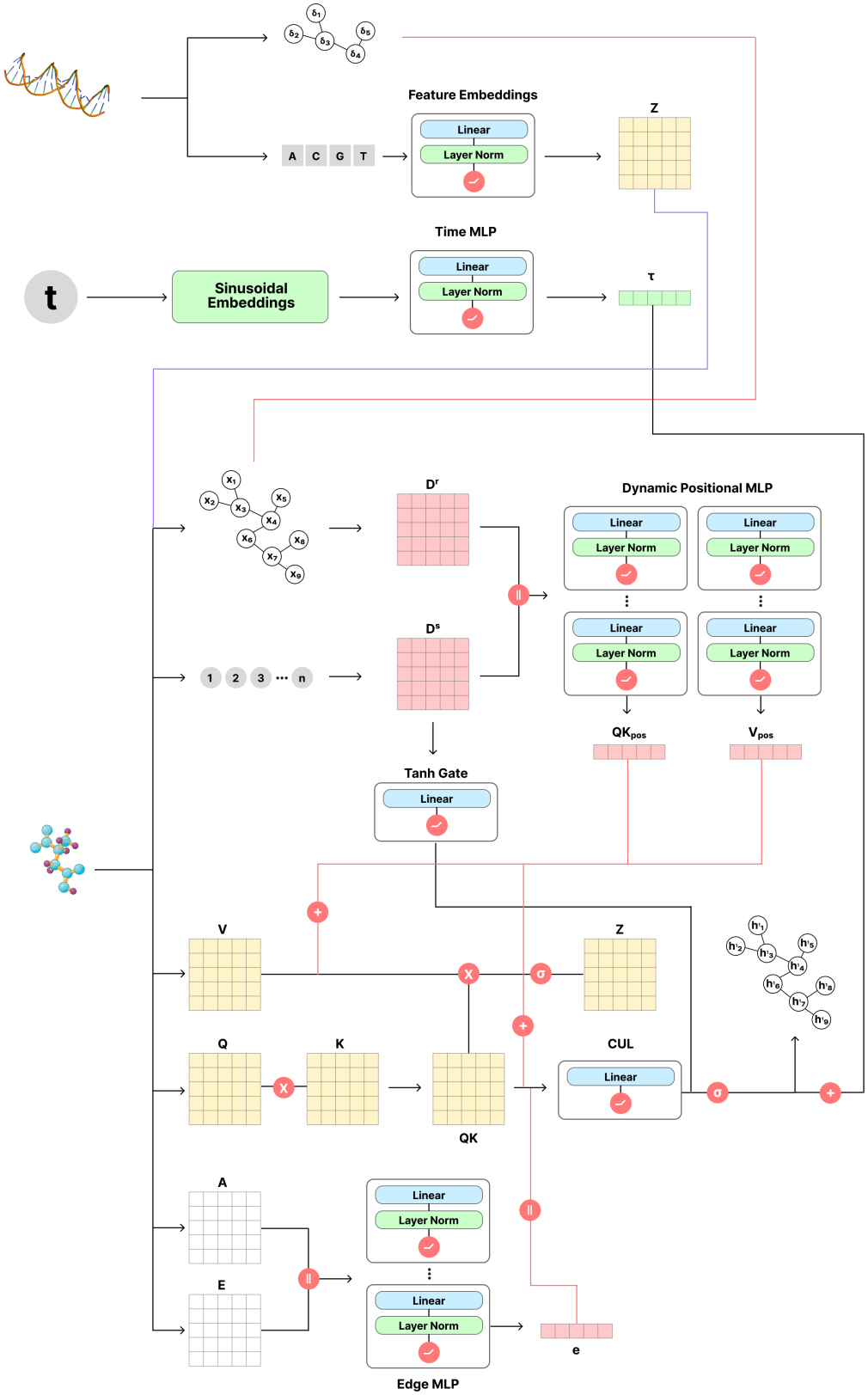

Figure 3-3: The full E(n)-Transformer architecture adopted to diffusion. Time embeddings and DNA conditions are added as additional inputs.

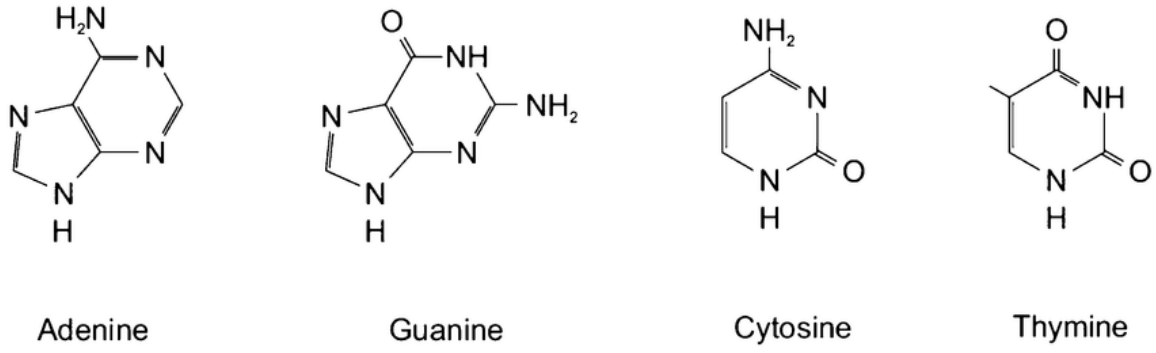

Figure 3-4: All four nucleotides structural formulas.

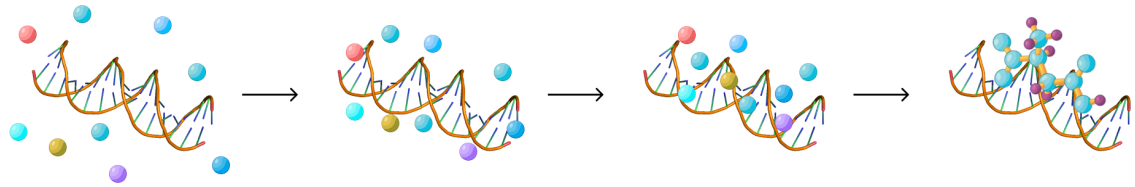

Figure 3-5: DNA-condition diffusion sampling. The left hand side is the initial step x_0 where the DNA structure is fixed in place. Then throughout the diffusion process atom coordinates are denoised into DNA binders.

set that corresponds to the atoms in the sequence δ. Nucleotides consist of multiple Carbon, Oxygen and Hydrogen atoms (see fig 3-4), however, to reduce computational complexity we compute a central atom position for each nucleotide. This central position (referred to as "centroid") serves an anchor for the entire nucleotide in future processing.

Recall that with our diffusion modelling we do not explicitly approximate the recovered coordinates, but rather we train the network to learn the noise ϵ. In this sense, conditioning as it is shown in figure 3-5 is not trivial. We suggest that our learning process is done in a two-fold fashion: First, the network initiates the features embedding vector with the one-hot encoded sequence. Second, since we do not noise the DNA structural representation, the noise at these coordinates is zero. This means that

Chapter 4

DNA-binding Protein Design

4.1 Protein-DNA Interaction

A definition for nucleotide-amino acid residue interaction is not trivial from the computational perspective. While we know that binds are often formed by hydrogen bonds, these are usually missing from X-ray crystallized structures and are considered expensive to compute. We propose here a simpler method for determining which amino acids in a protein make contact with corresponding nucleotides by measuring the Euclidean proximity between their atoms. We experiment with two metrics: d_{min} and d_{center}. The first method measures the distance between the two nearest atoms in each molecule and the second compares the distance between the two molecule centroids (See fig 4-1). Weaker interactions are existent in the form of Van der Waals bonds and are commonly classified into two systems of forces: London dispersion forces and Dipole-dipole interactions.

London dispersion forces occur between all atoms and molecules, regardless of their polarity. They arise from temporary fluctuations in electron distribution, leading to the creation of temporary dipoles. These temporary dipoles induce additional temporary dipoles in neighboring atoms or molecules, resulting in an attractive force. London dispersion forces generally increase with the size and shape of the molecules involved.

Dipole-dipole interactions: These forces occur between polar molecules. Polar

molecules have a permanent dipole moment due to the unequal sharing of electrons between atoms. The positive end of one molecule is attracted to the negative end of a neighboring molecule, resulting in dipole-dipole interactions. The strength of dipole-dipole interactions depends on the magnitude of the dipole moment and the proximity of the molecules.

4.2 Hallucination

As a benchmark to protein generation, we describe here a naive attempt to generate de novo DNA binders using the protein hallucination approach [17]. This approach relies on an existing sequence-to-structure folding algorithm such as AlphaFold. We initiate with a random amino acid sequence, then perform a recursive Markov Chain Monte Carlo (MCMC) sampling to predict a distance map. We optimize our MCMC by mutating the random sequence so that its resulting distance map is less and less "blurry".

To conduct an experiment with hallucination we first detect the amino acids which interact with most neighboring nucleotides in the sequence (see fig 4-2). We assign k to be the number of nearest amino acids to each nucleotide and aggregate across the entire sequence. Finally, we pick regions with more neighboring amino acids and preserve them, then mask the rest of the sequence and let the protein hallucination algorithm "inpaint" the remaining of the sequence. The main problem with this approach to solve our problem is that while it produces valid proteins, it does not take into account any DNA constraints, so the hallucinated proteins do not comply with amino-acid nucleotide bonds.

4.3 Experiment Validation

4.3.1 Distance-map Loss

To validate that our network indeed learns to generate proteins, it is not sufficient to observe the noise prediction loss. Instead, while training we perform a full sampling

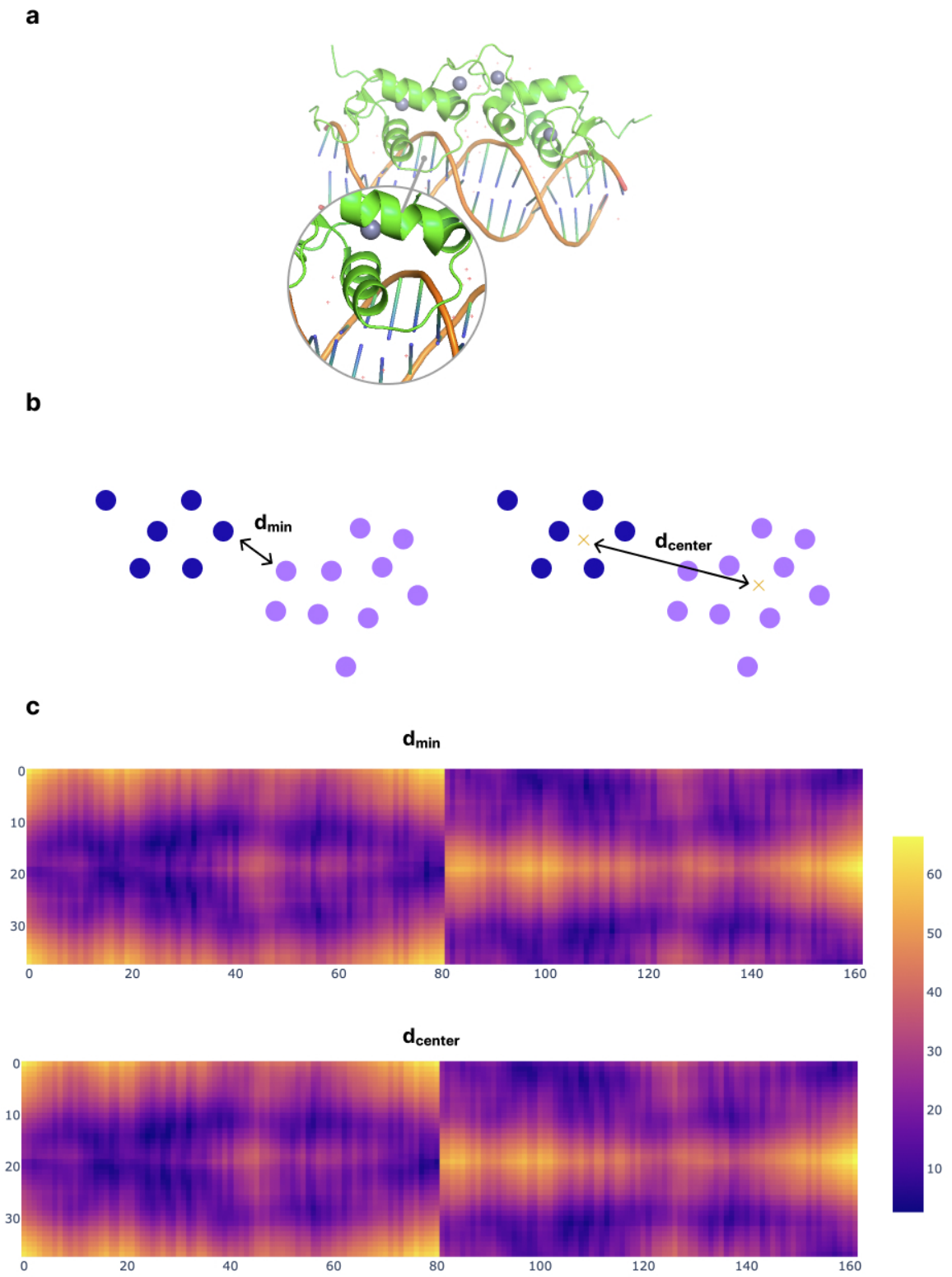

Figure 4-1: Contact between amino acids and nucleotides. a) Zoomed-in interaction within a sample X-ray crystallized structure. b) The definition of the two metrics experimented - d_{min} selects the two nearest atoms and d_{center} computes the point cloud centroids. c) Two typical value spectroscopes for d_{min} and d_{center}, lighter colors indicate closer Euclidean proximity.

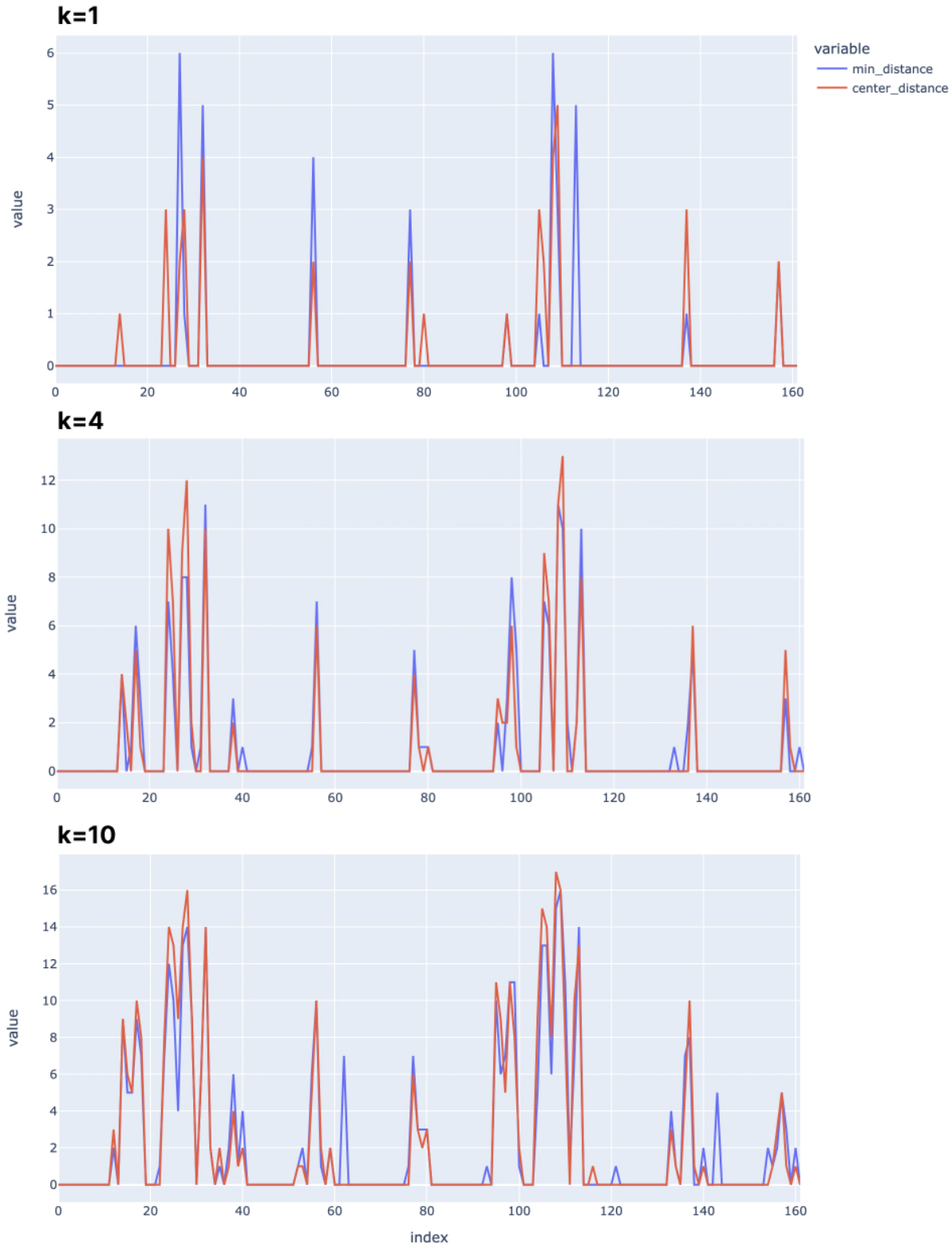

Figure 4-2: Nearest amino acids by position. For each nucleotide we pick the k nearest amino acids. The Y-axis value sums the number of occurrences for each indexed amino acid within this metric. The graphs show some amino acids are closer to multiple nucleotides and thus get a higher score.

procedure by running Algorithm 2 for timesteps $1, 2, ..., T$. We now strip the generated backbone complex to it's amino acid part only and compute a distance map similar to

4.3.2 TM-score

Template modeling score is another method of measuring similarity between two protein structures. Commonly, a TM-score higher than 0.5 indicates that the two proteins fold to the same structure. In our experiments we sought to maximize the score not only to indicate same fold, but rather to perfectly fit the structure. The TM-score is defined by:

$$TM(x, \hat{x}) = \max \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + \left(\frac{x_i - \hat{x}_i}{1.24 \sqrt[3]{n-15} - 1.8} \right)} \right)$$

4.3.3 Quantile Loss

Diffusion noise prediction loss can be misleading because on average it is hard to predict a random Markov process. A useful observation is to divide the noise prediction loss based on the timestep t at which the forward diffusion process was generated from. In our experiments, indeed it shows clearly that in timesteps the loss is indeed unstable, as it is more difficult for the network to distinguish noisy coordinates within similar structures. It is much easier for the network to push towards a zero-mean gaussian from noise that is seemingly completely random. In our experiments we used 10 quantiles $q_1, ..., q_10$, and plotted the loss of timestep t to separate graphs $\lfloor (t/T) * 10 \rfloor$.

4.4 Results

The work presented in this thesis has yielded some initial promising results in generating novel genome engineering proteins. While it is important to note that our trained model is not fully optimized or fine-tuned, it has demonstrated the capabil-

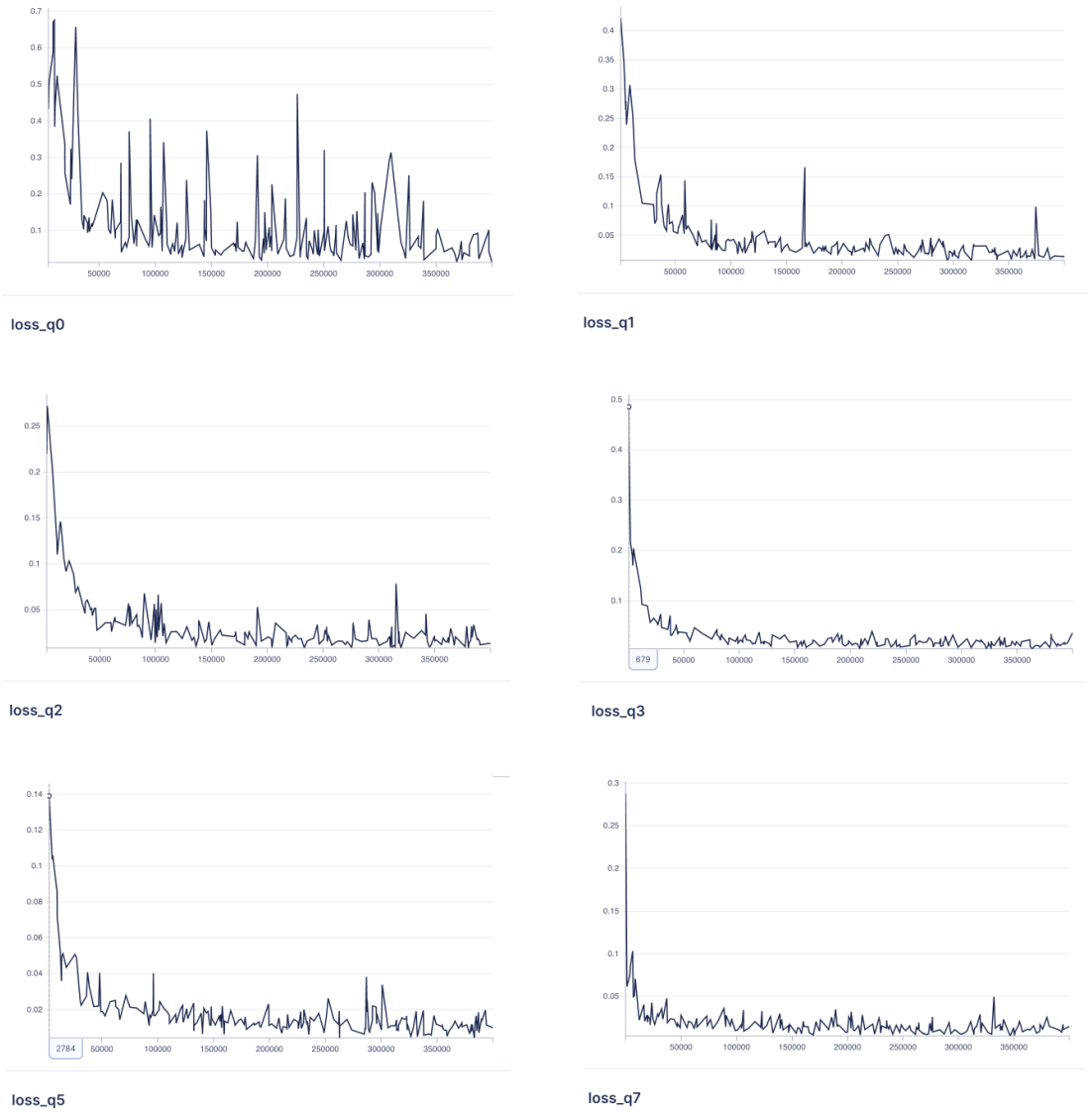

Figure 4-3: Loss quantiles. From top to bottom and left to right are earlier to later stage loss quantiles. Later quantiles losses are substantially lower than earlier ones.

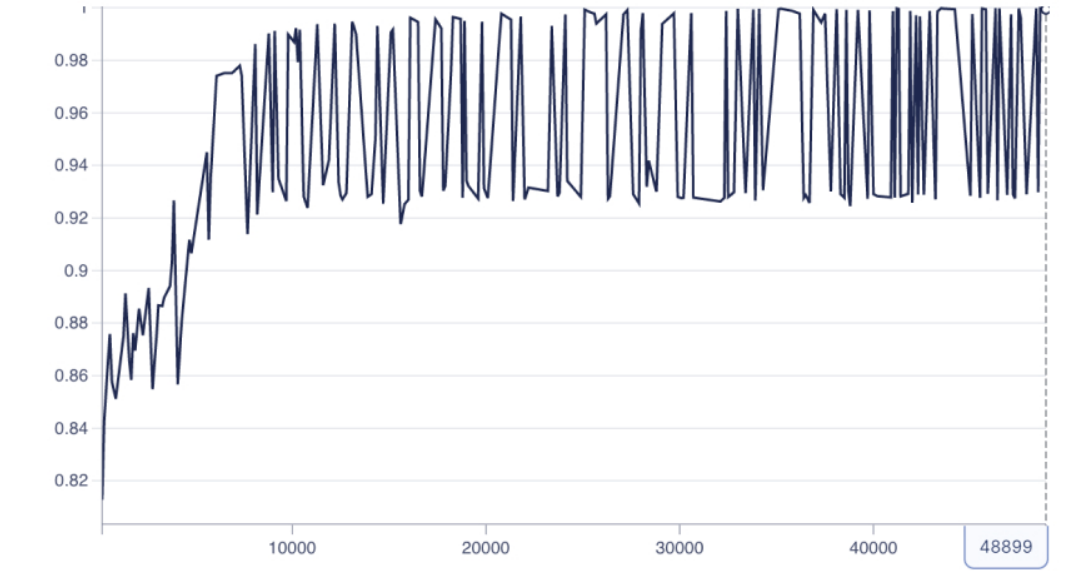

Figure 4-4: TM-score. scores > 0.5 indicate same protein folds. Here we show that the network learns to produce not only same-fold proteins but rather it fits perfectly to a TM-score close to 1.

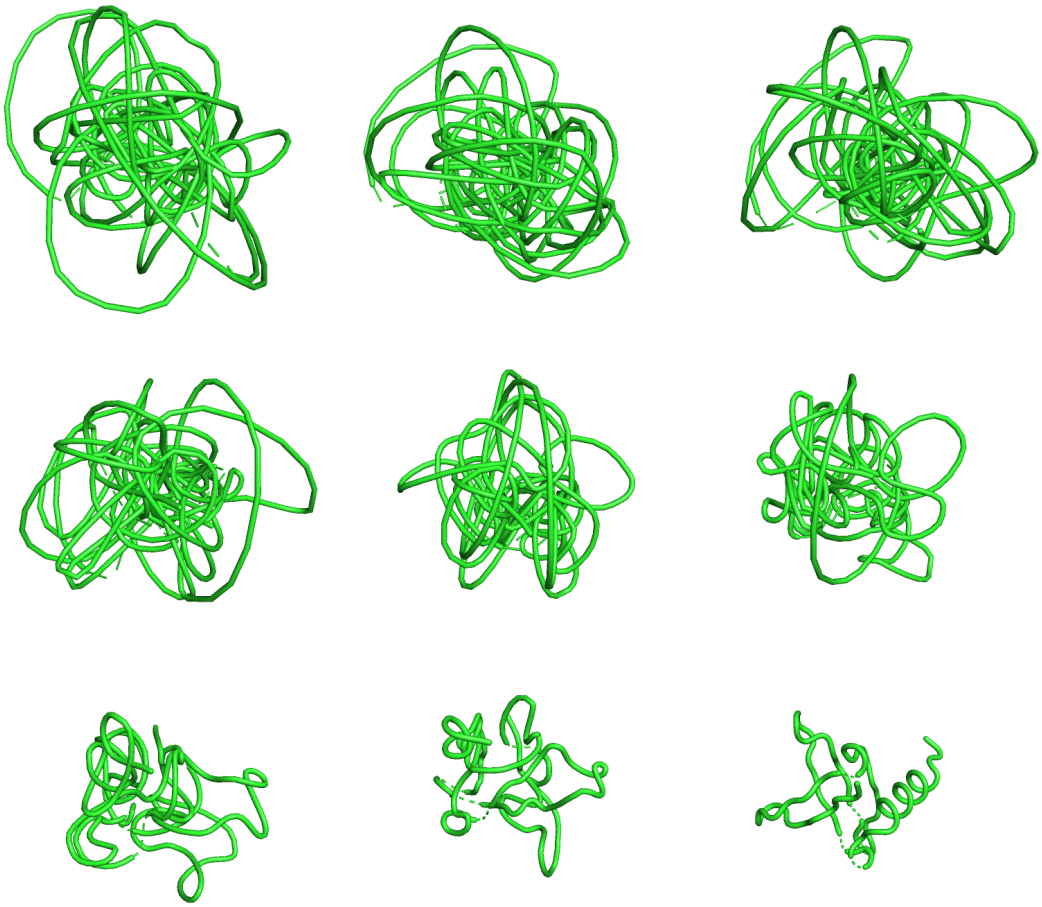

Figure 4-5: 1ZBI [48] diffusion over 250 timesteps. From left: we initiate a multivariate zero-mean normal noise. The forward diffusion noise predictions iteratively recover the protein.

ity to condition the generation process on DNA atoms and produce DNA protein binders with reasonable properties. Through rigorous evaluation and analysis, we have observed encouraging indications of the potential of deep learning techniques in the context of genome engineering. However, further refinement and optimization of the model are required to achieve more robust and reliable results. These preliminary findings highlight the early progress made in exploring the generation of DNA protein binders, setting the stage for future research and development in this area. In this sections figures we demonstrate some of the backbones which emerge from running the diffusion process for T timesteps and with some specific DNA condition.

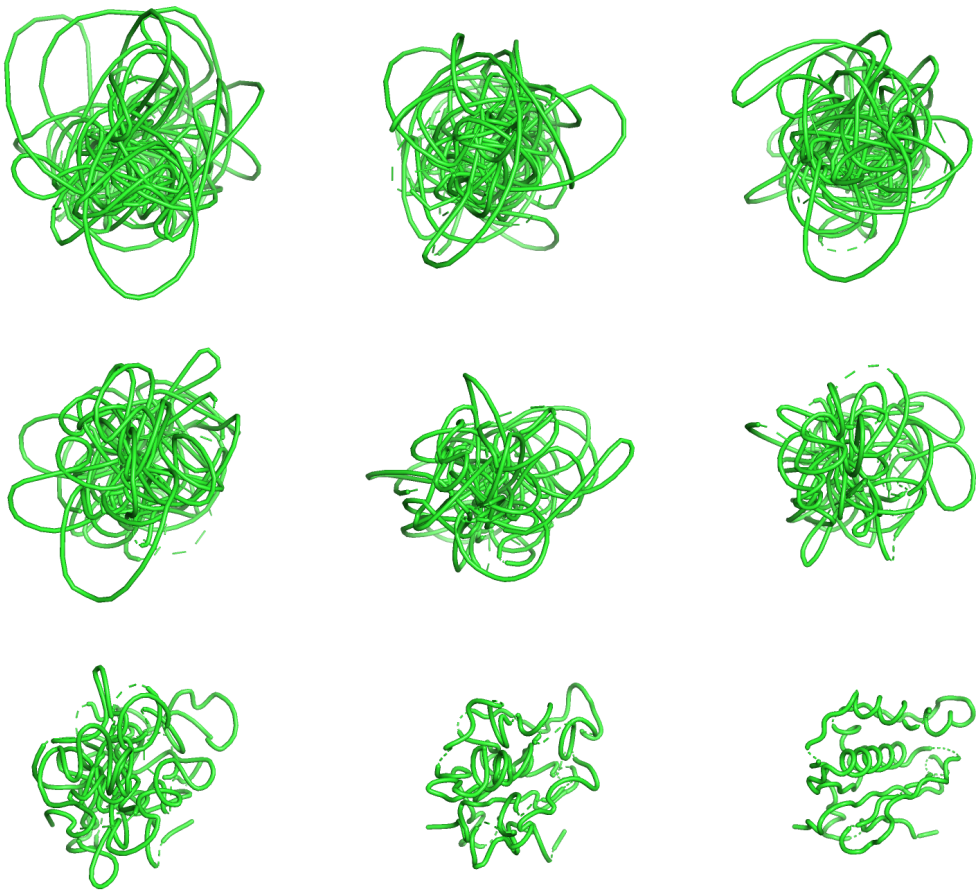

Figure 4-6: 1AZP [49] diffusion over 250 timesteps. From left: we initiate a multivariate zero-mean normal noise. The forward diffusion noise predictions iteratively recover the protein.

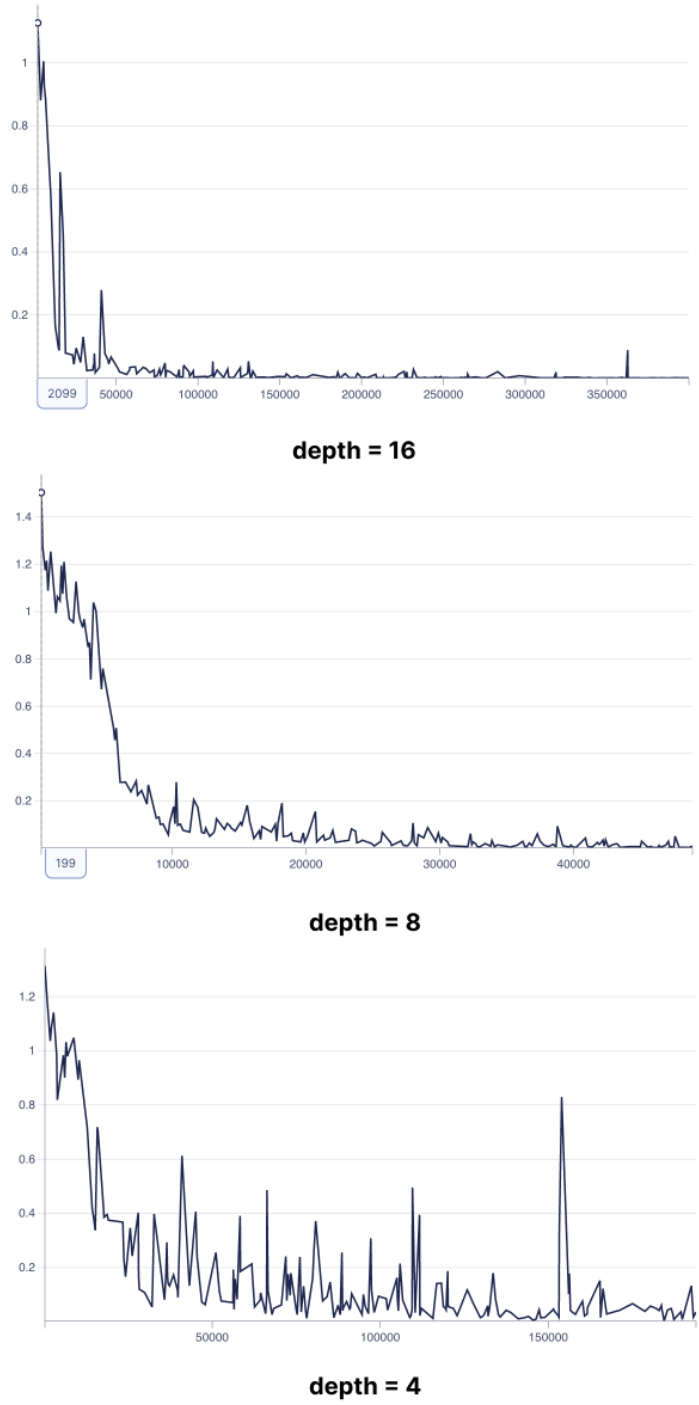

Figure 4-7: Model depth analysis. We experimented with various model depth (number of Equivariant Attention blocks). Deeper network converge faster but consume more memory and GPU computation time.

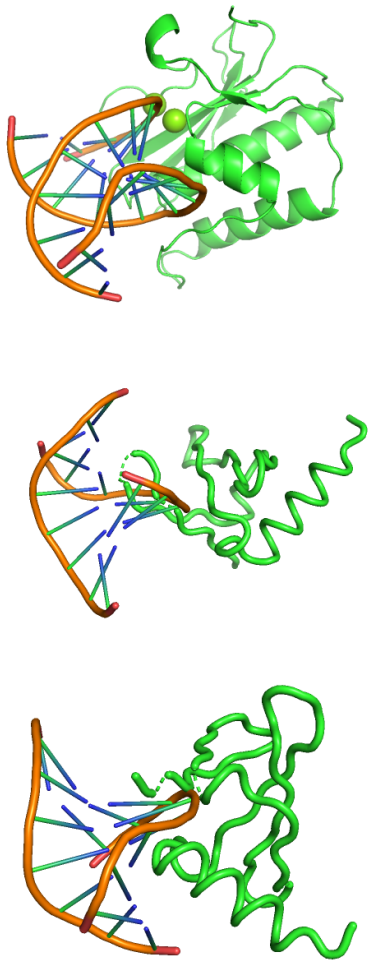

Figure 4-8: DNA-Conditional generation for sequence GCGATCGC. From top to bottom: the reference protein complex, the reference stripped to backbone atoms only and our predicted protein.

4.5 Programmable DNA Binders

The model proposed in this thesis offers the potential for extension towards the generation of programmable DNA binders. By replacing the initial features layer with a one-hot encoded vector representing the desired DNA sequence, the model can be adapted to specifically generate protein binders that target and interact with the given DNA sequence. This modification allows for precise control and customization of the generated binders, enabling the design of DNA-protein complexes. Since DNA structures is more predictable than protein folded structure, we may leave the structural DNA representation as it is in a reference complex. This idea is to be fully explored when we train the network on a larger dataset.

Chapter 5

Conclusions

5.1 Thesis Contribution

To the best of my knowledge, this thesis exhibits the first diffusion-based model to integrate nucleic information, marking a significant advancement in the field. While diffusion models for protein backbone generation have been explored in previous studies, none of them have proposed incorporating a prior condition in the form of DNA sequence and structure. This innovation opens up new possibilities for understanding the intricate relationship between DNA and protein structures. By leveraging diffusion models in combination with GNNs, we not only provide a novel framework for generating protein backbones conditioned on DNA information, but also uncover the potential of this approach for accurately capturing the intricate interplay between these two biomolecules. Our findings shed light on the feasibility of employing diffusion models as a powerful tool in the field of structural biology, offering valuable insights into the generation and exploration of complex macromolecular structures.

This thesis highlights the superiority of employing GNNs for encoding, and diffusion probabilistic models for sampling, compared to previously investigated methods like Hallucination and sampling from language models. The research showcases the distinct advantages of this novel approach, emphasizing its effectiveness in capturing complex relationships and generating high-quality samples. By leveraging GNNs and diffusion models, we surpass the limitations of existing approaches and provide a

robust framework for encoding and sampling that significantly improves upon prior methodologies. Our protein representation proves to preserve E(n) equivariance, and our noise prediction demonstrates high quality backbone generation from this representation.

We present compelling evidence supporting the efficacy of the recently introduced E(n)-Transformer architecture for protein design. The findings suggest that utilizing Transformer architectures in protein design represents a notable advancement in capturing the geometric latent representation of proteins. The E(n)-Transformer offers a promising alternative to the widely employed SE(3)-Transformer, offering a more compact and efficient framework.

The field of diffusion probabilistic models continues to be a promising and relatively unexplored area of research. Our work contributes to the expansion of this field by uncovering yet another valuable application. By exposing practical aspects and insights derived from our investigations, we not only advance the understanding of diffusion probabilistic models but also provide a framework that can be leveraged in various unsupervised sampling tasks beyond the immediate scope of protein structure generation. These practical aspects may include strategies for enhancing exploration, handling uncertainty, optimizing sampling efficiency, or adapting the models to different types of data. By identifying and sharing these insights, we aim to foster cross-disciplinary collaborations and inspire further research in utilizing diffusion probabilistic models for diverse unsupervised sampling tasks, opening up new avenues for scientific exploration and innovation.

5.2 Ethics

The ethics surrounding synthetic genome engineering proteins is a topic of significant concern and debate within the scientific community and broader society. At the core of this discussion lies the potential for scientists to manipulate and engineer genetic material, including the creation of synthetic proteins with specific functions. While synthetic genome engineering proteins hold immense promise for various applications,

such as disease treatment, biofuel production, and environmental remediation, their ethical implications cannot be ignored. One of the primary concerns is the potential for unintended consequences and unforeseen risks associated with releasing these proteins into the environment or introducing them into organisms. The long-term effects and ecological impacts of such manipulations are uncertain and require careful consideration. Additionally, questions surrounding the equitable access to synthetic genome engineering technologies and the potential for their misuse or weaponization raise important ethical considerations. The responsible development and use of synthetic genome engineering proteins necessitate robust ethical frameworks, transparent communication, and rigorous oversight to balance the potential benefits with the risks and ensure the well-being of both humans and the environment.

5.3 Future Discussion

5.3.1 Large Scale Training

As a crucial next step following this thesis, a significantly larger-scale training encompassing all available Protein-DNA complexes is underway. This expanded training aims to leverage a comprehensive dataset of diverse Protein-DNA interactions to further enhance the model's performance and broaden its applicability. By incorporating a wider range of Protein-DNA complexes, the model can effectively learn intricate patterns, dependencies, and preferences specific to various DNA sequences and their corresponding protein binders. The ongoing training procedure, which is anticipated to be completed within a few weeks, holds great promise for advancing the accuracy, versatility, and generalizability of the model. The comprehensive training will empower the model to generate programmable DNA binders with enhanced precision, specificity, and affinity, thereby fostering breakthroughs in genome engineering, synthetic biology, and related fields. The expected outcomes of this extended training will provide valuable insights and resources for the scientific community, unlocking new possibilities in the design and engineering of functional Protein-DNA interac-

tions, and catalyzing advancements in the fields of genome engineering, synthetic biology, and beyond.

5.3.2 In-vitro Experiments

In the realm of protein-DNA interactions, the most accurate and reliable assessment of affinity necessitates conducting wet lab experiments. While computational methods can provide valuable insights and predictions, they often rely on simplified models and approximations, limiting their ability to capture the intricacies of the complex protein-DNA binding process. Wet lab experiments, on the other hand, offer a direct and comprehensive approach to measure the actual binding affinity between proteins and DNA molecules. These experiments employ techniques such as electrophoretic mobility shift assays (EMSAs), isothermal titration calorimetry (ITC), and surface plasmon resonance (SPR), which allow for precise characterization of the binding kinetics, thermodynamics, and specificity of protein-DNA interactions. By combining computational predictions with rigorous wet lab experiments, a more holistic and accurate understanding of protein-DNA affinity can be achieved, enabling the development of effective strategies for genome engineering, gene regulation, and other applications reliant on precise protein-DNA interactions.

5.4 Published Artifacts

The majority of the research presented in this paper stems from a thorough examination of a meticulously crafted codebase, which affords the necessary versatility for conducting comprehensive experiments involving all aspects of the project. Notably, all the code developed during this study has been made openly accessible to the public, adhering to the MIT license.

5.4.1 Open-source Code

We publish the full code including both the configurable equivariant GNN and the diffusion sampling procedures. Some of the components include protein preprocessing, validation metrics, experiment management and a GPU-scalable training script. The codebase is available at `https://github.com/molecularmachines/genomator`

5.4.2 Moleculib

In parallel to this project we developed a platform to represent molecules in a format that is suitable and optimal for deep learning. Moleculib is extended to include Protein-DNA complexes with proper masking to allow post-processing separation. The Python library is available at: `https://github.com/molecularmachines/moleculib`

Bibliography

[1] Gregory A. Newby et al. "Base editing of haematopoietic stem cells rescues sickle cell disease in mice". eng. In: *Nature* 595.7866 (July 2021), pp. 295–302. ISSN: 1476-4687.

[2] P. Renaudier. "[Sickle cell pathophysiology]". fre. In: *Transfusion Clinique Et Biologique: Journal De La Societe Francaise De Transfusion Sanguine* 21.4-5 (Nov. 2014), pp. 178–181. ISSN: 1953-8022.

[3] So Hyun Park and Gang Bao. "CRISPR/Cas9 gene editing for curing sickle cell disease". In: *Transfusion and apheresis science : official journal of the World Apheresis Association : official journal of the European Society for Haemapheresis* 60.1 (Feb. 2021), p. 103060. ISSN: 1473-0502.

[4] David A. Wah et al. "Structure of FokI has implications for DNA cleavage". In: *Proceedings of the National Academy of Sciences of the United States of America* 95.18 (Sept. 1998), pp. 10564–10569. ISSN: 0027-8424.

[5] Scot A. Wolfe, Lena Nekludova, and Carl O. Pabo. "DNA Recognition by Cys2His2 Zinc Finger Proteins". In: *Annual Review of Biophysics and Biomolecular Structure* 29.1 (2000). _eprint: https://doi.org/10.1146/annurev.biophys.29.1.183, pp. 183–212.

[6] X. Bao and S. P. Palecek. "Chapter 1 - Genetic Engineering in Stem Cell Biomanufacturing". en. In: *Stem Cell Manufacturing*. Ed. by Joaquim M. S. Cabral et al. Boston: Elsevier, Jan. 2016, pp. 1–25. ISBN: 978-0-444-63265-4.

[7] Anuradha Bhardwaj and Vikrant Nain. "TALENs—an indispensable tool in the era of CRISPR: a mini review". In: *Journal of Genetic Engineering and Biotechnology* 19.1 (Aug. 2021), p. 125. ISSN: 2090-5920.

[8] Rajat M. Gupta and Kiran Musunuru. "Expanding the genetic editing tool kit: ZFNs, TALENs, and CRISPR-Cas9". In: *The Journal of Clinical Investigation* 124.10 (Oct. 2014), pp. 4154–4161. ISSN: 0021-9738.

[9] Juan P. Fernandez et al. "Optimized CRISPR-Cpf1 system for genome editing in zebrafish". en. In: *Methods*. Gene Editing, Genomics, and In Vivo Imaging in Zebrafish 150 (Nov. 2018), pp. 11–18. ISSN: 1046-2023.

[10] Mohammed Fatih Rasul et al. "Strategies to overcome the main challenges of the use of CRISPR/Cas9 as a replacement for cancer therapy". In: *Molecular Cancer* 21.1 (Mar. 2022), p. 64. ISSN: 1476-4598.

[11] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". en. In: *Nature* 596.7873 (Aug. 2021). Number: 7873 Publisher: Nature Publishing Group, pp. 583–589. ISSN: 1476-4687.

[12] Andriy Kryshtafovych et al. "Critical assessment of methods of protein structure prediction (CASP)—Round XIII". en. In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.2582 pp. 1011–1020. ISSN: 1097-0134.

[13] Joana Pereira et al. "High-accuracy protein structure prediction in CASP14". en. In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26171, pp. 1687–1699. ISSN: 1097-0134.

[14] *Enabling high-accuracy protein structure prediction at the proteome scale*. en.

[15] Raghav Shroff et al. "Discovery of Novel Gain-of-Function Mutations Guided by Structure-Based Deep Learning". In: *ACS Synthetic Biology* 9.11 (Nov. 2020). Publisher: American Chemical Society, pp. 2927–2935.

[16] Jooyoung Lee, Adam Liwo, and Harold A. Scheraga. "Energy-based de novo protein folding by conformational space annealing and an off-lattice united-residue force field: Application to the 10-55 fragment of staphylococcal protein A and to apo calbindin D9K". In: *Proceedings of the National Academy of Sciences* 96.5 (Mar. 1999). Publisher: Proceedings of the National Academy of Sciences, pp. 2025–2030.

[17] Ivan Anishchenko et al. "De novo protein design by deep network hallucination". en. In: *Nature* 600.7889 (Dec. 2021). Number: 7889 Publisher: Nature Publishing Group, pp. 547–552. ISSN: 1476-4687.

[18] J. Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (Oct. 2022). Publisher: American Association for the Advancement of Science, pp. 49–56.

[19] Gabriele Corso et al. *DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking*. arXiv:2210.01776 [physics, q-bio]. Oct. 2022.

[20] Brian L. Trippe et al. *Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem*. arXiv:2206.04119 [cs, q-bio, stat]. June 2022.

[21] Ashish Vaswani et al. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Dec. 2017.

[22] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs]. May 2019.

[23] Tom B. Brown et al. *Language Models are Few-Shot Learners*. arXiv:2005.14165 [cs]. July 2020.

[24] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv:1910.10683 [cs, stat]. July 2020.

[25] Alexander Rives et al. *Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences*. en. Pages: 622803 Section: New Results. Dec. 2020.

[26] Zeming Lin et al. *Language models of protein sequences at the scale of evolution enable accurate structure prediction*. en. Pages: 2022.07.20.500902 Section: New Results. July 2022.

[27] Maxim Zvyagin et al. *GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics*. en. preprint. Bioinformatics, Oct. 2022.

[28] Joseph L. Watson et al. *Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models*. en. Pages: 2022.12.09.519842 Section: New Results. Dec. 2022.

[29] Minkyung Baek et al. *Accurate prediction of protein structures and interactions using a 3-track network*. en. Pages: 2021.06.14.448402 Section: New Results. June 2021.

[30] Arnaud Doucet et al. "Sequential Monte Carlo Methods in Practice". In: (Jan. 2013). ISSN: 978-1-4419-2887-0.

[31] Jason Yim et al. *SE(3) diffusion model with application to protein backbone generation*. arXiv:2302.02277 [cs, q-bio, stat]. Feb. 2023.

[32] Ruidong Wu et al. *High-resolution de novo structure prediction from primary sequence*. en. Pages: 2022.07.21.500999 Section: New Results. July 2022.

[33] Robert Verkuil et al. *Language models generalize beyond natural proteins*. en. Pages: 2022.12.21.521521 Section: New Results. Dec. 2022.

[34] Michael M. Bronstein et al. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (July 2017). Conference Name: IEEE Signal Processing Magazine, pp. 18–42. ISSN: 1558-0792.

[35] Jan E. Gerken et al. *Geometric Deep Learning and Equivariant Neural Networks*. arXiv:2105.13926 [hep-th]. May 2021.

[36] Yongji Wu et al. "Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020). Number: 01, pp. 1054–1061. ISSN: 2374-3468.

[37] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. *Exploring Graph Neural Networks for Stock Market Predictions with Rolling Window Analysis*. arXiv:1909.10660 [cs, q-fin]. Nov. 2019.

[38] Peter W. Battaglia et al. *Interaction Networks for Learning about Objects, Relations and Physics*. arXiv:1612.00222 [cs]. Dec. 2016.

[39] Hanjun Dai et al. *Learning Combinatorial Optimization Algorithms over Graphs*. arXiv:1704.01665 [cs, stat]. Feb. 2018.

[40] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. arXiv:1609.02907 [cs, stat]. Feb. 2017.

[41] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. arXiv:1704.01212 [cs]. June 2017.

[42] Petar Veličković et al. *Graph Attention Networks*. arXiv:1710.10903 [cs, stat]. Feb. 2018.

[43] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. *E(n) Equivariant Graph Neural Networks*. arXiv:2102.09844 [cs, stat] version: 1. Feb. 2021.

[44] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. arXiv:1606.08415 [cs]. July 2020.

[45] Ian J. Goodfellow et al. *Generative Adversarial Networks*. arXiv:1406.2661 [cs, stat]. June 2014.

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

[47] Jascha Sohl-Dickstein et al. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. arXiv:1503.03585 [cond-mat, q-bio, stat]. Nov. 2015.

[48] *Crystal Structures of RNase H Bound to an RNA/DNA Hybrid: Substrate Specificity and Metal-Dependent Catalysis: Cell*.

[49] Howard Robinson et al. "The hyperthermophile chromosomal protein Sac7d sharply kinks DNA". en. In: *Nature* 392.6672 (Mar. 1998). Number: 6672 Publisher: Nature Publishing Group, pp. 202–205. ISSN: 1476-4687.