Degree Project in the Field of Technology Media Technology and the Main Field of Study Computer Science and Engineering

Second cycle, 30 credits

Predicting Protein-DNA Binding Affinity Using AlphaFold Embeddings & Graph Attention Networks

MARCUS WARNERFJORD

Predicting Protein-DNA Binding Affinity Using AlphaFold Embeddings & Graph Attention Networks

MARCUS WARNERFJORD

Master's Programme, Interactive Media Technology, 120 credits
Date: June 16, 2025

Supervisors: Jarmo Laaksolahti, Emil Marklund
Examiner: Joakim Gustafsson
School of Electrical Engineering and Computer Science
Swedish title: Prediktering av protein-DNA-bindningsaffinitet med AlphaFold-inbäddningar och graf-attentionsnätverk

Predicting Protein-DNA Binding Affinity Using AlphaFold Embeddings & Graph Attention Networks

MARCUS WARNERFJORD

Predicting the binding affinity between proteins and DNA is a fundamental challenge in molecular biology, crucial for understanding gene regulation and the molecular basis of diseases. Conventional methods often struggle with the complexity of these interactions and the limited availability of structural data. This thesis explores a novel deep learning framework for predicting protein-DNA binding affinity by leveraging structural embeddings generated by AlphaFold 3, a state-of-the-art protein structure prediction model. The proposed approach represents protein-DNA complexes as fully connected graphs, where amino acid and nucleotide residues are nodes and their potential interactions are edges. AlphaFold's single representation serves as node features, and the pair representation provides edge features, eliminating the need for handcrafted features. The study contains a comparative analysis of different Graph Neural Network (GNN) architectures, including Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and Graph Transformers. Models were evaluated on three distinct datasets: two high-throughput HiTS-FLIP datasets (KLF1 and LacI) and the more diverse PDBbind dataset. The results demonstrate that attention-based architectures, particularly GAT, outperform the GCN baseline on the HiTS-FLIP datasets, achieving a Pearson Correlation Coefficient (PCC) of up to 0.94 after hyperparameter optimization on the KLF1 dataset. However, this performance advantage was less pronounced on the heterogeneous PDBbind dataset. Our analysis revealed that model performance is more sensitive to node features than to edge features, suggesting that current GNN architectures may not fully utilize the rich interaction information in AlphaFold's pair embeddings. While the models show strong predictive power on specific datasets, experiments testing generalization across datasets indicate that further work is needed to overcome domain shift and improve applicability to unseen protein-DNA complexes.

SAMMANFATTNING

Att förutsäga bindningsaffiniteten mellan proteiner och DNA är en fundamental utmaning inom molekylärbiologin, avgörande för förståelsen av genreglering och den molekylära grunden för sjukdomar. Konventionella metoder har ofta svårt att hantera komplexiteten i dessa interaktioner och den begränsade tillgången på strukturdata. Denna avhandling utforskar ett nytt ramverk för djupinlärning för att förutsäga protein-DNA-bindningsaffinitet genom att utnyttja strukturella inbäddningar genererade av AlphaFold 3, en toppmodern modell för förutsägelse av proteinstrukturer. Den föreslagna metoden representerar protein-DNA-komplex som fullständigt anslutna grafer, där aminosyra- och nukleotidrester är noder och deras potentiella interaktioner är kanter. AlphaFolds "single representation" fungerar som nodegenskaper och dess "pair representation" tillhandahåller kantegenskaper, vilket eliminerar behovet av handgjorda features. Studien innehåller en jämförande analys av olika arkitekturer för grafneuronnätverk

Author's address: Marcus Warnerfjord, warner@kth.se, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, Stockholm, Sweden, SE 100 44.

Document date: June 16, 2025

Swedish title: Prediktering av protein-DNA-bindningsaffinitet med AlphaFold-inbäddningar och graf-attentionsnätverk

(GNN), inklusive Graph Convolutional Networks (GCN), Graph Attention Networks (GAT) och Graph Transformers. Modellerna utvärderades på tre olika dataset: två högkapacitets HiTS-FLIP-dataset (KLF1 och LacI) och det mer varierade PDBbind-datasetet. Resultaten visar att uppmärksamhetsbaserade arkitekturer, särskilt GAT, presterar bättre än GCN-baslinjen på HiTS-FLIP-dataseten och uppnår en Pearson-korrelationskoefficient (PCC) på upp till 0.94 efter hyperparameteroptimering på KLF1-datasetet. Prestandafördelen var dock mindre uttalad på det heterogena PDBbind-datasetet. Vår analys avslöjade att modellernas prestanda är mer känslig för nodegenskaper än för kantegenskaper, vilket tyder på att nuvarande GNN-arkitekturer kanske inte fullt ut utnyttjar den rika interaktionsinformationen i AlphaFolds "pair representation". Medan modellerna visar stark prediktiv förmåga på specifika dataset, indikerar experiment som testar generalisering över olika dataset att ytterligare arbete krävs för att övervinna domänskiften och förbättra tillämpbarheten på tidigare osedda protein-DNA-komplex.

CCS Concepts: • **Applied computing** → **Molecular structural biology**; • **Computing methodologies** → **Neural networks**.

Keywords: Protein-DNA binding affinity, Graph Neural Networks, AlphaFold, Bioinformatics, Structure-based prediction

Nyckelord: Protein-DNA-bindningsaffinitet, neurala grafnätverk,AlphaFold, Bioinformatik, Struktur baserad prediktion

1 INTRODUCTION

Transcription is the process of decoding DNA to messenger RNA and constitutes the core function of gene expression and, by extension, life itself [15]. Not only is this process fundamental to the development of life, but it is also the basis for many diseases and their treatments [1]. The process of transcription is regulated by a complex interplay of proteins and DNA. The proteins that interact with DNA and regulate transcription are known as transcription factors, and they bind to specific sequences in the DNA to either promote or inhibit the transcription of genes. Understanding the interactions between proteins and DNA is therefore crucial for understanding how genes are regulated and how this regulation can influence cell behavior and disease.

Several diseases, such as cancer, are caused by mutations in the DNA that affect the binding of transcription factors. For example, mutations that disrupt the binding of a transcription factor to its target site can lead to the overexpression of a gene, which in turn can cause uncontrolled cell growth and cancer [10]. Similarly, mutations that enhance the binding of a transcription factor to its target site can lead to the repression of a gene, which can also cause disease. Therefore, understanding the mechanisms by which transcription factors bind to DNA is essential for understanding the molecular basis of diseases and for developing new treatments. To understand and predict the binding of proteins to specific DNA sequences, researchers need to develop computational methods that can accurately model the interactions between proteins and DNA [9]. This is a challenging problem due to the complexity of the interactions involved and the limited data available for protein-DNA complexes. In general, we are currently unable to quantitatively predict the binding of proteins to DNA sequences, which limits our understanding of gene regulation and its implications for disease

Deep learning has emerged as a powerful tool for solving complex problems in the field of data-driven life sciences [17]. Deep learning is a type of machine learning that uses neural networks to learn complex patterns in data. Neural networks are composed of layers of interconnected nodes that can learn to represent complex relationships between input and output data. By training neural networks on large datasets, researchers can develop models that can predict complex biological phenomena, such as the binding of proteins to DNA.

Acquiring this data can be slow and expensive, and the data that is available is often noisy and biased [13]. This makes it difficult to develop accurate models for predicting the binding of proteins to DNA. With the recent development of high-throughput methods for measuring binding affinity, data acquisition has become significantly cheaper and more accurate [11, 16, 18, 23]. These methods use large libraries of DNA sequences and measure the binding affinity of one protein to each sequence variant. By analyzing the binding rates, researchers can identify the specific sequences that are bound by a protein and determine the binding affinity of hundreds of thousands of sequences at once. This allows researchers to generate large datasets of binding affinity data that can be used to train deep learning models.

2 TECHNICAL BACKGROUND

Genes are the basic units of heredity in living organisms. They contain the instructions for making proteins, which are the molecules that carry out most of the functions in cells. The process of gene expression, which involves the transcription of DNA into RNA and the translation of RNA into protein, is tightly regulated to ensure that the right proteins are made at the right time and in the right place. One of the key regulators of gene expression is the binding of proteins to specific sequences in the DNA. These proteins, known as transcription factors, can either promote or inhibit the transcription of genes by recruiting other proteins that help to activate or repress the transcription machinery [15].

DNA encodes the information for the sequence of amino acids in a protein. The sequence of amino acids determines the three-dimensional structure of the protein, which in turn determines its function. Proteins are large molecules that can adopt a wide range of shapes, and their function is often determined by their ability to bind to other molecules, such as DNA, RNA, or small molecules. The binding of proteins to DNA is mediated by a combination of electrostatic, hydrophobic, and van der Waals forces, which are difficult to model accurately [15].

2.1 Binding Affinity

To measure the strength of the binding between a protein and DNA, researchers often use a quantity known as the binding affinity, which is based on the equilibrium of a chemical reaction:

$$k_{on} \cdot [\text{Protein}] \cdot [\text{DNA}] = k_{off} \cdot [\text{Complex}] \tag{1}$$

where [Protein] is the concentration of the protein, [DNA] is the concentration of the DNA, and [Complex] is the concentration of the protein-DNA complex. k_{off} is the rate constant for the dissociation of the complex, and k_{on} is the rate constant for the formation of the complex. The binding affinity can be expressed in terms of the dissociation constant K_D, which is defined as the ratio of the rate constants for the dissociation and association of the complex and is equivalent to the equilibrium constant for the reaction:

$$K_D = \frac{k_{off}}{k_{on}} = \frac{[\text{Protein}]_{eq} \cdot [\text{DNA}]_{eq}}{[\text{Complex}]_{eq}} \tag{2}$$

The dissociation constant is measured in molarity (M) and is inversely proportional to the binding affinity, meaning that a lower value of K_D indicates a stronger binding affinity [15].

When working with K_D in a computational environment, it is more feasible to use the natural logarithm $\ln(K_D)$. Then the dissociation can be expressed in terms of $\Delta G = -RT \ln(K_D)$, where R is the universal gas constant and T is the temperature. This relationship is often expressed in units of $K_B T$, where K_B is the Boltzmann constant. K_BT represents the thermal energy available per molecule at a given temperature T, and using it as a unit for ΔG allows for a direct comparison of energy changes to the inherent thermal energy of the system. This provides a convenient way to assess the favorability of a binding event relative to random thermal fluctuations.

2.1.1 HiTS-FLIP. High Throughput Selection - Fluorescent Ligand Interaction Profiling (HiTS-FLIP) uses fluorescent microscopy and incorporates fluorescent ligands to measure the increase and decrease of the target molecule over time [18]. This method involves labeling the target molecule with a fluorescent tag, flowing it over an array with clusters of DNA sequence mutations, and then using image quantification to measure the increase and decrease of fluorescence—i.e., ligand concentration—as the fluorescent protein accumulates over certain DNA clusters. When the quantified fluorescence signal saturates, i.e., when the binding of the protein to the DNA has reached equilibrium for some cluster, K_D is the concentration of the ligand at the time when that particular signal has reached 50% of its final saturation value.

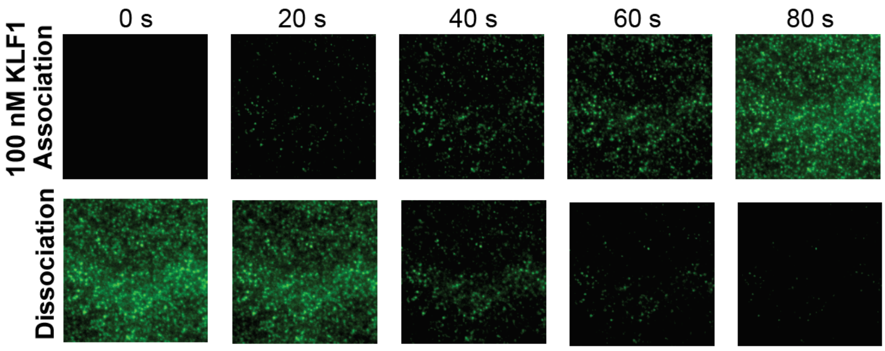


Figure 1: Fluorophores light up as the protein KLF1 binds to its DNA target, revealing an increase of complexes as the protein is flowed over the cell. Conversely, dissociation is seen as a decrease in fluorescent points. Images were generated by Emil Marklund.

2.2 Protein Folding and AlphaFold 3

Protein folding is the process by which a chain of amino acids adopts its three-dimensional structure, which is essential for its function. The three-dimensional structure of a protein is determined by its amino acid sequence, which is encoded in the DNA. The folding of a protein is a complex process that involves the formation of secondary structures, such as alpha helices and beta sheets, and the packing of these structures into a compact three-dimensional shape. The folding of a protein is driven by a combination of forces, including hydrogen bonding, van der Waals forces, and hydrophobic interactions. These forces act to stabilize the protein in its native conformation, which is the most stable state of the protein [15].

The problem of protein folding has been a long-standing challenge in biology, as the number of possible conformations that a protein can adopt is astronomically large. This is due to the fact that a protein is a long chain of amino acids that can adopt a wide range of conformations, and the interactions between the amino acids are complex and difficult to model. However, recent advances in deep learning have led to the development of new methods for predicting the three-dimensional structure of proteins with high accuracy. One of the most successful methods is AlphaFold, which uses a deep neural network to predict the structure of a protein from its amino acid sequence. AlphaFold has been shown to outperform other methods for protein structure prediction and has been used to predict the structures of thousands of proteins with high accuracy. The development of AlphaFold 3 has further extended this functionality to not only predict the structure of single proteins but also the structure of protein-DNA complexes [2].

The AlphaFold pipeline consists of *three* main modules. The first is the input preparation module, which takes the sequence of amino acids forming the protein and the sequence of nucleotides that forms the DNA as input; these are collectively known as *residues* when they are tokenized. The input preparation module uses a multiple sequence alignment (MSA) to identify homologous sequences in the protein and DNA sequences, which are then used to generate a set of features that describe the input data. It does so by aligning multiple sequences of DNA or protein to identify conserved regions and evolutionary relationships between the sequences. The input preparation module generates a set of features that describe the input data, such as the amino acid composition, secondary structure, and solvent accessibility of the protein, as well as the nucleotide composition and secondary structure of the DNA [2].

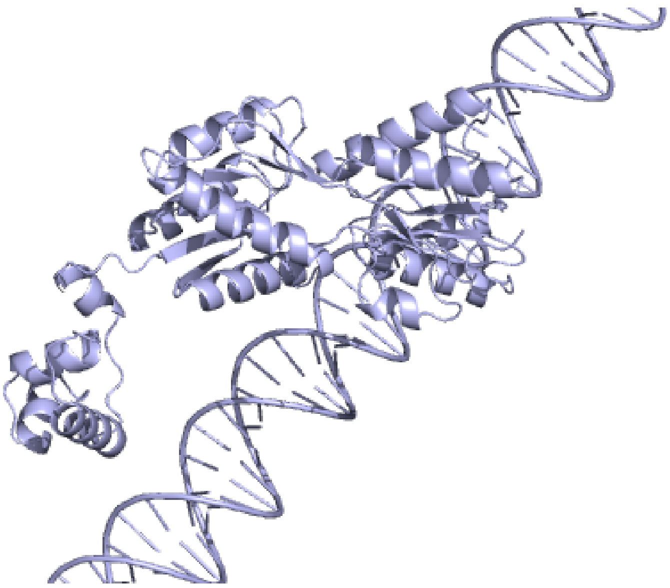


Figure 2: A PyMol 3D render of a protein-DNA complex output from AlphaFold 3.

The second part is the representation learning module, which is the main source of the model's predictive power. AlphaFold mainly uses two sets of representations throughout its network. First, the *single representation* is a list of feature vectors of shape $(N_{\text{tokens}}, 348)$ for the tokenized residues of the protein and DNA sequences. Second, the *pair representation* is a tensor of features of shape $(N_{\text{tokens}}, N_{\text{tokens}}, 128)$, where N_{tokens} is the number of tokenized residues from the protein and DNA sequences. The pair representation is used to model the interactions between residues in the protein and DNA sequences, which plays a crucial role in determining the structure of the complex. The representation learning module is the largest in the network and contributes by embedding the input data into a high-dimensional space, where the relationships between the residues can be modeled more effectively. This abstract representation is used by the third and final module to predict the three-dimensional structure of the protein-DNA complex as a list of positions [2]. An example of a 3D render of these positions can be seen in Figure 2.

2.3 Regression

Regression problems are a class of supervised learning tasks where the goal is to predict a continuous numerical value based on input data. These problems arise in various domains, such as predicting house prices, temperature, or stock market trends. In a regression setting, the model learns a mapping $f : X \rightarrow y$ from input features $X \in \mathbb{R}^d$ to a real-valued output y by minimizing a loss function that measures the difference between predicted and actual values, commonly using metrics like mean squared error [19].

Deep learning approaches regression using artificial neural networks composed of multiple layers of interconnected neurons. These networks are universal function approximators, capable of learning complex, nonlinear relationships in data through hierarchical representations. By adjusting the weights of the connections during training via backpropagation and optimization algorithms like stochastic gradient descent, deep learning models can capture subtle patterns in large datasets. This makes them particularly effective for high-dimensional or unstructured data such as images, audio, or time series, where traditional regression techniques may struggle to perform adequately. One drawback of using traditional machine learning is that it often requires feature engineering when designing X. This usually entails relying on human expertise to engineer features that accurately capture some key aspect of the underlying distribution of the data. Deep learning mitigates this by learning representations implicitly from the data, and can therefore make use of the features most relevant for the task at hand [19].

2.4 Graph Neural Networks

Graph neural networks (GNNs) are a type of neural network that is specifically designed to learn on data that exhibits gridless dependencies. In contrast to other deep neural networks that are trained on grid-based structures such as images or text, GNNs are trained to propagate data based on the interactions between nodes in a graph. More formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a mathematical structure that consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ that connect the nodes. In the context of neural networks, we define a graph as a node feature matrix $X \in \mathbb{R}^{N \times d}$, where N is the number of nodes and d is the number of features per node. The edges are defined as an adjacency matrix $A \in \mathbb{R}^{N \times N}$, which defines the connections between the nodes in the graph. The adjacency matrix is a square matrix that represents the connections between the nodes in the graph, where $\mathbf{A}_{ij} = 1$ if there is an edge between nodes i and j, and $\mathbf{A}_{ij} = 0$ otherwise. In addition to the node features and adjacency matrix, GNNs can also contain a set of edge features $\mathbf{A}_{ij} = \mathbf{e}_{ij}$ where $\mathbf{e}_{ij} \in \mathbb{R}^m$, which are capable of describing the relationship between node i and j in m dimensions, thus enabling more complex relationships between the nodes in the graph to be captured [8].

To learn on graph data, GNNs use a message passing algorithm that allows the nodes in the graph to exchange information with their neighbors. The message-passing model can be expressed in terms of an update function and an aggregation function. The update function is used to update the features of the nodes in the graph, while the aggregation function is used to combine the features of the neighbors of each node. The update function can be expressed as:

$$h_i^{(t+1)} = \text{UPDATE}(h_i^{(t)}, \text{AGGREGATE}(\{h_j^{(t)} \mid j \in N(i)\})) \tag{3}$$

where $h_i^{(t)}$ is the feature vector of node i at time t, and $N(i)$ is the set of neighbors of node i. The aggregation function is used to combine the features of the neighbors of the node. It is important that the operations used in the aggregation are invariant to the order of the neighbors, as the order of the neighbors should not affect the output of the function. This is what makes GNNs able to learn on unstructured data, as the order of the nodes in the graph does not matter. The aggregation function can be any function that is invariant to the order of the neighbors, such as summation, mean, or max pooling. The update function can be any function that takes as input the feature vector of the node and the aggregated features of its neighbors, such as a neural network or a linear transformation [8].

By iteratively applying these update functions, GNNs can learn to represent complex relationships between the nodes and edges in the graph. GNNs have been shown to be effective in a wide range of applications, including social network analysis, recommendation systems, and molecular property prediction. In these applications, GNNs have been used to learn the relationships between nodes and edges in the graph, which allows them to make predictions about the properties of the graph. For example, in molecular property prediction, GNNs have been used to predict the binding affinity of ligands to proteins by learning the relationships between the atoms in the ligand and protein molecules [8].

Graph attention networks (GATs) are a type of GNN that uses attention mechanisms to learn the relationships between nodes in the graph [21]. Attention mechanisms allow the model to focus on specific nodes and edges in the graph by learning attention weights that scale the contribution of each message based on the learned importance of the edge, which can improve the performance of the model. A message passing network for GATs can be expressed as:

$$h_i^{(t+1)} = \sum_{j \in N(i)} \alpha_{ij} W h_j^{(t)} \tag{4}$$

where α_{ij} is the attention weight for the edge between nodes i and j, and W is a learnable weight matrix. The attention weight was originally derived only from the node features of the graph, but in later implementations, the framework PyTorch Geometric calculates the attention weights as:

$$\alpha_{ij} = \frac{\exp\left(\mathbf{a}^T \text{LeakyReLU}(W h_{i(t)} + W h_{j(t)} + W_e \mathbf{e}_{ij})\right)}{\sum_{k \in N(i)} \exp\left(\mathbf{a}^T \text{LeakyReLU}(W h_{i(t)} + W h_{k(t)} + W_e \mathbf{e}_{ik})\right)}$$

where W and W_e are learnable weight matrices for the node and edge features respectively, $\mathbf{a}$ is a learnable weight vector, LeakyReLU is a non-linear activation function, and $\mathbf{e}_{ij}$ is the edge feature vector for the edge between nodes i and j [5, 6].

3 RELATED WORK

Predicting binding affinity using machine learning can be categorized into two main approaches: interaction-free and interaction-based models. These approaches differ in their reliance on physical interactions to inform predictions, with interaction-free models focusing on data-driven inference and interaction-based models utilizing structural information from protein-ligand complexes [13].

3.1 Interaction-based models

Interaction-based models predict binding affinity by explicitly modeling the *physical interactions* between proteins and ligands. These models often leverage physical properties or the three-dimensional structures of protein-ligand complexes, employing methods such as 3D voxel grids and graph neural networks (GNNs) to capture the spatial relationships and interactions between atoms [13].

Interaction-based approaches suffer from limitations, as they require known protein-ligand complex structures to make accurate predictions. This restricts their applicability to cases where the binding site and interactions are well-characterized. Also, these models often rely on human-engineered features, which can be suboptimal due to the complexity of the interactions involved and our limited understanding of the underlying chemistry [3, 4]. For example, voxel grid-based methods discretize the 3D space around the binding site into a grid, which can lead to loss of information and inaccuracies in modeling interactions. Similarly, graph-based methods rely on predefined graph structures that may not capture all relevant interactions in the complex [13].

In contrast, interaction-free methods can generalize better to unknown protein-ligand complexes, as they do not rely on specific structural information.

3.2 Interaction-free models

In interaction-free models, the binding affinity is predicted without explicitly modeling the physical interactions between proteins and ligands. These models rely on data-driven approaches to infer binding affinity from the sequences of molecules and proteins without considering their structural information. These models are primarily based on sequence-based representations, such as SMILES for molecules and protein sequences for proteins. The models learn to predict binding affinity by analyzing the patterns in the sequences and their corresponding binding affinity values [13].

In previous work, researchers have used sequential prediction models, such as recurrent neural networks, to model the interactions between proteins and DNA. Fries et al. showed that the relationship between KLF1 and various DNA sequences could be determined with a coefficient of determination $R^2 = 0.97837$ on the validation set [7]. This model performs well, but since the protein is fixed, the generalization is limited as the model will inevitably fail to represent diversity in the protein domain. Furthermore, because the input consists of a text caption of the DNA, the model is forced to learn from the sequential patterns of the DNA input strings rather than modeling any underlying distribution related to the physical nature of the complex. Consequently, it fails to generalize to new proteins as these models learn to relate the order of the nucleotides to the binding affinity of one specific protein. This problem might still occur when using physical properties in the input features, as the sequence still risks being learned implicitly from the input data if this presents a local minimum in the loss function.

SAMPDI-3D instead trains on two separate datasets, with one dataset containing protein mutations and the other containing DNA mutations. However, this model is also interaction-based, and since these two datasets are trained separately, the model is also unable to generalize to new protein-DNA complexes [20]. This is because the model learns to relate the order of the nucleotides to the binding affinity of one specific protein, or the sequence of amino acids to a specific DNA sequence if one of them is fixed, without considering the structural information of the protein-DNA complex. Therefore, it is crucial to vary both the protein and DNA sequences in the input data, to avoid implicitly learning the sequence patterns of one specific protein-DNA complex.

4 CONTRIBUTIONS

The objective of this study is to develop a structure-based deep learning framework for predicting protein-DNA binding affinity by exclusively leveraging AlphaFold-generated structural embeddings without handcrafted features or sequence information. We investigate whether attention-based GNNs (GAT, Graph Transformers) outperform baseline GCNs by better capturing complex residue-nucleotide interactions in fully connected graphs. Further, we hypothesize that utilization of edge features should greatly influence performance and therefore evaluate model reliance on edge vs. node features via noise ablation.

We then further investigate the performance of the chosen architectures by evaluating their ability to generalize across diverse protein-DNA complexes. This study aims to address the challenge of predicting protein-DNA binding affinity by making the following contributions:

- We propose and evaluate a novel approach that utilizes embeddings from AlphaFold 3, a state-of-the-art protein structure prediction model, as input features for Graph Neural Networks (GNNs) to predict protein-DNA binding affinity. Specifically, AlphaFold's single representation is used for node features (amino acids and nucleotides) and the pair representation for edge features (interactions between them).
- We conduct a comparative analysis of different GNN architectures, including Graph Convolutional Networks (GCN) as a baseline, Graph Attention Networks (GAT), and Graph Transformers, to determine their efficacy in modeling protein-DNA interactions from AlphaFold embeddings.
- We investigate the relative importance and contribution of node features (derived from AlphaFold's single representation) versus edge features (derived from AlphaFold's pair representation) in predicting binding affinity. This includes assessing model performance with and without edge features, and sensitivity to noise in these features.
- We systematically evaluate the proposed models on multiple datasets: two high-quality, high-throughput HiTS-FLIP datasets (KLF1 and LacI, where the protein is fixed and DNA varies) and the more diverse PDBbind dataset (various protein-DNA complexes).
- We perform hyperparameter optimization for the best-performing GNN architecture to maximize its predictive power.
- We assess the generalization capabilities of the optimized model, including its performance when trained on one dataset and tested on another, and when trained on combined datasets, to evaluate its potential for predicting affinity for unseen protein-DNA complexes.

5 METHOD

5.1 Data Collection

The data used in this study were collected from the PDBbind database and two internal HiTS-FLIP experiments. The former contains binding affinity data for a wide range of protein-DNA complexes, while the latter consists of two datasets—KLF1 and *lac* repressor (LacI)—in which the protein sequences are fixed over a wide range of DNA mutations. The PDBbind dataset is used to test the model's ability to predict binding affinity with variations in both protein and DNA. KLF1 contains measurements that are directly comparable with the measurements in the PDBbind dataset and therefore has the potential to further extend the dataset by combining the two. The LacI dataset has an undetermined scale factor that might make it less suitable in that regard. However, it contains a substantial amount of measurements and will therefore be incorporated.

5.1.1 PDBbind. PDBbind is a database of protein-ligand complexes that provides a comprehensive collection of binding affinity data for a wide range of protein-ligand pairs [14]. The database contains several subsets for different ligand types; in this case, the protein–nucleic acid set was used. Each entry in the dataset contains information on the three-dimensional structure of the protein-ligand complex, as well as the binding affinity data for each complex. The binding affinity data are provided in the form of dissociation constants (K_D), inhibition constants (K_i), or half maximal inhibitory concentrations

(IC_{50}). The set was pruned to only include those with K_D values, as these are the most reliable and consistent measures of binding affinity [13].

5.1.2 KLF1 and LacI. The KLF1 and LacI datasets were collected from two internal HiTS-FLIP experiments. The KLF1 dataset contains binding affinity data for a fixed protein (KLF1) with varying DNA mutations, while the LacI dataset contains binding affinity data for a fixed protein (LacI) with varying DNA mutations. These datasets were chosen because they are comparatively large and contain high-quality data with low measurement error.

Table 1: Dataset and Split Sizes across all datasets

Dataset	Train	Validation	Total
KLF1	247	53	300
LacI	1564	335	1899
PDBbind	488	105	593
Total	**2299**	**493**	**2792**

5.2 Graph Representation of Protein-DNA Complexes

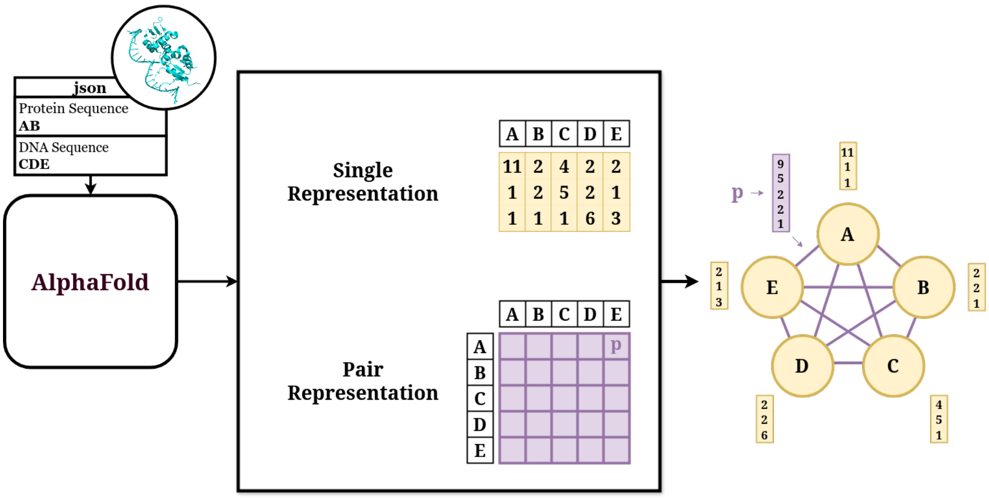


Figure 3: High-level overview of the graph representation and data pipeline. A JSON specification of the protein-DNA sequences is fed into AlphaFold 3, which constructs the single and pair embeddings. These are then used to build fully connected graphs.

Protein-DNA complexes are represented as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for processing by graph neural network (GNN) models. Here, $\mathcal{V}$ denotes the set of nodes, corresponding to amino acid residues from the protein and nucleotides from the DNA, and $\mathcal{E}$ denotes the set of edges that represent potential interactions between them.

5.2.1 Node Features. Each node $v \in \mathcal{V}$, representing either an amino acid or a nucleotide, is associated with a (348)-dimensional AlphaFold single representation vector $\mathbf{x}_k$. This vector, obtained as further described in Section 5.3, serves as the initial feature representation for the node. Collectively, these vectors form the node feature matrix $X \in \mathbb{R}^{N \times 348}$, where N is the total number of residues and nucleotides in the complex.

5.2.2 Edge Features and Connectivity. To comprehensively model all potential residue-nucleotide and intra-chain interactions retrieved from the pair embeddings, we construct fully connected graphs. This means an edge (v_i, v_j) is defined between every pair of nodes i and j, including protein–protein, DNA–DNA, and protein–DNA pairs. Each edge (v_i, v_j) is attributed with a (128)-dimensional feature vector derived from the corresponding AlphaFold pair representation tensor element $\mathbf{A}_{ij}$. Although this approach results in a large set of edges, we have no prior knowledge of which edges are important for the task at hand. Because of this, we can not exclude any edges a priori, and therefore need to use a model that can learn to focus on the most relevant edges.

5.3 Data Preprocessing

The datasets were constructed by specifying protein-DNA sequence information in JSON files formatted for input to AlphaFold 3. These files were then processed through the AlphaFold pipeline to generate the single and pair embeddings used as model input.

To stabilize training and improve model performance, all features were z-normalized to have zero mean and unit variance. This normalization ensures that features are on a similar scale, which can aid model convergence. The z-normalization is defined as:

$$x_i = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}} \tag{5}$$

where x_i is a node or edge feature, and μ_{x_i}, σ_{x_i} are the mean and standard deviation of feature x_i, computed over the training set.

The labels y were also z-normalized. However, due to the range of values and to reduce the risk of numerical instability, a natural logarithm transformation was applied prior to normalization:

$$y' = \ln(y) \tag{6}$$

followed by z-normalization of y'.

5.4 Models

5.4.1 Baseline: Graph Convolutional Network (GCN). The GCN [12] serves as our baseline model. GCN layers update node representations by aggregating feature information from their local graph neighborhood. The propagation rule typically involves a normalized sum of neighbor features followed by a linear transformation and a nonlinear activation function. This allows the model to learn hierarchical representations of node features.

5.4.2 Attention-Based Architectures. To capture more complex and potentially long-range interactions within the fully connected protein-DNA graphs, we evaluate two attention-based GNN architectures. The underlying hypothesis is that GAT and Graph Transformer models will exhibit superior performance over the baseline GCN due to their enhanced capacity to capture complex relationships inherent in protein-DNA interaction data. In part due to their ability to incorporate edge features in the input graphs, but also because the attention mechanism allows these models to weigh the importance of different connections within the graphs differently. As the edge features are derived from the AlphaFold-generated embeddings, we won't know a priori which edges are important for the task at hand, and should therefore benefit from a model that can learn to focus on the most relevant edges.

Graph Attention Network (GAT). GATs [21] introduce an attention mechanism to the aggregation process. Instead of using fixed, normalized convolution weights like GCNs, GATs learn to assign different importance scores (attention weights) to different nodes within a neighborhood. This allows the model to selectively focus on more relevant interactions when updating node representations, which is hypothesized to be beneficial for modeling intricate protein-DNA interfaces.

Graph Transformer. Graph Transformers [22] extend the concept of attention to a global scale, akin to Transformers in natural language processing. Each node can attend to all other nodes in the graph (especially relevant in our fully connected graph setting), allowing for a more expressive and flexible way to model dependencies between distant residues and nucleotides. Positional encodings and edge features are typically incorporated to provide structural information.

5.5 Model Training and Optimization

The training strategy is designed to effectively learn the mapping from structural embeddings to binding affinities and to enable fair comparison between different model architectures.

5.5.1 Loss Function. All models are trained by minimizing the Huber loss between the predicted and true binding affinities. The Huber loss is chosen for its robustness to outliers compared to mean squared error, as it behaves quadratically for small errors and linearly for large errors. The target values y are the z-normalized $\ln(K_D)$ values, as described in the Data Preprocessing section.

Let $a_i = \hat{y}_i - y_i$ be the error for the i-th sample, where y_i is the true preprocessed binding affinity and $\hat{y}_i$ is the model's prediction. The Huber loss for a single sample, $L_\delta(a_i)$, is defined as:

$$L_\delta(a_i) = \begin{cases} \frac{1}{2}a_i^2 & \text{if } |a_i| \leq \delta \\ \delta(|a_i| - \frac{1}{2}\delta) & \text{if } |a_i| > \delta \end{cases} \tag{7}$$

where δ is a hyperparameter that defines the threshold at which the loss transitions from quadratic to linear.

The overall Huber loss for a batch of B samples is then the average of these individual losses:

$$\mathcal{L}_{\text{Huber}} = \frac{1}{B}\sum_{i=1}^{B} L_\delta(y_i - \hat{y}_i) \tag{8}$$

This formulation helps mitigate the disproportionate influence of outliers during training, potentially leading to more stable and generalizable models. The value of δ is typically determined through hyperparameter tuning but will remain fixed at 1 throughout.

5.5.2 Optimizer and Hyperparameters. Stochastic Gradient Descent (SGD) with momentum (0.995) and an exponentially decaying learning rate ($\gamma = 0.995$) is used for training all models. Initial hyperparameters, such as learning rate, number of GNN layers, hidden dimension sizes, and dropout rates, are kept consistent across models during the initial "Model Selection" phase described in the Evaluation section. For the "Final Model Optimization" phase, a Bayesian search algorithm is employed to find the optimal set of hyperparameters for the best-performing architecture on each dataset.

5.5.3 Training Procedure. Models are trained for a fixed number of epochs or using an early stopping criterion based on performance on a validation set (a subset of the training data). This helps prevent overfitting and selects the model that generalizes best to unseen data within that training fold. The data is typically shuffled before each epoch. The specifics of data splitting for training, validation, and testing are further detailed in the context of each experiment in the Evaluation section.

5.5.4 Hardware and Software. All model training and hyperparameter optimization experiments were performed on the Berzelius supercomputer, a high-performance computing (HPC) cluster. The computations were accelerated using NVIDIA A100-SXM4 GPUs, with both 40GB and 80GB memory variants available on the cluster. The graph neural network models were implemented using the PyTorch deep learning framework, with graph-specific layers and data structures provided by the PyTorch Geometric (PyG) library. To ensure reproducibility and to facilitate analysis, all training metrics, model configurations, and validation results were systematically logged and visualized using the Weights & Biases (WandB) platform.

5.6 Evaluation

The evaluation of the proposed approach is designed to assess its performance and generalization capabilities. The model's predictive accuracy is examined by comparing results across different datasets, ensuring that the framework not only achieves proficiency on the high-quality HiTS-FLIP datasets but also demonstrates comparable performance on the PDBbind dataset. This comparative analysis across datasets serves as a crucial indicator of the model's robustness and potential for broader applicability.

5.6.1 Model Selection. A model selection process was conducted to compare the performance of the baseline GCN against more sophisticated attention-based GNNs, namely GAT and Graph Transformers. For this comparison, all models were configured with the fixed hyperparameters in Figure 2 and trained on the same datasets. The evaluation relies on comparing the Pearson Correlation Coefficient (PCC) and Mean Squared Error (MSE) achieved by each architecture.

Table 2: Hyperparameter values during model selection

Hyperparameter	**Value**
Batch Size	16
Learning Rate	0.04
Epochs	100
Embedding Size	128
Layers	2
Number of Heads	4
Dropout	0.2
Weight Decay	0.0005

5.6.2 Edge Features versus Node Features. The relative importance of edge features versus node features was investigated. This involves evaluating the performance of the baseline GCN, GAT, and Graph Transformer models when edge features are entirely omitted. Additionally, the impact of noisy edge features was assessed by introducing additive noise to these features, allowing for an understanding of model sensitivity and reliance on the information encoded in the graph edges.

5.6.3 Final Model Optimization. After identifying the best-performing model architecture from the preceding experiments, final model optimization was performed. This stage involved hyperparameter tuning specifically for the selected model. A Bayesian search algorithm was employed to explore the hyperparameter space, focusing on parameters such as learning rate, dropout rate, and the number of layers. Due to computational resource constraints, the LacI set was omitted.

The hyperparameter configurations used for the two experimental sweeps are detailed in Table 3. This table outlines the specific values or ranges explored for each hyperparameter across the KLF1 and PDBbind datasets. For parameters like learning rate and weight decay, a uniform distribution was sampled within the specified minimum and maximum bounds. Other parameters, such as batch size and model-specific architectural choices (dropout, embedding size, layers, and number of heads), were selected from a predefined set of discrete values. This approach allowed for a comprehensive search of the hyperparameter space for each dataset.

Table 3: Hyperparameter Ranges for Sweeps on KLF1 and PDBbind Datasets

Hyperparameter	**KLF1**	**PDBbind**
Batch Size	[16, 32]	[16]
Learning Rate	Uniform(0.01, 0.04)	Uniform(0.01, 0.05)
Weight Decay	Uniform(0.001, 0.01)	Uniform(0.001, 0.01)
Dropout	[0.3, 0.4, 0.5]	[0.3, 0.4, 0.5]
Embedding Size	[64, 128, 256]	[64, 128]
Layers	[1, 2, 3, 4]	[1, 2, 3]
Number of Heads	[2, 4]	[2, 4]

5.6.4 Generalization Across Datasets. The generalization capabilities across datasets are assessed. This involves training the optimized model on the PDBbind dataset and subsequently validating its performance on the KLF1 dataset to test its ability to generalize to data it has not encountered during training. Further investigation explores whether combining datasets during the training phase can lead to increased generalization performance across varying protein sequences. The ultimate aim is to evaluate the model's performance on entirely unseen protein-DNA complexes.

5.6.5 Metrics. The model's predictive performance is quantified using two primary metrics. The first metric is the Mean Squared Error (MSE), which measures the average squared difference between the true and predicted binding affinities. The MSE is calculated as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

where y_i represents the true binding affinity, $\hat{y}_i$ is the predicted binding affinity, and n is the total number of samples in the dataset.

The second metric is the Pearson Correlation Coefficient (PCC), which assesses the linear correlation between the predicted binding affinities and the experimentally measured binding affinities. The PCC is calculated as follows:

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{10}$$

where x_i and y_i are the predicted and experimental binding affinities for the i-th sample, respectively, $\bar{x}$ and $\bar{y}$ are the mean values of the predicted and experimental binding affinities across all samples, and n is the number of data points. The PCC value ranges from -1 to 1, where a value of 1 indicates a perfect positive linear correlation, -1 signifies a perfect negative linear correlation, and 0 indicates no linear correlation.

6 RESULTS

This section presents the outcomes of the experiments detailed in the Evaluation section 5.6. The results are organized according to the experimental objectives, starting with model selection, followed by an analysis of feature importance, final model optimization, and concluding with an assessment of generalization capabilities. All results are collected from evaluations on the validation sets.

6.1 Model Architecture Performance Comparison

The initial set of experiments focused on comparing the predictive performance of the baseline Graph Convolutional Network (GCN) with the attention-based Graph Attention Network (GAT) and Graph Transformer architectures. Performance was assessed using Mean Squared Error (MSE) and Pearson Correlation Coefficient (PCC) across the HiTS-FLIP (KLF1 and lacI) and PDBbind datasets, with all models configured with fixed hyperparameters.

Table 4 summarizes the MSE and PCC values obtained for each model across the tested datasets. Figure 4 provides a visual comparison of PCC values of each tests. The GAT and Graph Transformer models exhibited lower MSE and higher PCC values compared to the baseline GCN over the HiTS-FLIP datasets, with marginal performance differences between them. However, on the PDBbind dataset, the more complex architectures failed to outperform the simpler GCN model in terms of MSE and showing minuscule performance gains overall.

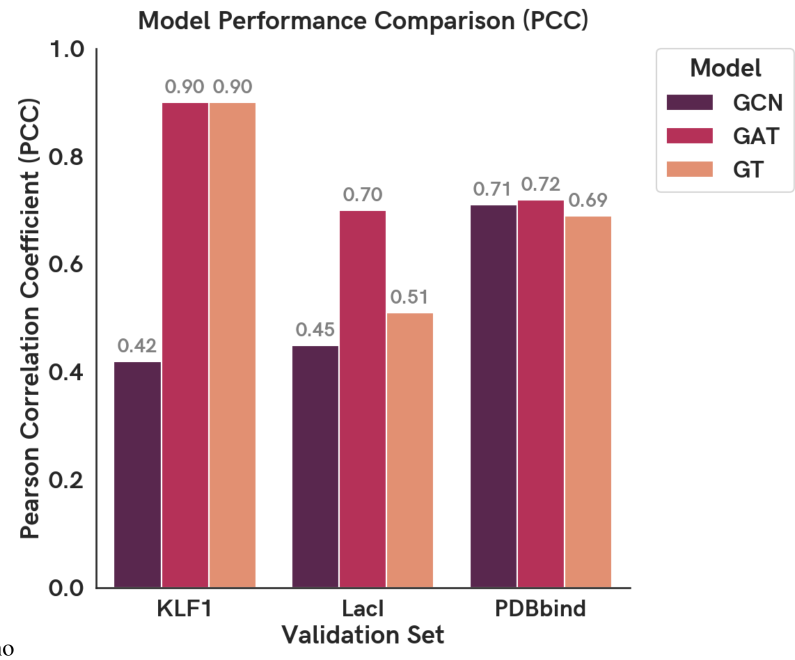
Table 4: Performance comparison of GCN, GAT, and Graph Transformer models across datasets. Lower MSE and higher PCC indicate better performance, with best values for each dataset in bold.

	KLF1			lacI			PDBbind		
Metric	GCN	GAT	GT	GCN	GAT	GT	GCN	GAT	GT
MSE	1.10	0.24	**0.21**	0.84	**0.57**	0.78	**0.56**	0.57	0.66
PCC	0.42	0.90	**0.90**	0.45	**0.70**	0.51	0.71	**0.72**	0.69

6.2 Impact of Edge and Node Features

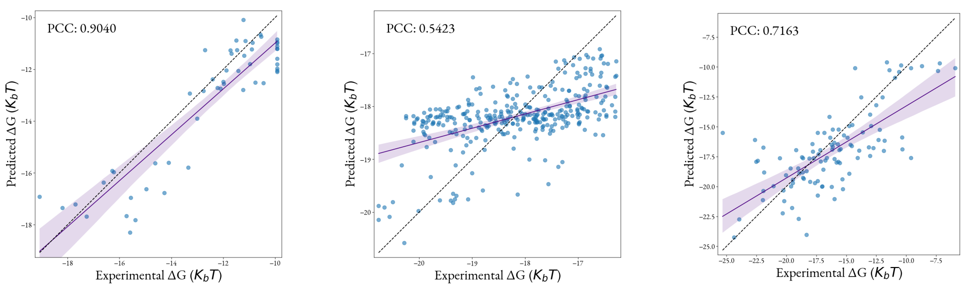
To understand the contribution of different feature types, experiments were conducted to evaluate model performance on the PDBbind dataset under two conditions: first, with edge features entirely removed, and second, with additive Gaussian noise ($\mu = 0, \sigma = 0.2$) introduced to the normalized features over 30 epochs.

Fgure 6 shows the performance of GAT and Graph Transformer models with and without edge features. Only for the KLF1 dataset do edge features make a considerable contribution to model performance. This is further supported by the fact that GCN, which by default doesn't support edge features, also performed worst on the KLF1 dataset. This indicates that edge features do not consistently contribute to performance across all cases. Figure 7 details the impact of progressively adding noise to edge features (with no noise on node features) and vice versa. Here we see that the models are, on average, much more sensitive to noise in the node features (dashed lines) than when noise is progressively added to the edges



no

Figure 4: Visual comparison of PCC values for GCN, GAT, and GT across KLF1, lacI, and PDB-bind datasets.



(a) GAT for KLF1 **(b) GAT for LacI** **(c) GAT for PDBbind**

Figure 5: Correlation plots for best performing models on all three datasets. Dashed line represent ideal correlation

(filled lines), further suggesting that the models may fail to utilize interaction information from edge features and instead rely more heavily on node features.

6.3 Optimized Model Performance

Based on its performance and robustness in the preceding experiments, the GAT architecture was selected for further optimization. A Bayesian hyperparameter search was conducted for learning rate, dropout probability, and the number of attention layers, independently for each dataset. The optimal hyperparameters found are presented in Table 5.

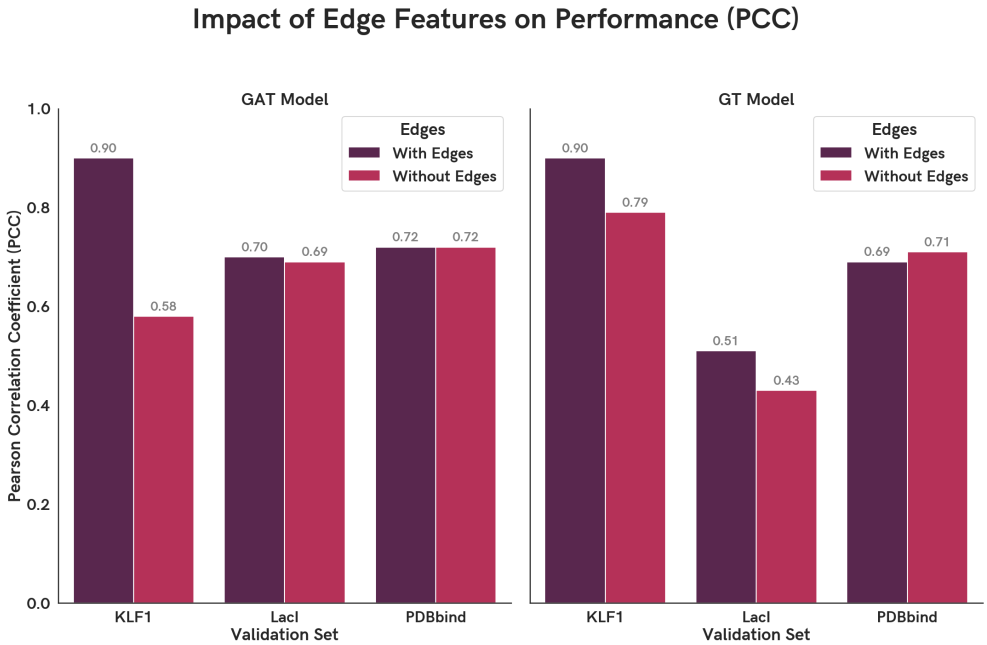


Figure 6: Visual comparison of PCC values for GAT, and GT, with and without edge features across KLF1, and PDBbind datasets.

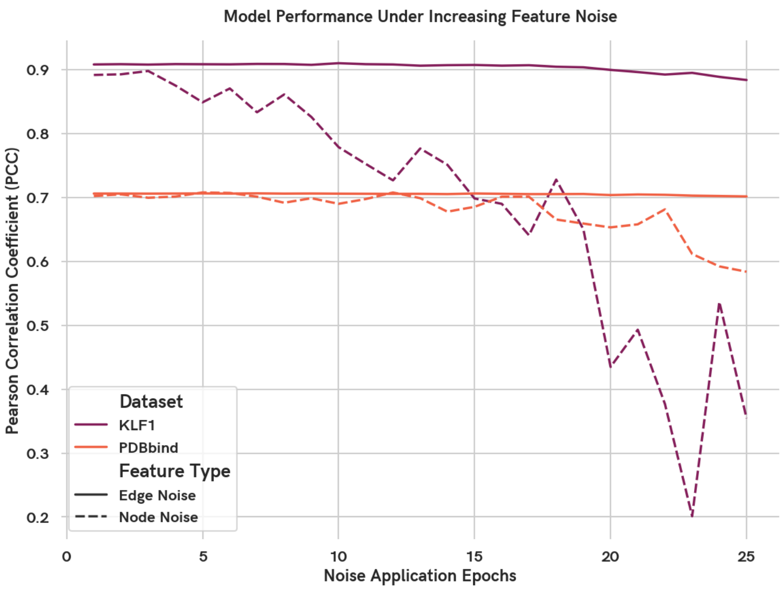
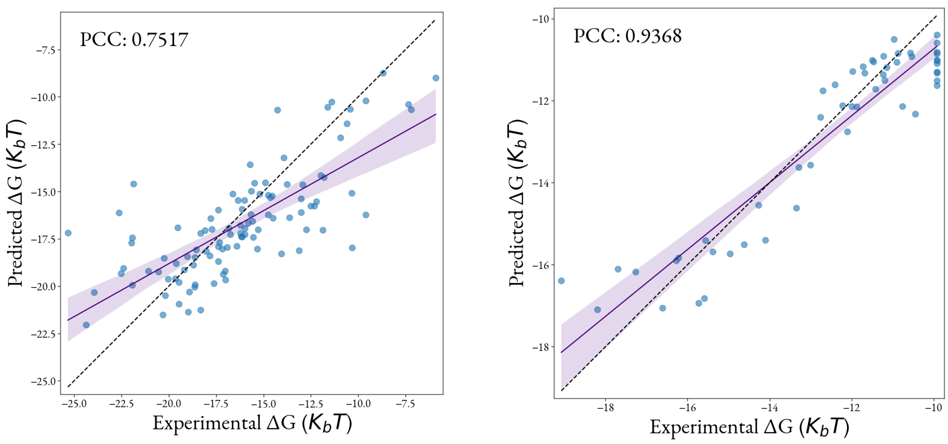


Figure 7: Average PCC when applying additive noise to node features (dashed lines) and edge features (filled lines) over 25 epochs for KLF1 and PDBbind across all trained models.

The performance of the GAT model after hyperparameter optimization is shown in Table 6, compared to its pre-optimization baseline. Optimization led to consistent improvements across both datasets (KLF1 and PDBbind).

Table 5: Optimal hyperparameters for the GAT model identified via Bayesian search.

Hyperparameter	KLF1	PDBbind
Learning Rate	0.032	0.033
Dropout	0.5	0.5
Number of Layers	4	4
Number of Heads	4	3
Embedding size	256	128
Weight Decay	0.0021	0.0021



(a) Optimized GAT for PDBbind

(b) Optimized GAT for KLF1

Figure 8: Correlation plots for sweep optimized models. Dashed line represent ideal correlation

Table 6: Performance of the GAT model before and after hyperparameter optimization.

Dataset	Pre-Optimization		Post-Optimization	
	MSE	PCC	MSE	PCC
KLF1	0.24	0.90	**0.37**	**0.94**
PDBbind	0.58	0.70	**0.48**	**0.75**

6.4 Generalization Across Diverse Datasets

The generalization capabilities of the optimized GAT model were assessed by training on one dataset and evaluating on another, specifically training on PDBbind and validating on the KLF1 dataset and vice versa. Also by training on a combined dataset of PDBbind + KLF1 and then evaluating on the two datasets separatly. The PDBbind-trained model is the best performing from the hyperparameter optimization, and the model with combined datasets was trained with the same hyperparameters.

Figure 9 presents the performance for these cross-dataset generalization tasks. The results show weak generalization capability at this stage. When evaluating the best model trained on the PDBbind data on the KLF1 data, the performance is close to zero. Furthermore, combining datasets shows decreased performance on both validation sets.

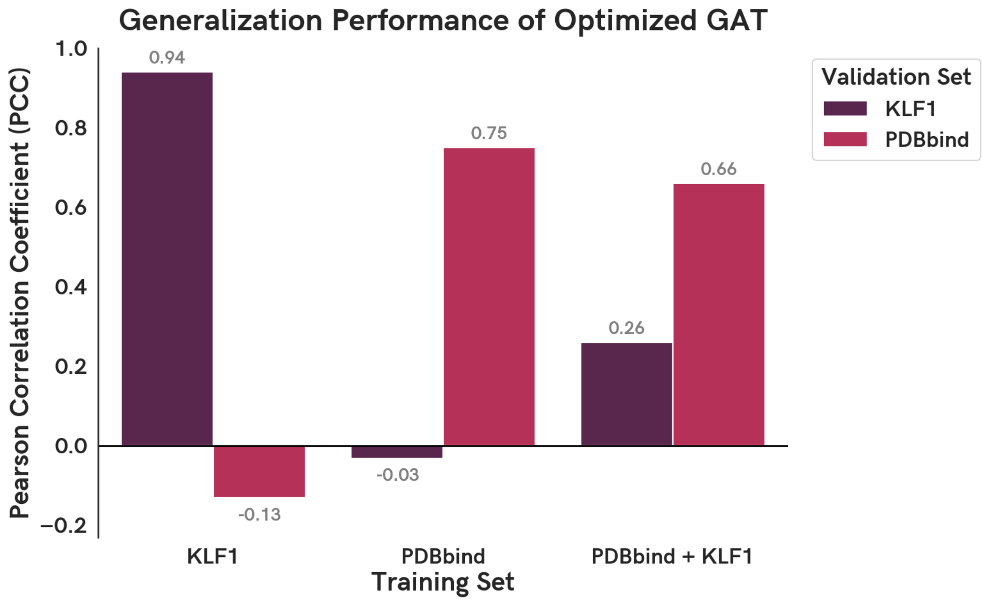


Figure 9: Generalization performance of the optimized GAT model.

7 DISCUSSION

This study investigated the use of AlphaFold embeddings coupled with Graph Neural Networks for predicting protein-DNA binding affinity. Our results provide several insights into the performance of different GNN architectures and the utility of AlphaFold-derived features.

The initial comparison of GNN architectures revealed that attention-based models, specifically GAT and Graph Transformers, outperformed the baseline GCN on the HiTS-FLIP datasets (KLF1 and LacI) in terms of both MSE and PCC (Table 4). This suggests that the attention mechanisms, which allow the models to weigh the importance of different nodes and their interactions, are beneficial for capturing the nuances of binding affinity when protein identity is fixed and DNA sequences vary. However, on the more diverse PDBbind dataset, the simpler GCN model achieved comparable or slightly better performance than the GAT and Graph Transformer models. This could indicate that on datasets with greater heterogeneity in both proteins and DNA, the increased complexity of attention-based models might not readily translate to better performance without more extensive data or fine-tuning, or that the GCN's simpler aggregation scheme provides a more robust baseline across varied structures.

The investigation into feature importance highlighted that node features (AlphaFold's single representation) appear to be more consistently critical for model performance than edge features (AlphaFold's pair representation). While edge features contributed positively to performance on the KLF1 dataset when using GAT or Graph Transformer models (Table 4), the models' higher sensitivity to noise in node features compared to edge features across datasets suggests a potential underutilization or less robust integration of the rich interaction information encoded in the edge features. The GCN model, which by default does not use edge features, performed worst on the KLF1 dataset where edge features seemed to matter more for the other architectures. This implies that while edge features contain valuable information, current GNN architectures may not always extract it optimally for this task across all data contexts.

The generalization experiments, crucial for assessing the broader applicability of the models, were outlined (Section 6.3), but the results show that the models currently fail to generalize well enough to be transferred into new domains. In the case of the model trained only on PDBbind, the lack of generalizability is likely caused by a drastic underrepresentation of the size and structure that the KLF1 complexes exhibit. PDBbind contains a span of a long range of sizes in its dataset, thereby causing a considerable domain shift during evaluation. However, when trained on both datasets, the opposite is likely true, as the KLF1 domain now likely becomes overrepresented and skews the training distribution by diluting other structures in favor of KLF1-like molecules.

Limitations of this study include the observed variability in edge feature utility and the models' sensitivity to node feature noise. The undetermined scale factor for the LacI dataset might also affect direct comparisons or combined dataset training. Further analysis is needed to understand why simpler models sometimes outperform more complex ones on diverse datasets and to improve the generalization capabilities across varied protein-DNA complexes. Exploring different graph construction methods or more sophisticated readout functions might also yield improvements. Furthermore, the AlphaFold predictions also contain uncertainty which is not accounted for. By pruning or weighting the data points based on the provided confidence metrics, also provided in the predictions, there might be considerable performance gains for this approach.

8 CONCLUSION

Predicting protein-DNA binding affinity is a complex but vital task for understanding gene regulation and disease. This work explored the application of AlphaFold embeddings with various Graph Neural Network architectures (GCN, GAT, Graph Transformer) to tackle this challenge.

Our findings demonstrate that GAT and Graph Transformer models can achieve strong predictive performance, particularly on datasets with fixed proteins and varying DNA sequences like KLF1, outperforming a baseline GCN model. However, on the more heterogeneous PDBbind dataset, the GCN model remained competitive. The study indicates that AlphaFold's single representation (node features) provides a strong foundation for affinity prediction, while the utility of the pair representation (edge features) appears to be more context-dependent with current models, suggesting an area for future architectural improvements.

While the current models show promise, further research is necessary to fully utilize the structural information from AlphaFold, particularly concerning edge features, and to ensure robust generalization across the domain of protein-DNA interactions.

9 ETHICAL AND SUSTAINABILITY CONSIDERATIONS

As a purely computational study, this project does not involve human or animal subjects, and therefore the ethical considerations common to clinical or biological research are not directly applicable. The primary ethical and sustainability dimensions of this work relate to data integrity, the responsible application of artificial intelligence, and the consumption of computational resources.

The main ethical responsibility in this research lies in ensuring the integrity and transparency of the scientific findings. The models developed are highly dependent on the quality and composition of the training data sourced from databases like PDBbind and internal HiTS-FLIP experiments. The models risk inheriting biases present in the available data. For example, if certain protein families are overrepresented, the model's predictive accuracy may be artificially high for those families and poor for others. The observed difficulty in generalizing from the PDBbind-trained model to the KLF1 dataset underscores this challenge. It is an ethical imperative for the researcher to transparently report these limitations to prevent the overestimation of the model's capabilities in real-world applications.

The direct environmental footprint of this project is minimal; however, the sustainability implications are primarily linked to the energy consumption of computational resources. Training deep learning models, particularly large Graph Neural Networks and the prerequisite AlphaFold predictions, is computationally intensive and requires significant electrical energy. The hyperparameter optimization phase, which was limited by computational constraints for the LacI dataset, is an example of this high demand. The carbon footprint of this research is therefore tied to the energy efficiency of the data centers used and the carbon intensity of their electricity source.

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude to the many individuals who have supported me throughout this master's thesis project. First and foremost, I wish to thank my supervisor, Emil Marklund. His invaluable guidance, insightful feedback, and encouragement were vital in shaping this work. I am also deeply grateful to Nina Warnke, Axel Truedson, and Nicklas Eckert Elfving. As someone with no prior background in molecular biology, their willingness to share their knowledge and provide clear explanations was fundamental to my understanding of the subject. Their support was crucial in bridging the gap between computer science and life science.

A special thanks also goes to Patrick Bryant for suggesting the novel approach of using AlphaFold embeddings. This key idea formed the very foundation of this thesis.

REFERENCES

[1] [n. d.]. Systematic Localization of Common Disease-Associated Variation in Regulatory DNA | Science. https://www.science.org/doi/10.1126/science.1222794

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. 2024. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* 630, 8016 (June 2024), 493–500. https://doi.org/10.1038/s41586-024-07487-w

[3] Benoit Ballester, Alejandra Medina-Rivera, Dominic Schmidt, Mar Gonzàlez-Porta, Matthew Carlucci, Xiaoting Chen, Kyle Chessman, Andre J Faure, Alister PW Funnell, Angela Goncalves, et al. 2014. Multi-species, multi-transcription factor binding highlights conserved control of tissue-specific biological pathways. *Elife* 3 (2014), e02626.

[4] Pedro J Ballester and John BO Mitchell. 2010. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* 26, 9 (2010), 1169–1175.

[5] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks? https://doi.org/10.48550/arXiv.2105.14491 arXiv:2105.14491 [cs].

[6] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[7] Torbjörn Fries. [n. d.]. Deep learning model for prediction of protein-DNA binding rates and affinities. ([n. d.]).

[8] William L Hamilton. [n. d.]. Graph Representation Learning. ([n. d.]).

[9] K. Harini, Masakazu Sekijima, and M. Michael Gromiha. 2025. Bioinformatics Approaches for Understanding the Binding Affinity of Protein–Nucleic Acid Complexes. In *Prediction of Protein Secondary Structure*, Andrzej Kloczkowski, Lukasz Kurgan, and Eshel Faraggi (Eds.). Springer US, New York, NY, 315–330. 978-1-0716-4196-5 https://doi.org/10.1007/978-1-0716-4196-5_18

[10] Cyriac Kandoth, Michael D. McLellan, Fabio Vandin, Kai Ye, Beifang Niu, Charles Lu, Mingchao Xie, Qunyuan Zhang, Joshua F. McMichael, Matthew A. Wyczalkowski, Mark D. M. Leiserson, Christopher A. Miller, John S. Welch, Matthew J.

Walter, Michael C. Wendl, Timothy J. Ley, Richard K. Wilson, Benjamin J. Raphael, and Li Ding. 2013. Mutational landscape and significance across 12 major cancer types. *Nature* 502, 7471 (Oct. 2013), 333–339. https://doi.org/10.1038/nature12634 Publisher: Nature Publishing Group.

[11] Yuxi Ke, Eesha Sharma, Hannah K. Wayment-Steele, Winston R. Becker, Anthony Ho, Emil Marklund, and William J. Greenleaf. 2024. High-Throughput DNA melt measurements enable improved models of DNA folding thermodynamics. https://doi.org/10.1101/2024.01.08.574731 Pages: 2024.01.08.574731 Section: New Results.

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. https://doi.org/10.48550/arXiv.1609.02907 arXiv:1609.02907 [cs].

[13] Xuefeng Liu, Songhao Jiang, Xiaotian Duan, Archit Vasan, Chong Liu, Chih-chan Tien, Heng Ma, Thomas Brettin, Fangfang Xia, Ian T. Foster, and Rick L. Stevens. 2024. Binding Affinity Prediction: From Conventional to Machine Learning-Based Approaches. https://doi.org/10.48550/arXiv.2410.00709 arXiv:2410.00709 [q-bio].

[14] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. 2015. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* 31, 3 (2015), 405–412.

[15] Harvey F Lodish. 2008. *Molecular cell biology*. Macmillan.

[16] Emil Marklund, Yuxi Ke, and William J. Greenleaf. 2023. High-throughput biochemistry in RNA sequence space: predicting structure and function. *Nature Reviews Genetics* 24, 6 (June 2023), 401–414. https://doi.org/10.1038/s41576-022-00567-5 Publisher: Nature Publishing Group.

[17] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. 2016. Deep learning in bioinformatics. *Briefings in Bioinformatics* (July 2016), bbw068. https://doi.org/10.1093/bib/bbw068

[18] Razvan Nutiu, Robin Friedman, Shujun Luo, Irina Khrebtukova, David Silva, Robin Li, Gary Schroth, and Christopher Burge. 2011. Direct measurement of DNA affinity landscapes on a high-throughput sequencing instrument. *Nature biotechnology* 29 (June 2011), 659–64. https://doi.org/10.1038/nbt.1882

[19] Simon J D Prince. [n. d.]. Understanding Deep Learning. ([n. d.]).

[20] Prawin Rimal, Shamrat Kumar Paul, Shailesh Kumar Panday, and Emil Alexov. 2025. Further Development of SAMPDI-3D: A Machine Learning Method for Predicting Binding Free Energy Changes Caused by Mutations in Either Protein or DNA. *Genes* 16, 1 (Jan. 2025), 101. https://doi.org/10.3390/genes16010101 Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. https://arxiv.org/abs/1710.10903v3

[22] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2020. Graph Transformer Networks. https://doi.org/10.48550/arXiv.1911.06455 arXiv:1911.06455 [cs].

[23] Zhengrong Zhu and John Cuozzo. 2009. High-Throughput Affinity-Based Technologies for Small-Molecule Drug Discovery. *SLAS Discovery* 14, 10 (Dec. 2009), 1157–1164. https://doi.org/10.1177/1087057109350114

TRITA-EECS-EX-2025:474
Stockholm, Sweden 2025

www.kth.se