Protein-Nucleic Acid Complex Modeling with Frame Averaging Transformer

Tinglin Huang[1*] **Zhenqiao Song**[2] **Rex Ying**[1] **Wengong Jin**[3,4]

[1]Yale University, [2]Carnegie Mellon University,
[3]Northeastern University, Khoury College of Computer Sciences
[4]Broad Institute of MIT and Harvard

Abstract

Nucleic acid-based drugs like aptamers have recently demonstrated great therapeutic potential. However, experimental platforms for aptamer screening are costly, and the scarcity of labeled data presents a challenge for supervised methods to learn protein-aptamer binding. To this end, we develop an unsupervised learning approach based on the predicted pairwise contact map between a protein and a nucleic acid and demonstrate its effectiveness in protein-aptamer binding prediction. Our model is based on FAFormer[2], a novel equivariant transformer architecture that seamlessly integrates frame averaging (FA) within each transformer block. This integration allows our model to infuse geometric information into node features while preserving the spatial semantics of coordinates, leading to greater expressive power than standard FA models. Our results show that FAFormer outperforms existing equivariant models in contact map prediction across three protein complex datasets, with over 10% relative improvement. Moreover, we curate five real-world protein-aptamer interaction datasets and show that the contact map predicted by FAFormer serves as a strong binding indicator for aptamer screening.

1 Introduction

Nucleic acids have recently shown significant potential in drug discovery, as shown by the success of mRNA vaccines [26, 61, 60] and aptamers [15, 32, 14, 41]. Aptamers are single-stranded nucleic acids capable of binding to a wide range of molecules, including previously undruggable targets [20, 11]. Currently, aptamer discovery is driven by high-throughput screening, which is time-consuming and labor-intensive. While machine learning can potentially accelerate this process, the limited availability of labeled data presents a significant challenge in ML-guided aptamer discovery [57, 44, 16]. Given this challenge, our goal is to build an unsupervised protein-nucleic acid interaction predictor for large-scale aptamer screening.

Motivated by previous work on unsupervised protein-protein interaction prediction [34], we focus on predicting the contact map between proteins and nucleic acids at the residue/nucleotide level. The main idea is that a predicted contact map offers insights into the likelihood of a protein forming a complex with an aptamer, thereby encoding the binding affinity between them. Concretely, as shown in Figure 1(a), our model is trained to identify specific contact pairs between residues and nucleotides when forming a complex. The maximum contact probability across all pairs is then interpreted as the binding affinity, which is subsequently used for aptamer screening.

*Correspondence to tinglin.huang@yale.edu

[2]https://github.com/Graph-and-Geometric-Learning/Frame-Averaging-Transformer

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-4038

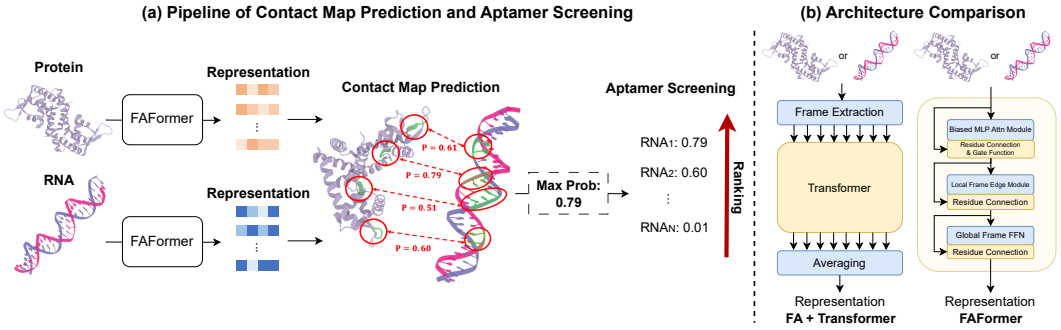


Figure 1: **(a)** The pipeline of contact map prediction between protein and nucleic acid, and applying the predicted results for screening in an unsupervised manner. The affinity score is quantified as the maximum contact probability over all pairs. **(b)** Comparison between Transformer with vanilla frame averaging framework and FAFormer, where the blue cells indicate FA-related modules.

One key factor contributing to the accuracy of contact map prediction models is their capacity to learn equivariant transformations for symmetry groups [67, 39, 66, 74, 52]. A novel line of research focuses on adapting Transformer [83] to equivariant frameworks, leveraging its great expressive power. However, these studies have encountered issues with either (1) high computational overhead with spherical harmonics-based models [47, 27], which complicate encoding by introducing irreducible representations; or (2) limited expressive capability with frame averaging (FA) [63], which diminishes geo-information exploitation by simply concatenating coordinates with node representations.

In light of this, we propose **FAFormer**, an equivariant Transformer architecture that integrates FA as a geometric module within each layer. FA as a geometric component offers flexibility to effectively integrate geometric information into node representations while preserving the spatial semantics of coordinates, eliminating the need for complex geometric feature extraction. FAFormer consists of a *Local Frame Edge Module* that embeds local pairwise interactions between each node and its neighbors; a *Biased MLP Attention Module* that integrates relational bias from edge representation within MLP attention and equivariantly updates coordinates; and a *Global Frame FFN* that integrates geometric features into node representations within the global context.

To validate the advantage of our model architecture, we evaluate FAFormer on two tasks: (1) protein-nucleic acid contact prediction and (2) unsupervised aptamer virtual screening. In the first task, our model consistently surpasses state-of-the-art equivariant models with over 10% relative improvement across three protein complex datasets. For the second task, we collected five real-world protein-aptamer interaction datasets with experimental binding labels. Our results show that FAFormer, trained on the contact map prediction task, is an effective binding indicator for aptamer screening. Compared to RoseTTAFoldNA, a large pretrained model for complex structure prediction, FAFormer achieves comparable performance on contact map prediction and better results on aptamer screening, while offering 20-30x speedup.

2 Related Work

Aptamer screening Aptamers are single-stranded RNA or DNA oligonucleotides that can bind various molecules with high affinity and specificity [14, 15, 32, 41]. SELEX (Systematic Evolution of Ligands by EXponential Enrichment) is a conventional technique used for high-throughput screening of aptamers [23, 72, 23], which iteratively selects and amplifies target-bound sequences. Despite its effectiveness, SELEX needs to take substantial time to identify a small number of aptamers [73, 51]. There are some recent studies applying machine learning techniques to aptamer research [18], including generating aptamer structures [36], optimizing SELEX protocols [6], and predicting and modeling protein-aptamer interactions [24, 46, 68, 57]. However, these methods require the user to provide labeled data from time-consuming SELEX assays, thus do not apply to new targets without any SELEX data. Our work focuses on predicting the contact map between protein and nucleic acids using 3D structures and conducting unsupervised screening based on the predicted contact maps, which has not yet been thoroughly explored.

https://doi.org/10.52202/079017-4038

Protein complex modeling The prediction and understanding of interactions between proteins and molecules play a crucial role in biomedicine. For example, some prior studies focus on developing a geometric learning method to predict the conformation of a small molecule when it binds to a protein target [71, 19, 53, 50]. As for the protein-protein complex, [25, 29] explore the application of machine learning in predicting the structure of protein multimer. Some studies [77, 56] investigate the protein-protein interface prediction in the physical space. Jin et al. [38] studies the protein-protein affinity prediction in an unsupervised manner. For protein-nucleic acid, some prior works explore the identification of the nucleic-acid-binding residues on protein [65, 86, 89, 37, 85] or predict the binding probability of RNA on proteins [79, 82, 87, 54]. Some previous studies focus on modeling protein-nucleic acid complex by computational method [81, 80, 28]. AlphaFold3 [1] and RoseTTAFoldNA [5] are the recent progresses in this field, which both are pretrained language models for complex structure prediction.

Geometric deep learning Recently, geometric deep learning achieved great success in chemistry, biology, and physics domains [13, 90, 40, 55, 52, 64, 10, 70]. The previous methods roughly fall into four categories: 1) Invariant methods extract invariant geometric features from the molecules, such as pairwise distance and torsion angles, to exhibit invariant transformations [67, 31, 30]; 2) Spherical harmonics-based models leverage the functions derived from spherical harmonics and irreducible representations to transform data equivariantly [27, 47, 75]; 3) Some methods encode the coordinates and node features in separate branches, interacting these features through the norm of coordinates [66, 39]; 4) Frame averaging (FA) [63, 21] framework proposes to model the coordinates in eight different frames extracted by PCA, achieving equivariance by averaging the encoded representations.

The proposed FAFormer combines the strengths of FA and the third category of methods by encoding and interacting coordinates with node features using FA-based components. Besides, FAFormer can also be viewed as a combination of GNN and Transformer architectures, as the edge representation calculation functions similarly to message passing in GNN. This integration is made possible by the flexibility provided by FA as an integrated component.

3 Frame Averaging Transformer

In this section, we present our proposed FAFormer, a frame averaging (FA)-based transformer architecture. We first introduce the FA framework in Section 3.1 and elaborate on the proposed FAFormer in Section 3.2. Discussion on the equivariance is provided in Section 3.3, and the computational complexity analysis can be found in Appendix B.

3.1 Background: Frame Averaging

Frame averaging (FA) [63] is an encoder-agnostic framework that can make a given encoder equivariant to the Euclidean symmetry group. FA applies the principle components derived via Principal Component Analysis (PCA) to construct the frame capable of achieving $E(3)$ equivariance. Specifically, the frame function $\mathcal{F}(\cdot)$ maps a given set of coordinates $\boldsymbol{X}$ to eight transformations:

$$\mathcal{F}(\boldsymbol{X}) = \{(\boldsymbol{U}, \boldsymbol{c}) | \boldsymbol{U} = [\alpha_1 \boldsymbol{u}_1, \alpha_2 \boldsymbol{u}_2, \alpha_3 \boldsymbol{u}_3], \alpha_i \in \{-1, 1\}\} \tag{1}$$

where $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ are the three principal components of $\boldsymbol{X}$, $\boldsymbol{U} \in \mathbb{R}^{3\times3}$ denotes the rotation matrix based on the principal components, and $\boldsymbol{c} \in \mathbb{R}^3$ is the centroid of $\boldsymbol{X}$. The main idea of FA is to encode the coordinates as projected by the transformations, followed by averaging these representations. We introduce $f_{\mathcal{F}}(\cdot)$ to represent the projections of given coordinates via $\mathcal{F}(\cdot)$:

$$\begin{aligned} f_{\mathcal{F}}(\boldsymbol{X}) &:= \{(\boldsymbol{X} - \boldsymbol{c})\boldsymbol{U} \mid (\boldsymbol{U}, \boldsymbol{c}) \in \mathcal{F}(\boldsymbol{X})\} \\ &:= \{\boldsymbol{X}^{(g)}\}_{\mathcal{F}} \end{aligned} \tag{2}$$

where $\boldsymbol{X}^{(g)}$ denotes the coordinates transformed by g-th transformations. We can apply any encoder $\Phi(\cdot)$ to the projected coordinates and achieve equivariance by averaging, which can be formulated as an inverse mapping $f_{\mathcal{F}^{-1}}(\cdot)$:

$$f_{\mathcal{F}^{-1}}\left(\{\Phi(\boldsymbol{X}^{(g)})\}_{\mathcal{F}}\right) := \frac{1}{|\mathcal{F}(\boldsymbol{X})|} \sum_g \Phi(\boldsymbol{X}^{(g)}) \boldsymbol{U}_g^{-1} + \boldsymbol{c} \tag{3}$$

https://doi.org/10.52202/079017-4038

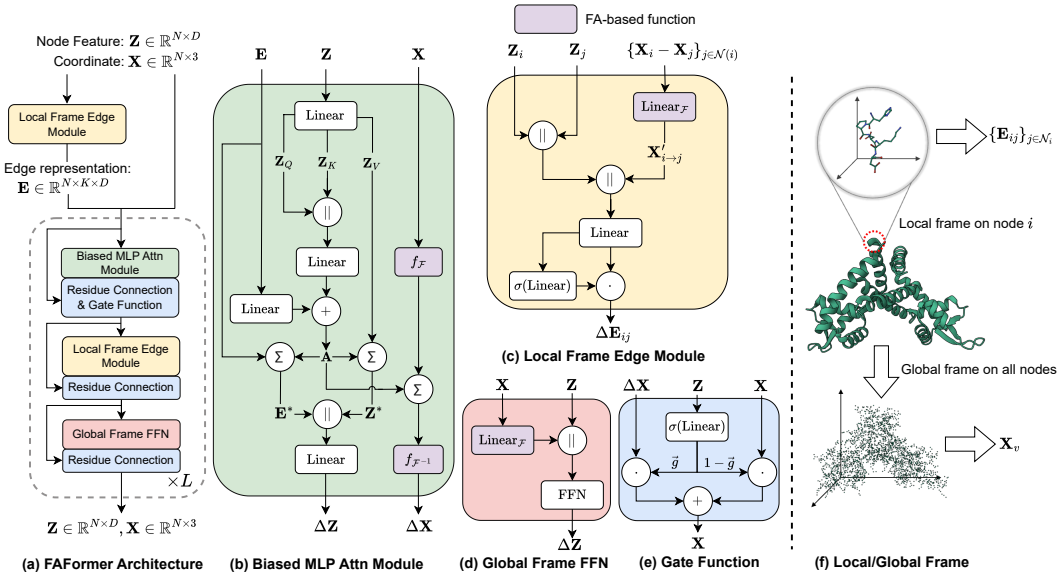


Figure 2: Overview of FAFormer architecture. The input consists of the node features, coordinates, and edge representations, which are processed by a stack of **(b)** Biased MLP Attention Module, **(c)** Local Frame Edge Module, **(d)** Global Frame FFN, and **(e)** Gate Function. $\sum$ deontes aggregation, $\cdot$ is multiplication, $+$ is addition, and $||$ indicates concatenation. Purple cells indicate the operation related to FA. **(f)** illustrates the difference between the local and global frames, where the local frame captures local interactions among the immediate neighbors for each node, while the global frame captures long-range correlations among all nodes.

where $\boldsymbol{U}_g^{-1}$ is the inverse matrix of g-th transformations and the result exhibit $E(3)$ equivariance. The outcome is invariant when simply averaging the representations without inverse matrix.

3.2 Model Architecture

Instead of serving FA as an external encoder wrapper, we propose instantiating FA as an integral geometric component within Transformer. This integration preserves equivariance and enables the model to encode coordinates effectively in the latent space, ensuring compatibility with Transformer architecture. The overall architecture is illustrated in Figure 2.

Graph Construction Each molecule can be naturally represented as a graph [47, 35, 43] where the residues/nucleic acids are the nodes and the interactions between them represent the edges. To efficiently model the macro-molecules (i.e., protein and nucleic acid), we restrict the attention for each node i to its K-nearest neighbors $\mathcal{N}_{\text{top-}K}(i)$, within a predetermined distance cutoff c:

$$\mathcal{N}(i) = \{j | d_{ij} \leq c \text{ and } j \in \mathcal{N}_{\text{top-}K}(i)\} \tag{4}$$

where we use $\mathcal{N}(i)$ to denote the valid neighbors of node i, and d_{ij} denotes the distance between node i and j. The long-range context information can be captured by iterative attention within the local neighbors for each node i.

Overall Architecture As shown in Figure 2(a), the input of FAFormer comprises the node features $\boldsymbol{Z} \in \mathbb{R}^{N\times D}$, coordinates $\boldsymbol{X} \in \mathbb{R}^{N\times 3}$, and edge representations $\boldsymbol{E} \in \mathbb{R}^{N\times K\times D}$ derived by our proposed edge module, where N is the number nodes and D is the hidden size. FAFormer processes and updates the input features at each layer:

$$\boldsymbol{Z}^{(l+1)}, \boldsymbol{X}^{(l+1)}, \boldsymbol{E}^{(l+1)} = f^{(l)}(\boldsymbol{Z}^{(l)}, \boldsymbol{X}^{(l)}, \boldsymbol{E}^{(l)}) \tag{5}$$

where $f^{(l)}(\cdot)$ represents l-th layer of FAFormer. In each model layer, we first update coordinates and node features using the Biased MLP Attention Module. Then, these features are fed into Local Frame Edge Module to refine the edge representations. Finally, the node representations undergo further updates through Global Frame FFN.

FA Linear Module Based on FA, we generalize the vanilla linear module to encode coordinates in the latent space invariantly:

$$\text{Linear}_{\mathcal{F}}(\boldsymbol{X}) := \frac{1}{|\mathcal{F}(\boldsymbol{X})|}\sum_{g}\text{Norm}(\boldsymbol{X}^{(g)})\boldsymbol{W}_g \tag{6}$$

where $\{\boldsymbol{X}^{(g)}\}_{\mathcal{F}}$ is obtained using Equ.(2), $\boldsymbol{W}_g \in \mathbb{R}^{3\times D}$ is a learnable matrix for g-th transformations, and $\text{Norm}(\boldsymbol{X}) := \boldsymbol{X}/\sqrt{\frac{1}{\nu}||\boldsymbol{X}||_2^2}$ is the normalization which scales the coordinates such that their root-mean-square norm is one [39]. $\text{Linear}_{\mathcal{F}}(\cdot)$ is an invariant transformation and will serve as a building block within each layer of the model.

Local Frame Edge Module We explicitly embed the interactions between each node and its neighbors as the edge representation $\boldsymbol{E} \in \mathbb{R}^{N\times K\times D}$, where K is the number of the neighbors. It encodes the relational information and represents the bond interaction between nodes, which is critical in understanding the conformation of molecules [40, 4, 35].

As shown in Figure 2(f), unlike the vanilla FA which *globally* encodes the geometric context of the entire molecule, the edge module builds frame *locally* around each node's neighbors. Specifically, given a node i and its neighbor $j \in \mathcal{N}(i)$, the geometric context is encoded within the local neighborhood:

$$\{\boldsymbol{X}'_{i\to j}\}_{\mathcal{N}(i)} = \text{Linear}_{\mathcal{F}}\left(\{\boldsymbol{X}_i - \boldsymbol{X}_j\}_{\mathcal{N}(i)}\right) \tag{7}$$

where $\{\boldsymbol{X}_i - \boldsymbol{X}_j\}_{\mathcal{N}(i)}$ denotes the direction vectors from center node i to its neighbors, and $\boldsymbol{X}'_{i\to j} \in \mathbb{R}^d$ is the encoded representation. With the local frame, the spatial information sent from one source node depends on the target node, which is compatible with the attention mechanism. Then the node features are engaged with geometric features, and the edge representation is finalized through the residual connection with the gate mechanism:

$$\boldsymbol{m}_{ij} = \text{Linear}(\boldsymbol{Z}_i||\boldsymbol{Z}_j||\boldsymbol{X}'_{i\to j}) \quad \text{and} \quad \boldsymbol{E}'_{ij} = \boldsymbol{g}_{ij}\cdot\boldsymbol{m}_{ij} + \boldsymbol{E}_{ij} \tag{8}$$

where $(\cdot||\cdot)$ is the concatenation operation, and the calculated gate $\boldsymbol{g}_{ij} = \sigma(\text{Linear}(\boldsymbol{m}_{ij}))$ provides flexibility in regulating the impact of updated edge representation.

The encoded edge representation in FAFormer plays a crucial role in modeling the pairwise relationships between nodes, especially for nucleic acid due to their specific base pairing rules [58, 45]. The incorporation of FA facilitates the encoding of the pairwise relationships in a geometric context, resulting in an expressive representation.

Biased MLP Attention Module As shown in Figure 2(b), the attention module of FAFormer first transforms the node features $\boldsymbol{Z}$ into query, key, and value representations:

$$\boldsymbol{Z}_Q = \boldsymbol{Z}\boldsymbol{W}_Q,\ \boldsymbol{Z}_K = \boldsymbol{Z}\boldsymbol{W}_K,\ \boldsymbol{Z}_V = \boldsymbol{Z}\boldsymbol{W}_V \tag{9}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{D\times D}$ are the learnable projections. We adopt MLP attention [12] to derive the attention weight between node pairs, which can effectively capture any attention pattern. The relational information from the edge representation is integrated as an additional bias term:

$$a_{ij} = \text{Softmax}_i\left(\text{Linear}(\boldsymbol{Z}_{Q,i}||\boldsymbol{Z}_{K,j}) + b_{ij}\right), \tag{10}$$

where $b_{ij} = \text{Linear}(\text{LN}(\boldsymbol{E}_{ij}))$ represents the scalar bias term based on the edge representation, a_{ij} denotes the attention score between i-th and j-th nodes, $\boldsymbol{Z}_{*,i}$ is i-th representation of the matrix $\boldsymbol{Z}_*$, $\text{Softmax}_i(\cdot)$ is the softmax function operated on the attention scores of node i's neighbors, and $\text{LN}(\cdot)$ is layernorm function [3].

Besides the value embeddings, the edge representation will also be aggregated to serve as the context for the update of node feature in FAFormer:

$$\boldsymbol{Z}^*_i = \sum_{j\in\mathcal{N}(i)} a_{ij}\boldsymbol{Z}_j, \boldsymbol{E}^*_i = \sum_{j\in\mathcal{N}(i)} a_{ij}\boldsymbol{E}_{ij}, \tag{11}$$

$$\boldsymbol{Z}'_i = \text{LN}\left(\text{Linear}(\boldsymbol{Z}^*_i||\boldsymbol{E}^*_i)\right) + \boldsymbol{Z}_i \tag{12}$$

https://doi.org/10.52202/079017-4038

where $\boldsymbol{Z}_i'$ is the update representation of node i. The above attention can be extended to a multi-head fashion by performing multiple parallel attention functions. For the coordinates, we employ an equivariant aggregation function that supports multi-head attention:

$$\boldsymbol{X}^* = f_{\mathcal{F}^{-1}}\left(\{[\boldsymbol{A}^{(0)}\boldsymbol{X}^{(g)}, \cdots, \boldsymbol{A}^{(H)}\boldsymbol{X}^{(g)}]\boldsymbol{W}\}_{\mathcal{F}}\right) \tag{13}$$

where $\{\boldsymbol{X}^{(g)}\}_{\mathcal{F}} = f_{\mathcal{F}}(\boldsymbol{X})$, H is the number of attention heads, $[\cdot]$ is the tensor stack operation and $\boldsymbol{W} \in \mathbb{R}^{H\times 1}$ is a linear transformation for aggregating coordinates in different heads. An additional gate function that uses node representations as input to modulate the aggregation is applied:

$$\boldsymbol{X}' = \boldsymbol{g}_{\text{attn}} \cdot \boldsymbol{X}^* + (1 - \boldsymbol{g}_{\text{attn}}) \cdot \boldsymbol{X} \tag{14}$$

where $\boldsymbol{g}_{\text{attn}} = \sigma(\text{Linear}(\boldsymbol{Z}))$ is the vector-wise gate designed to modulate the integration between the aggregated and the original coordinates. This introduced gate mechanism further encourages the communication between node features and geometric features.

Global Frame FFN To further exploit the interaction between node features and coordinates, we extend the conventional FFN to *Global Frame* FFN which integrates spatial locations with node features through FA, which is illustrated in Figure 2(d):

$$\boldsymbol{X}_v = \text{Linear}_{\mathcal{F}}(\boldsymbol{X}) \quad \text{and} \quad \boldsymbol{Z}' = \text{FFN}(\boldsymbol{Z}||\boldsymbol{X}_v) + \boldsymbol{Z} \tag{15}$$

where $\text{FFN}(\cdot)$ denotes a two-layer fully connected feed-forward network. This integration of spatial information $\boldsymbol{X}_v$ into the feature vectors enables the self-attention mechanism to operate in a geometric-aware manner. Unlike the edge module that focuses on each node's local neighbors, global frame FFN encodes the coordinates of all nodes, thereby capturing the long-range correlation among nodes.

3.3 Equivariance

The function $\text{Linear}_{\mathcal{F}}(\cdot)$ exhibits invariance since results are simply averaged across different transformations. In light of this, the node representation generated by the edge module and FFN are also invariant. The biased attention is based on the scalar features so the output is always invariant.

The update of coordinates within FAFormer leverages a multi-head attention aggregation with a gate function. Both functions are $E(3)$-equivariant: attention aggregation is based on frame averaging, while gate function is linear and also exhibits $E(3)$-equivariance, with a formal proof in Appendix C.

In conclusion, FAFormer is symmetry-aware which exhibits invariance for node representations and $E(3)$-equivariance for coordinates.

4 Experiments

In this section, we present three protein complex datasets and five aptamer datasets to explore protein complex interactions and evaluate the effectiveness of FAFormer. More details regarding the experiments and datasets can be found in Appendix A and D. Additional experiments, including the ablation studies, and the comparison with AlphaFold3, can be found in Appendix E. All the datasets used in this study are included in our anonymous repository.

4.1 Dataset

Protein Complexes We cleaned up and constructed three 3D structure datasets of protein complexes from multiple sources [8, 7, 2, 77]. A residue-nucleotide pair is determined to be in contact if any of their atoms are within 6Å from each other [77, 78]. We conduct dataset splitting based on the protein sequence identity, using a threshold of 50% for protein-RNA/DNA complexes[3] and 30% for protein-protein complexes. The details of all datasets are shown in Table 1.

The protein's and nucleic acid's structures in the validation/test sets are generated by ESMFold [48] (proteins) or RoseTTAFoldNA (nucleic acids). This offers a more realistic scenario, given that the crystal structures are often unavailable.

[3]Note that we don't use 30% as the threshold since it results in a very limited validation and test set.

Table 1: Protein complex dataset statistics.

	#Train	#Val	#Test	Label
Protein-RNA	1,009	115	118	1.517%
Protein-DNA	2,590	134	134	1.215%
Protein-Protein	4,402	544	545	0.469%

Table 2: Aptamer dataset statistics.

Target	GFP	NELF	HNRNPC	CHK2	UBLCP1
#Positive	520	797	233	1,255	892
#Candidate	1,875	9,833	3,328	10,000	10,000

Aptamers Our aptamer datasets come from the previous studies [76, 46, 73], including five protein targets and their corresponding aptamer candidates. The affinity of each candidate to the target is experimentally determined. Each dataset is equally split into validation and test sets.

We construct the protein-RNA training set by excluding complexes from our collected dataset with over 30% protein sequence identity to these protein targets, resulting in 1,238 training cases. The 3D structures of proteins are obtained from AlphaFold Database [5], and the structures of RNAs are generated by RoseTTAFoldNA without using MSAs. The statistics are presented in Table 2.

Feature The coordinates of the C_α atoms from residue and the C_3 atoms from nucleotide are used as coordinate features. For node feature generation, we employ ESM2 [49] for proteins and RNA-FM [17] for RNA. The one-hot embedding is utilized as DNA's node feature.

4.2 Contact Map Prediction

As shown in Figure 1(a), this task aims to predict the exact contact pairs between protein $\{S_i\}_N$ and nucleic acid $\{S'_j\}_{N'}$ which conducts binary classification over all pairs:

$$\text{Model}(S_i, S'_j) = \begin{cases} 1, & S_i \text{ contacts with } S'_j \\ 0, & \text{Other} \end{cases} \tag{16}$$

Baselines We compare FAFormer with four classes of methods: 1) Vanilla Transformer [83] which doesn't utilize 3D structure; 2) Spherical harmonics-based models Equiformer [47] and SE(3)Transformer [27]; 3) GNN-based models EGNN [66] and GVP-GNN [39]; 4) Transformer with FA [63]. The protein and nucleic acid will be separately encoded with two encoders to avoid label leakage. The representations of residues and nucleotides are concatenated from all pairs and fed into a MLP classifier to conduct prediction.

Results To comprehensively evaluate the performance of label-imbalanced datasets, we apply F1 and PRAUC as the evaluation metrics. The comparison results are presented in Table 3 from which FAFormer reaches the best performance over all the baselines with a relative improvement over 10%.

Additionally, some geometric methods fail to outperform the vanilla Transformer in certain cases. We attribute this to overfitting on crystal structures during training, which hinders their ability to generalize well to unbound structures during evaluation. Compared with serving FA as an external equivariant framework on Transformer, the performance gain on FAFormer verifies the effectiveness of embedding FA as a geometric component within Transformer.

Table 3: Comparison results on three datasets of contact map prediction task.

	Metric	Transformer	SE(3)Transformer	Equiformer	EGNN	GVP-GNN	FA	FAFormer
Protein-RNA	F1	$0.1021_{.007}$	$0.0816_{.001}$	$0.0990_{.005}$	$0.1093_{.004}$	$0.1091_{.010}$	$\underline{0.1150}_{.003}$	$\mathbf{0.1284}_{.003}$
	PRAUC	$\underline{0.1015}_{.002}$	$0.0881_{.001}$	$0.0985_{.004}$	$0.0964_{.002}$	$0.1008_{.004}$	$0.0965_{.005}$	$\mathbf{0.1113}_{.004}$
Protein-DNA	F1	$0.0963_{.006}$	$0.0824_{.015}$	$0.0925_{.010}$	$0.1208_{.009}$	$0.1225_{.006}$	$\underline{0.1283}_{.001}$	$\mathbf{0.1457}_{.005}$
	PRAUC	$0.1111_{.002}$	$0.1022_{.007}$	$0.0913_{.002}$	$0.1139_{.010}$	$\underline{0.1195}_{.006}$	$0.1092_{.006}$	$\mathbf{0.1279}_{.006}$
Protein-Protein	F1	$0.0756_{.004}$	$0.1147_{.011}$	$0.1039_{.002}$	$\underline{0.1461}_{.001}$	$0.1302_{.001}$	$0.1011_{.002}$	$\mathbf{0.1596}_{.002}$
	PRAUC	$0.0707_{.002}$	$0.0906_{.008}$	$0.0834_{.002}$	$\underline{0.1245}_{.001}$	$0.1181_{.002}$	$0.0830_{.001}$	$\mathbf{0.1463}_{.003}$

4.3 Binding Site Prediction

In addition to contact map prediction, we examine the nucleic acid binding site prediction task, which is a node-level task, to comprehensively evaluate our model. This task solely takes a protein $\{S_i\}_N$

https://doi.org/10.52202/079017-4038

as input and aims to identify the nucleic-acid-binding residues on the protein:

$$\text{Model}(S_i) = \begin{cases} 1, & S_i \text{ contacts with nucleic acid} \\ 0, & \text{Other} \end{cases} \tag{17}$$

Predicting the nucleic acid binding site offers promising therapeutic potential for undruggable targets by conventional small molecule drug [33, 9, 88], expanding the range of potential therapeutic targets.

Baselines We compare FAFormer with two state-of-the-art geometric deep learning models: GraphBind [86] and GraphSite [89] in this task. The protein structure is embedded with the geometric encoder and the residue's representations are fed into a classifier for prediction.

Results F1 and PRAUC are applied as the evaluation metrics and we report the average results over three different seeds in Table 4. We can observe that FAFormer achieves the best performance over all the baselines, demonstrating the effectiveness of FAFormer on nucleic acid-related tasks and modeling the geometric 3D structure.

Table 4: Comparison results on binding site prediction.

	Metric	GraphBind	GraphSite	FAFormer
Protein-DNA	F1	$0.492_{.004}$	$0.416_{.007}$	$\mathbf{0.506_{.005}}$
	PRAUC	$0.520_{.003}$	$0.541_{.001}$	$\mathbf{0.549_{.004}}$
Protein-RNA	F1	$0.449_{.004}$	$0.400_{.007}$	$\mathbf{0.472_{.003}}$
	PRAUC	$0.471_{.005}$	$0.479_{.001}$	$\mathbf{0.507_{.004}}$

4.4 Unsupervised Aptamer Screening

This task aims to screen the positive aptamers from a large number of candidates for a given protein target. We quantify the binding affinities between RNA and the protein target as the highest contact probability among the residue-nucleotide pairs. The main idea is that two molecules with a high probability of contact are very likely to form a complex [34]. The models are first trained on the protein-RNA complexes training set using the contact map prediction, then the aptamer candidates are ranked based on the calculated highest contact probabilities.

Results Top-10 precision, Top-50 precision, and PRAUC are used as the metrics. As shown in Table 5, the geometric encoders outperform sequence-based Transformer in most cases, and FAFormer generally reaches the best performance. This demonstrates the great potential of an accurate interaction predictor in determining unsupervisedly promising aptamers.

Table 5: Comparison results of zero-shot aptamer screening.

	Metric	Transformer	SE(3)Transformer	Equiformer	EGNN	GVP-GNN	FA	FAFormer
GFP	Top10 Prec.	$0.2333_{.094}$	$0.2000_{.141}$	$\underline{0.3000}_{.000}$	$0.2666_{.047}$	$0.3000_{.081}$	$0.3000_{.141}$	$\mathbf{0.4000_{.078}}$
	Top50 Prec.	$0.2733_{.073}$	$0.2666_{.061}$	$0.3666_{.024}$	$0.3400_{.081}$	$\underline{0.3799}_{.082}$	$0.3333_{.033}$	$\mathbf{0.4133_{.041}}$
	PRAUC	$0.2881_{.018}$	$0.2883_{.015}$	$0.3106_{.005}$	$0.3076_{.013}$	$0.3170_{.071}$	$0.2895_{.014}$	$\mathbf{0.3224_{.004}}$
HNRNPC	Top10 Prec.	$0.0000_{.000}$	$0.1000_{.081}$	$0.2333_{.124}$	$\underline{0.2666}_{.124}$	$0.2333_{.124}$	$0.0666_{.047}$	$\mathbf{0.3333_{.160}}$
	Top50 Prec.	$0.0266_{.018}$	$0.1533_{.052}$	$0.1666_{.065}$	$0.2266_{.047}$	$\underline{0.2266}_{.047}$	$0.1533_{.024}$	$\mathbf{0.2399_{.043}}$
	PRAUC	$0.0641_{.005}$	$0.1178_{.016}$	$0.1191_{.033}$	$\underline{0.1525}_{.010}$	$0.1434_{.035}$	$0.0913_{.003}$	$\mathbf{0.1628_{.080}}$
NELF	Top10 Prec.	$0.1666_{.124}$	$\underline{0.2000}_{.134}$	$0.1666_{.124}$	$0.2000_{.141}$	$0.0666_{.041}$	$0.1666_{.124}$	$\mathbf{0.2333_{.041}}$
	Top50 Prec.	$0.1866_{.049}$	$0.1599_{.041}$	$\underline{0.2000}_{.033}$	$0.1333_{.037}$	$0.1133_{.061}$	$0.1533_{.049}$	$\mathbf{0.2399_{.032}}$
	PRAUC	$0.0972_{.003}$	$0.0931_{.006}$	$\mathbf{0.1065_{.003}}$	$0.0969_{.013}$	$0.0850_{.005}$	$\underline{0.0982}_{.001}$	$0.0963_{.008}$
CHK2	Top10 Prec.	$0.1000_{.081}$	$0.1666_{.047}$	$\underline{0.2000}_{.000}$	$0.1333_{.124}$	$\mathbf{0.2666_{.124}}$	$0.1666_{.047}$	$0.1000_{.081}$
	Top50 Prec.	$0.1066_{.037}$	$0.0933_{.009}$	$\underline{0.1533}_{.047}$	$0.1199_{.032}$	$0.1466_{.033}$	$0.1333_{.049}$	$\mathbf{0.1733_{.037}}$
	PRAUC	$\underline{0.1273}_{.004}$	$0.1253_{.003}$	$0.1271_{.003}$	$0.1268_{.002}$	$0.1249_{.003}$	$0.1251_{.005}$	$\mathbf{0.1297_{.005}}$
UBLCP1	Top10 Prec.	$0.1000_{.000}$	$0.0599_{.043}$	$0.0666_{.047}$	$\underline{0.1266}_{.009}$	$0.0799_{.032}$	$0.1000_{.000}$	$\mathbf{0.1800_{.009}}$
	Top50 Prec.	$\underline{0.1400}_{.014}$	$0.1050_{.036}$	$0.1266_{.024}$	$0.1149_{.007}$	$0.1116_{.010}$	$0.1133_{.047}$	$\mathbf{0.1500_{.050}}$
	PRAUC	$0.1004_{.004}$	$0.0956_{.006}$	$\underline{0.1026}_{.002}$	$0.0977_{.002}$	$0.0968_{.002}$	$0.0977_{.003}$	$\mathbf{0.1070_{.001}}$

4.5 Comparison with RoseTTAFoldNA

In this section, we investigate the performance of RoseTTAFoldNA [5] which is a pretrained protein complex structure prediction model and compare it with FAFormer. The performance of FAFormer is evaluated on the individual predicted protein and nucleic acid structures by ESMFold and RoseTTAFoldNA. We additionally test the performance of AlphaFold3 [1] on a subset of the screening tasks due to AlphaFold3 server submission limits (Appendix E).

Dataset For the contact map prediction task, we select the test cases used in RoseTTAFoldNA to create the test set, yielding 86 protein-DNA and 16 protein-RNA cases. Furthermore, the complexes from our collected dataset that have more than 30% protein sequence identity to these test examples are removed. This leads to 1,962 training cases for protein-DNA and 1,094 for protein-RNA, which are used for training FAFormer. The MSAs of proteins and RNAs are retrieved for RoseTTAFoldNA.

For the aptamer screening task, we construct a smaller candidate set for each protein target by randomly sampling 10% candidates, given that the inference of RoseTTAFoldNA with MSA searching is time-consuming. The datasets will be equally split into validation and test sets. More details of these datasets can be found in Appendix D.

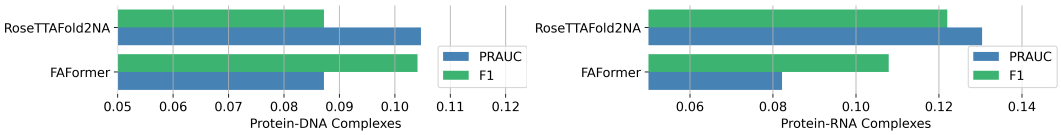


Figure 3: Contact map prediction on RoseTTAFoldNA test set.

Table 6: Comparison results with RoseTTAFoldNA using the sampled datasets, which accounts for the performance differences of FAFormer as shown in Table 5.

	Metric	GFP	HNRNPC	NELF	CHK2	UBLCP1
RoseTTAFoldNA	Top10 Prec.	0.4000	0.1000	0.0	0.0	0.0
	Top50 Prec.	0.3600	0.0599	0.0	0.1000	0.0199
	PRAUC	0.3926	0.1452	0.0481	0.1176	0.0722
FAFormer	Top10 Prec.	$0.4000_{.0}$	$0.1666_{.124}$	$0.1666_{.081}$	$0.1333_{.124}$	$0.1000_{.094}$
	Top50 Prec.	$0.3800_{.018}$	$0.0800_{.024}$	$0.0866_{.018}$	$0.1266_{.039}$	$0.0866_{.009}$
	PRAUC	$0.4027_{.022}$	$0.1781_{.089}$	$0.1044_{.018}$	$0.1374_{.013}$	$0.0762_{.016}$

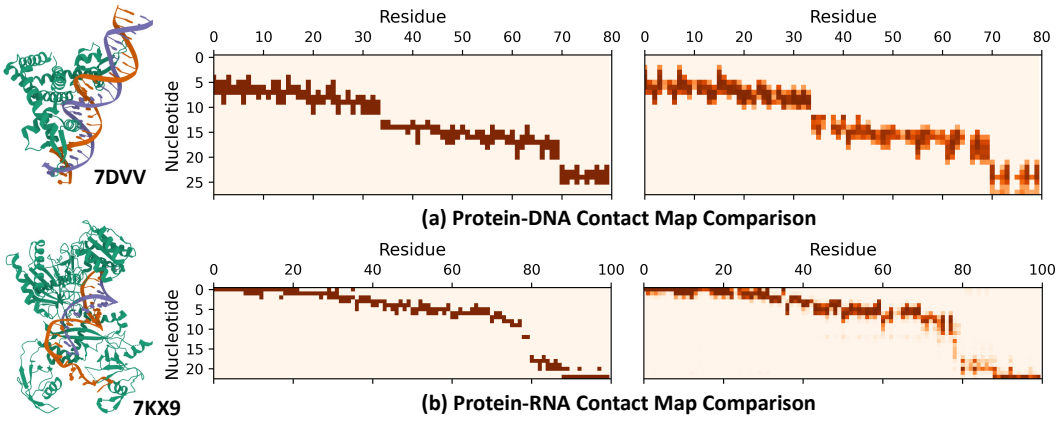


Figure 4: Case study based on two complex examples (PDB id: 7DVV and 7KX9), where each heatmap entry represents the contact probability between the nucleotide and residue. In each row, the figure on the left displays the ground truth contact maps, while the figure on the right displays the results predicted by FAFormer.

Results The comparison results of contact map prediction are presented in Figure 3, where FAFormer can achieve comparable performance to RoseTTAFoldNA using unbounded structures.

https://doi.org/10.52202/079017-4038

Specifically, our method has higher F1 scores for protein-DNA complexes (0.103 vs. 0.087) and performs comparably for protein-RNA complexes (0.108 vs. 0.12). Besides, Table 6 shows that RoseTTAFoldNA fails to receive positive aptamers for some targets, e.g., NELF and UBLCP1, while FAFormer consistently outperforms RoseTTAFoldNA for all the targets.

Aligning Figure 3 with Table 6, we observe that while FAFormer does not surpass RoseTTAFoldNA in contact map prediction, it significantly excels in aptamer screening. We attribute this to two reasons: 1) Similar to AlphaFold3 (Table 10), RoseTTAFoldNA as a foundational structure prediction model is optimized for general complex structures, which might bias its performance on some specific protein targets. For example, it can achieve good performance on proteins GFP and HNRNPC but fails for NELF. 2) The protein-RNA test set used by RoseTTAFoldNA for contact map prediction is limited, which may not comprehensively evaluate FAFormer.

Case Study Two examples of protein-DNA (PDB id: 7DVV) and protein-RNA (PDB id: 7KX9) complexes are provided in Figure 4, which shows a visual comparison between the actual (left) and the predicted (right) contact maps. Note that only the residues involved in the actual contact map are presented for a clear demonstration[4]. The complete contact map can be found in Appendix E.1. We can find that despite the sparsity of contact pairs, the predicted contact maps show a high degree of accuracy when compared to the ground truth.

Time Comparison Table 7 shows the average and total inference time of FAFormer and RoseTTAFoldNA on the test cases for the contact map prediction task, including the time for predicting the unbound structures for FAFormer. The predicted structures used for the evaluation of FAFormer are generated without protein and RNA MSAs, demonstrating a significantly faster inference speed by orders of magnitude. For the unsupervised aptamer screening task, while RoseTTAFoldNA requires only a single MSA search for each protein target, it needs to search MSAs for each RNA candidate sequence separately. Besides, the inclusion of MSAs results in a large input sequence matrix, leading to a time-consuming folding process of RoseTTAFoldNA.

	Protein-DNA		Protein-RNA	
	Avg.	Total	Avg.	Total
RoseTTAFoldNA	0.175h	15.12h	0.440h	7.04h
FAFormer	32.65s	0.78h	51.75s	0.23h

Table 7: Inference time on contact map prediction, denoted in seconds ("s") and hours ("h").

5 Conclusion

This research focuses on predicting contact maps for protein complexes in the physical space and reformulates the task of large-scale unsupervised aptamer screening as a contact map prediction task. To this end, we propose FAFormer, a frame averaging-based Transformer, with the main idea of incorporating frame averaging within each layer of Transformer. Our empirical results demonstrate the superior performance of FAFormer on contact map prediction and unsupervised aptamer screening tasks, which outperforms eight baseline methods on all the tasks.

Broader Impacts The proposed paradigm for aptamer screening can be extended to other modalities, such as protein-small molecules and antibody-antigen. Moreover, the strong correlation between contact prediction and affinity estimation demonstrated in our paper can guide future model development. Besides, FAFormer introduces a novel approach to equivariant model design by leveraging the flexibility of FA. This idea opens up numerous possibilities for future research, including exploring different ways to integrate FA with various neural network architectures.

Limitation In this study, the geometric features utilized in the encoders are limited to the coordinates of the C_α and C_3 atoms. Features extracted from the backbone or sidechains have not been used, which may limit the performance of the geometric encoders.

[4]We sort the residue IDs alongside the row ID so that the contact map appears diagnostic.

6 Acknowledgements

We thank Yangtian Zhang, Junchi Yu, Weikang Qiu, and anonymous reviewers for their valuable feedback on the manuscript. This work was supported by the BroadIgnite Award, the Eric and Wendy Schmidt Center at the Broad Institute of MIT and Harvard, NSF IIS Div Of Information & Intelligent Systems 2403317, and Amazon research.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[2] Bartosz Adamczyk, Maciej Antczak, and Marta Szachniuk. Rnasolo: a repository of cleaned pdb-derived rna 3d structures. *Bioinformatics*, 38(14):3668–3670, 2022.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Minkyung Baek, Ivan Anishchenko, Ian Humphreys, Qian Cong, David Baker, and Frank DiMaio. Efficient and accurate prediction of protein structure using rosettafold2. *bioRxiv*, pages 2023–05, 2023.

[5] Minkyung Baek, Ryan McHugh, Ivan Anishchenko, David Baker, and Frank DiMaio. Accurate prediction of nucleic acid and protein-nucleic acid complexes using rosettafoldna. *bioRxiv*, pages 2022–09, 2022.

[6] Ali Bashir, Qin Yang, Jinpeng Wang, Stephan Hoyer, Wenchuan Chou, Cory McLean, Geoff Davis, Qiang Gong, Zan Armstrong, Junghoon Jang, et al. Machine learning guided aptamer refinement and discovery. *Nature Communications*, 12(1):2366, 2021.

[7] Helen M Berman, Catherine L Lawson, and Bohdan Schneider. Developing community resources for nucleic acid structures. *Life*, 12(4):540, 2022.

[8] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[9] Nicholas Boyd, Brandon M Anderson, Brent Townshend, Ryan Chow, Connor J Stephens, Ramya Rangan, Matias Kaplan, Meredith Corley, Akshay Tambe, Yuzu Ido, et al. Atom-1: A foundation model for rna structure and function built on chemical mapping data. *bioRxiv*, pages 2023–12, 2023.

[10] Johann Brehmer, Pim De Haan, Sönke Behrends, and Taco S Cohen. Geometric algebra transformer. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Edward N Brody and Larry Gold. Aptamers as therapeutic and diagnostic agents. *Reviews in Molecular Biotechnology*, 74(1):5–13, 2000.

[12] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[13] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[14] Andrey A Buglak, Alexey V Samokhvalov, Anatoly V Zherdev, and Boris B Dzantiev. Methods and applications of in silico aptamer design and modeling. *International Journal of Molecular Sciences*, 21(22):8420, 2020.

[15] Jonghoe Byun. Recent progress and opportunities for nucleic acid aptamers. *Life*, 11(3):193, 2021.

[16] Chetan Chandola and Muniasamy Neerathilingam. Aptamers for targeted delivery: current challenges and future opportunities. *Role of novel drug delivery vehicles in nanobiomedicine*, pages 1–22, 2019.

[17] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *bioRxiv*, pages 2022–08, 2022.

[18] Zihao Chen, Long Hu, Bao-Ting Zhang, Aiping Lu, Yaofeng Wang, Yuanyuan Yu, and Ge Zhang. Artificial intelligence in aptamer–target binding prediction. *International journal of molecular sciences*, 22(7):3605, 2021.

 https://doi.org/10.52202/079017-4038

[19] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[20] Maria Francisca Coutinho, Liliana Matos, Juliana Inês Santos, and Sandra Alves. Rna therapeutics: how far have we gone? *The mRNA Metabolism in Human Disease*, pages 133–177, 2019.

[21] Alexandre Agm Duval, Victor Schmidt, Alex Hernández-Garcı'a, Santiago Miret, Fragkiskos D Malliaros, Yoshua Bengio, and David Rolnick. Faenet: Frame averaging equivariant gnn for materials modeling. In *International Conference on Machine Learning*, pages 9013–9033. PMLR, 2023.

[22] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[23] Andrew D Ellington and Jack W Szostak. In vitro selection of rna molecules that bind specific ligands. *nature*, 346(6287):818–822, 1990.

[24] Neda Emami and Reza Ferdousi. Aptanet as a deep learning approach for aptamer–protein interaction prediction. *Scientific reports*, 11(1):6074, 2021.

[25] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pages 2021–10, 2021.

[26] Enyue Fang, Xiaohui Liu, Miao Li, Zelun Zhang, Lifang Song, Baiyu Zhu, Xiaohong Wu, Jingjing Liu, Danhua Zhao, and Yuhua Li. Advances in covid-19 mrna vaccine development. *Signal transduction and targeted therapy*, 7(1):94, 2022.

[27] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

[28] Michal J Gajda, Irina Tuszynska, Marta Kaczor, Anastasia Yu Bakulina, and Janusz M Bujnicki. Filtrest3d: discrimination of structural models using restraints from experimental data. *Bioinformatics*, 26(23):2986–2987, 2010.

[29] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

[30] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

[31] Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

[32] Larry Gold, Nebojsa Janjic, Thale Jarvis, Dan Schneider, Jeffrey J Walker, Sheri K Wilcox, and Dom Zichi. Aptamers and the rna world, past and present. *Cold Spring Harbor perspectives in biology*, 4(3):a003582, 2012.

[33] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

[34] IR Humphreys, J Pei, M Baek, A Krishnakumar, I Anishchenko, S Ovchinnikov, J Zhang, TJ Ness, S Banjade, S Bagde, et al. Structures of core eukaryotic protein complexes. biorxiv 2021: 2021.09. 30.462231.

[35] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[36] Natsuki Iwano, Tatsuo Adachi, Kazuteru Aoki, Yoshikazu Nakamura, and Michiaki Hamada. Generative aptamer discovery using raptgen. *Nature Computational Science*, 2(6):378–386, 2022.

[37] Zheng Jiang, Yue-Yue Shen, and Rong Liu. Structure-based prediction of nucleic acid binding residues by merging deep learning-and template-based approaches. *PLOS Computational Biology*, 19(9):e1011428, 2023.

[38] Wengong Jin, Siranush Sarkizova, Xun Chen, Nir Hacohen, and Caroline Uhler. Unsupervised protein-ligand binding energy prediction via neural euler's rotation equation. *arXiv preprint arXiv:2301.10814*, 2023.

[39] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

[40] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[41] Anthony D Keefe, Supriya Pai, and Andrew Ellington. Aptamers as therapeutics. *Nature reviews Drug discovery*, 9(7):537–550, 2010.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[44] AV Lakhin, Vyacheslav Zalmanovich Tarantul, and LV3890987 Gening. Aptamers: problems, solutions and prospects. *Acta Naturae*, 5(4 (19)):34–43, 2013.

[45] Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of rna base pairs. *Rna*, 7(4):499–512, 2001.

[46] Shuya Li, Fanghong Dong, Yuexin Wu, Sai Zhang, Chen Zhang, Xiao Liu, Tao Jiang, and Jianyang Zeng. A deep boosting based approach for capturing the sequence binding preferences of rna-binding proteins from high-throughput clip-seq data. *Nucleic acids research*, 45(14):e129–e129, 2017.

[47] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

[48] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

[49] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[50] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. *arXiv preprint arXiv:2204.09410*, 2022.

[51] Qingxiu Liu, Wei Zhang, Siying Chen, Zhenjing Zhuang, Yi Zhang, Lingli Jiang, and Jun Sheng Lin. Selex tool: a novel and convenient gel-based diffusion method for monitoring of aptamer-target binding. *Journal of biological engineering*, 14:1–13, 2020.

[52] Shengchao Liu, Weitao Du, Yanjing Li, Zhuoxinran Li, Zhiling Zheng, Chenru Duan, Zhiming Ma, Omar Yaghi, Anima Anandkumar, Christian Borgs, et al. Symmetry-informed geometric representation for molecules, proteins, and crystalline materials. *arXiv preprint arXiv:2306.09375*, 2023.

[53] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

[54] Daniel Maticzka, Sita J Lange, Fabrizio Costa, and Rolf Backofen. Graphprot: modeling binding preferences of rna-binding proteins. *Genome biology*, 15(1):1–18, 2014.

[55] Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 2023.

[56] Alex Morehead, Chen Chen, Ada Sedova, and Jianlin Cheng. Dips-plus: The enhanced database of interacting protein structures for interface prediction. *Scientific Data*, 10(1):509, 2023.

[57] Flemming Morsch, Iswarya Lalitha Umasankar, Lys Sanz Moreta, Paridhi Latawa, Danny B Lange, Jesper Wengel, Huram Konjen, and Christian Code. Aptabert: Predicting aptamer binding interactions. *bioRxiv*, pages 2023–11, 2023.

[58] Wilma K Olson, Manju Bansal, Stephen K Burley, Richard E Dickerson, Mark Gerstein, Stephen C Harvey, Udo Heinemann, Xiang-Jun Lu, Stephen Neidle, Zippora Shakked, et al. A standard reference frame for the description of nucleic acid base-pair geometry. *Journal of molecular biology*, 313(1):229–237, 2001.

 https://doi.org/10.52202/079017-4038

[59] John M Pagano, Hojoong Kwak, Colin T Waters, Rebekka O Sprouse, Brian S White, Abdullah Ozer, Kylan Szeto, David Shalloway, Harold G Craighead, and John T Lis. Defining nelf-e rna binding in hiv-1 and promoter-proximal pause regions. *PLoS genetics*, 10(1):e1004090, 2014.

[60] Norbert Pardi, Michael J Hogan, Frederick W Porter, and Drew Weissman. mrna vaccines—a new era in vaccinology. *Nature reviews Drug discovery*, 17(4):261–279, 2018.

[61] Jung Woo Park, Philip NP Lagniton, Yu Liu, and Ren-He Xu. mrna vaccines for covid-19: what, why and how. *International journal of biological sciences*, 17(6):1446, 2021.

[62] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[63] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.

[64] Weikang Qiu, Huangrui Chu, Selena Wang, Haolan Zuo, Xiaoxiao Li, Yize Zhao, and Rex Ying. Learning high-order relationships of brain regions. In *Forty-first International Conference on Machine Learning*, 2023.

[65] Rahmatullah Roche, Bernard Moussad, Md Hossain Shuvo, Sumit Tarafder, and Debswapna Bhattacharya. Equipnas: improved protein-nucleic acid binding site prediction using protein-language-model-informed equivariant deep graph neural networks. *bioRxiv*, pages 2023–09, 2023.

[66] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[67] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

[68] Incheol Shin, Keumseok Kang, Juseong Kim, Sanghun Sel, Jeonghoon Choi, Jae-Wook Lee, Ho Young Kang, and Giltae Song. Aptatrans: a deep neural network for predicting aptamer-protein interaction using pretrained encoders. *BMC bioinformatics*, 24(1):447, 2023.

[69] Bo Shui, Abdullah Ozer, Warren Zipfel, Nevedita Sahu, Avtar Singh, John T Lis, Hua Shi, and Michael I Kotlikoff. Rna aptamers that functionally interact with green fluorescent protein and its derivatives. *Nucleic acids research*, 40(5):e39–e39, 2012.

[70] Zhenqiao Song, Tinglin Huang, Lei Li, and Wengong Jin. Surfpro: Functional protein design based on continuous surface. *arXiv preprint arXiv:2405.06693*, 2024.

[71] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International conference on machine learning*, pages 20503–20521. PMLR, 2022.

[72] Regina Stoltenburg, Christine Reinemann, and Beate Strehlitz. Selex—a (r) evolutionary method to generate high-affinity nucleic acid ligands. *Biomolecular engineering*, 24(4):381–403, 2007.

[73] Kylan Szeto, David R Latulippe, Abdullah Ozer, John M Pagano, Brian S White, David Shalloway, John T Lis, and Harold G Craighead. Rapid-selex for rna aptamers. *PloS one*, 8(12):e82667, 2013.

[74] Philipp Thölke and Gianni De Fabritiis. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

[75] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[76] Jacob M Tome, Abdullah Ozer, John M Pagano, Dan Gheba, Gary P Schroth, and John T Lis. Comprehensive analysis of rna-protein interactions by high-throughput sequencing–rna affinity profiling. *Nature methods*, 11(6):683–688, 2014.

[77] Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

[78] Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

[79] Ameni Trabelsi, Mohamed Chaabane, and Asa Ben-Hur. Comprehensive evaluation of deep learning architectures for prediction of dna/rna sequence binding specificities. *Bioinformatics*, 35(14):i269–i277, 2019.

[80] Irina Tuszynska, Marcin Magnus, Katarzyna Jonak, Wayne Dawson, and Janusz M Bujnicki. Npdock: a web server for protein–nucleic acid docking. *Nucleic acids research*, 43(W1):W425–W430, 2015.

[81] Irina Tuszynska, Dorota Matelska, Marcin Magnus, Grzegorz Chojnowski, Joanna M Kasprzak, Lukasz P Kozlowski, Stanislaw Dunin-Horkawicz, and Janusz M Bujnicki. Computational modeling of protein–rna complex structures. *Methods*, 65(3):310–319, 2014.

[82] Michael Uhl, Van Dinh Tran, Florian Heyl, and Rolf Backofen. Rnaprot: an efficient and feature-rich rna binding protein binding site predictor. *GigaScience*, 10(8):giab054, 2021.

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[84] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[85] Junkang Wei, Siyuan Chen, Licheng Zong, Xin Gao, and Yu Li. Protein–rna interaction prediction with deep learning: structure matters. *Briefings in bioinformatics*, 23(1):bbab540, 2022.

[86] Ying Xia, Chun-Qiu Xia, Xiaoyong Pan, and Hong-Bin Shen. Graphbind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues. *Nucleic acids research*, 49(9):e51–e51, 2021.

[87] Yiran Xu, Jianghui Zhu, Wenze Huang, Kui Xu, Rui Yang, Qiangfeng Cliff Zhang, and Lei Sun. Prismnet: predicting protein–rna interaction using in vivo rna structural information. *Nucleic Acids Research*, page gkad353, 2023.

[88] Zichao Yan, William L Hamilton, and Mathieu Blanchette. Neural representation and generation for rna secondary structures. *arXiv preprint arXiv:2102.00925*, 2021.

[89] Qianmu Yuan, Sheng Chen, Jiahua Rao, Shuangjia Zheng, Huiying Zhao, and Yuedong Yang. Alphafold2-aware protein–dna binding site prediction using graph transformer. *Briefings in Bioinformatics*, 23(2):bbab564, 2022.

[90] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

 https://doi.org/10.52202/079017-4038

A Experimental Details

Running environment. The experiments are conducted on a single Linux server with The AMD EPYC 7513-32 Core Processor, 1024G RAM, and 4 Tesla A40-48GB. Our method is implemented on PyTorch 1.13.1 and Python 3.9.6.

Training details. For all the baseline models and FAFormer, we fix the batch size as 8, the number of layers as 3, the dimension of node representation as 64, and the optimizer as Adam [42]. Binary cross-entropy loss is used for contact map identification tasks with a positive weight of 4. The gradient norm is clipped to 1.0 in each training step to ensure learning stability. We report the model's performance on the test set using the best-performing model selected based on its performance on the validation set. All the results are reported based on three different random seeds.

The learning rate is tuned within {1e-3, 5e-4, 1e-4} and is set to 1e-3 by default, as it generally yields the best performance. For each model, we search the hyperparameters in the following ranges: dropout rate in [0, 0.5], the number of nearest neighbors for the GNN-based methods in {10, 20, 30}, and the number of attention heads in {1, 2, 4, 8}. The hyperparameters used in each method are shown below:

- FAFormer: The number of attention heads, dropout rate, and attention dropout rate are 4, 0.2, and 0.2 respectively. We initialize the weight of the gate function with zero weights, and bias with a constant value of 1, ensuring a mostly-opened gate. SiLU [22] is used as the activation function. The distance threshold c is set as 1e5Å and the number of neighbors is 30.
- GVP-GNN[5]: The dimensions of node/edge scalar features and node/edge vector features are set as 64 and 16 respectively. The dropout rate is fixed at 0.2. For a fair comparison, we only extract the geometric feature based on C_α, i.e., the forward and reverse unit vectors oriented in the direction of C_α between neighbor residues.
- EGNN[6]: The number of neighbors is set as 30. Besides, we apply gate attention to each edge update module and residue connection to the node update module. SiLU [22] is used as the activation function.
- Equiformer[7] and SE(3)Transformer[8]: The number of attention heads, and the hidden size of each attention head are set as 4 and 16. We exclude the neighbor nodes with a distance greater than 100Å and set the number of neighbors as 30. Based on our experiments, we set the degree of spherical harmonics to 1, as higher degrees tend to lead to performance collapse according to our experiments.
- Transformer and FA[9]: The Transformer is applied as FA's backbone encoder. The dropout and attention dropout rates are 0.2 and 0.2. The number of attention heads is set as 4.
- GraphBind[10]: The dropout ratio and the number of neighbors are 0.5 and 30. We apply addition aggregation to the node and edge update module, following the suggested setting presented in the paper.
- GraphSite[11]: The number of neighbors and dropout ratio are 30 and 0.2. The number of attention layers and attention heads are 2 and 4 respectively. Besides, we additionally use the DSSP features as the node features, as suggested in the paper.
- RoseTTAFoldNA[12]: We employ the released pretrained weight of RoseTTAFoldNA, and set up all the required databases, including UniRef, BFD, structure templates, Rfam, and RNAcentral following the instructions.

[5] https://github.com/drorlab/gvp-pytorch
[6] https://github.com/vgsatorras/egnn
[7] https://github.com/atomicarchitects/Equ.iformer
[8] https://github.com/FabianFuchsML/se3-transformer-public
[9] https://github.com/omri1348/Frame-Averaging
[10] http://www.csbio.sjtu.edu.cn/bioinf/GraphBind/sourcecode.html
[11] https://github.com/biomed-AI/GraphSite
[12] https://github.com/uw-ipd/RoseTTAFold2NA

B Efficiency Analysis

B.1 Computational Complexity

As a core component in the model, frame averaging's time complexity mainly comes from the calculation of PCA among the input coordinates. This operation is practically efficient due to the low dimensionality of the input (only 3 for coordinates). Besides, the calculation of eigenvalue decomposition can be significantly accelerated by some libraries, such as PyTorch [62] and SciPy [84]. We ignore the complexity used for calculating projected coordinates in the following analysis.

As for the local frame edge module in FAFormer, linear transformations are employed to compute the pairwise representation (Equ. (7) and Equ. (8)), and an additional gate is applied to regulate these messages (Equ. (8)), resulting in a computational complexity of $O(NKD + NKD^2)$. Considering the residue connection, the overall complexity of the local frame edge module is $O(NKD+NKD^2+ND)$.

As for the self-attention module, linear transformations are performed on token embeddings and edge representations (Equ. (9)), and the multi-head MLP attention computation is limited to nearest neighbors (Equ. (10)), leading to the complexity of $O(NHD^2+NKHD)$ where H is the number of attention heads. Further operations, including aggregation, linear projection, and residual connections (Equ. (11) and Equ. (12)), add $O(NKHD + NKHD^2 + ND)$. Moreover, applying a gate function to the combination of aggregated coordinates and original coordinates (Equ. (13) and Equ. (14)) adds $O(NKH + ND + ND^2)$. As a result, the total complexity for the self-attention module is $O(NKHD + NKHD^2 + NKH + ND + ND^2)$.

Regarding the FFN, most operations are linear transformations that have the complexity of $O(ND^2)$. The gate function and linear combination of coordinates contribute $O(ND^2 + ND)$. So the total computational complexity of FFN is $O(ND^2 + ND)$.

B.2 Wall-Clock Time Performance Comparison

We conduct a comparison of wall-clock time performance between FAFormer and other geometric baseline models under the same computational environment. Specifically, we measure the average training time for one epoch of each model, with the results illustrated in Figure 5. It can be observed that FAFormer demonstrates greater efficiency compared to spherical harmonics-based models and achieves performance comparable to the GNN-based method GVP-GNN. Such efficiency is attributed to the utilization of top-K neighbor graphs and FA-based modules in FAFormer, which enable efficient modeling of coordinates through linear transformations.

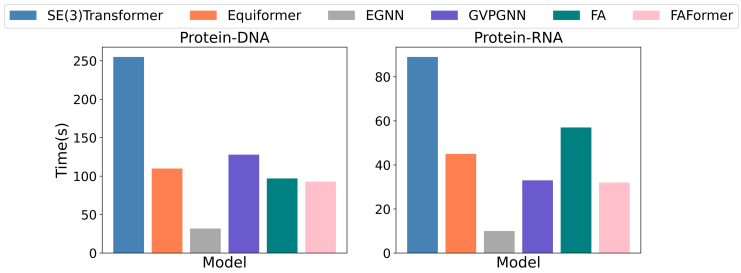


Figure 5: Training time comparison between FAFormer and the other baseline models.

C Equivariance Proof

In this section, we investigate the equivariance of the gate function (Figure 2(e)), which can be formulated as:

$$\text{Gate}(\boldsymbol{X}_i, f_\Delta(\boldsymbol{X}_i), \boldsymbol{Z}_i) := \boldsymbol{g}_i \Delta \boldsymbol{X}_i + (1 - \boldsymbol{g}_i)\boldsymbol{X}_i \tag{18}$$

$$:= \boldsymbol{X}'_i \tag{19}$$

where $\boldsymbol{X}_i$ represents the coordinates of i-th node, $\boldsymbol{g}_i = \sigma(\text{Linear}(\boldsymbol{Z}_i))$ is the gate score based on i-th node's feature, and $\Delta \boldsymbol{X}_i = f_\Delta(\boldsymbol{X}_i)$ denotes the coordinate update function, i.e., the multi-head

https://doi.org/10.52202/079017-4038

aggregation function Equ.(13) which is based on the frame averaging and thus is $E(3)$ equivariant:

$$\boldsymbol{Q}\Delta\boldsymbol{X}_i + \boldsymbol{t} = f_\Delta(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}) \tag{20}$$

where $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector and $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$ is an orthogonal matrix.

We aim to prove that the gate function is $E(3)$ equivariant, meaning it is translation equivariant for any translation vector $\boldsymbol{t} \in \mathbb{R}^3$ and rotation/reflection equivariant for any orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$. Specifically, we want to show:

$$\boldsymbol{Q}\boldsymbol{X}'_i + \boldsymbol{t} = \text{Gate}(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}, f_\Delta(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}), \boldsymbol{Z}_i) \tag{21}$$

Derivation.

$$\begin{aligned}
\text{Gate}(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}, f_\Delta(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}), \boldsymbol{Z}_i) &= \boldsymbol{g}_i(\boldsymbol{Q}\Delta\boldsymbol{X}_i + \boldsymbol{t}) + (1 - \boldsymbol{g}_i)(\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{t}) && (22)\\
&= \boldsymbol{g}_i\boldsymbol{Q}\Delta\boldsymbol{X}_i + (1 - \boldsymbol{g}_i)\boldsymbol{Q}\boldsymbol{X}_i + \boldsymbol{g}_i\boldsymbol{t} + (1 - \boldsymbol{g}_i)\boldsymbol{t} && (23)\\
&= \boldsymbol{Q}\left(\boldsymbol{g}_i\Delta\boldsymbol{X}_i + (1 - \boldsymbol{g}_i)\boldsymbol{X}_i\right) + \boldsymbol{t} && (24)\\
&= \boldsymbol{Q}\boldsymbol{X}'_i + \boldsymbol{t} && (25)
\end{aligned}$$

Therefore, we have proven that applying rotation and translation to $\boldsymbol{X}_i$ results in the identical rotation and translation being applied to $\boldsymbol{X}'_i$.

D Dataset Descriptions

Protein Complex Datasets We collect the complexes from PDB [8], NDB [7], RNASolo [2] and DIPS [77] databases. Complexes are excluded if they have protein sequences shorter than 5 or longer than 800 residues, or nucleic acid sequences shorter than 5 or longer than 500 nucleotides. The redundant proteins with over 90% sequence similarity to other sequences within the datasets are removed. The protein and the binder structures will be separated and decentered.

Aptamer Datasets Detailed information on each protein target and the aptamer candidates is presented below. The threshold for categorizing sequences as positive or negative aptamers is determined by referencing previous studies or identifying natural cutoffs in the affinity score distributions.

- GFP[13]: Green fluorescent protein. The aptamer candidates are mutants of GFPapt [69], with K_d values ranging from 0nM to 125nM as affinity measures. Candidates with K_d values lower than 10nM are considered positive cases.
- NELF[14]: Negative elongation factor E. The aptamer candidates are mutants of NELFapt [59], with K_d values ranging from 0nM to 183nM as affinity measures. Candidates with K_d values lower than 5nM are considered positive cases.
- HNRNPC[15]: Heterogeneous nuclear ribonucleoproteins C1/C2. The aptamer candidates are the randomly generated RNA 7mers and we apply the affinity values provided by the previous studies [46]. Candidates with affinity scores lower than -0.5 are positive.
- CHK2[16]: Serine/threonine-protein kinase Chk2. Szeto et al [73] applied SELEX (Systematic Evolution of Ligands by EXponential Enrichment) [23] to screen aptamers from a large library of random nucleic acid sequences through multiple rounds. During each round, the bound sequences were amplified and isolated. We use the final round of sequences as the candidates, with sequences having multiplicities over 100 considered positive aptamers.
- UBLCP1[17]: Ubiquitin-like domain-containing CTD phosphatase 1. Similar to CHK2, the final round of sequences are considered candidates, with sequences having multiplicities over 200 considered positive aptamers.

[13] https://www.uniprot.org/uniprotkb/P42212/entry
[14] https://www.uniprot.org/uniprotkb/P92204/entry
[15] https://www.uniprot.org/uniprotkb/P07910/entry
[16] https://www.uniprot.org/uniprotkb/O96017/entry
[17] https://www.uniprot.org/uniprotkb/Q8WVY7/entry

Table 8: Sampled aptamer dataset statistics.

Target	GFP	NELF	HNRNPC	CHK2	UBLCP1
#Positive	55	62	20	122	66
#Candidate	188	981	328	1,000	1,000

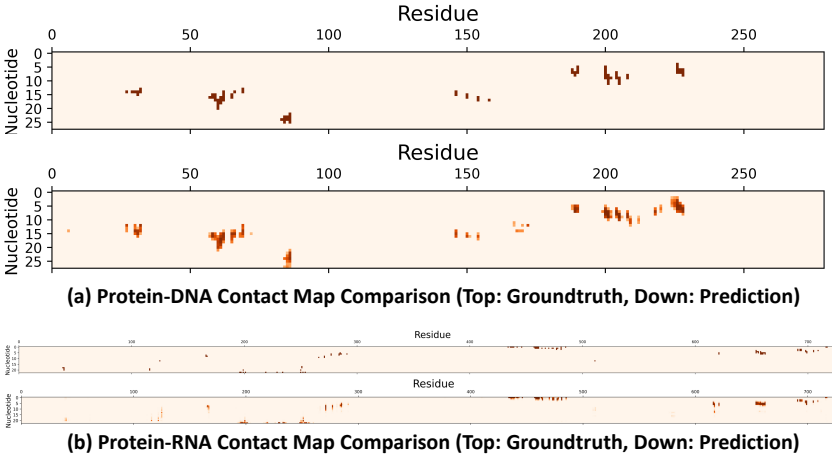
Sampled Aptamer Datasets We construct smaller aptamer datasets for the comparison between RoseTTAFoldNA and FAFormer by randomly sampling 10% candidates from the original datasets. The statistics are shown in Table 8. We equally split each dataset into a validation set and a test set. The performance of FAFormer on the test set is reported with the best performance on the validation set. RoseTTAFoldNA is directly evaluated on the test set.

Test datasets of RoseTTAFoldNA The test datasets used to evaluate RoseTTAFoldNA are available at this accessible link. We downloaded the dataset and filtered out non-dimer complexes, resulting in 86 protein-DNA and 16 protein-RNA complexes.

E Additional Experiments

E.1 Case Study

Figure 6 presents the complete groundtruth and predicted contact maps of the cases used in Figure 4, where the model accurately captures the sparse pattern.



(a) Protein-DNA Contact Map Comparison (Top: Groundtruth, Down: Prediction)

(b) Protein-RNA Contact Map Comparison (Top: Groundtruth, Down: Prediction)

Figure 6: Case study based on two complex examples (PDB id: 7DVV and 7KX9).

E.2 Ablation Study

In this section, we conduct an ablation study to investigate the impact of FAFormer's core modules. Specifically, we individually disable the edge module and attention mechanism, and replace the proposed FFN with the conventional FFN in FAFormer. The results are presented in Table 9.

Table 9: Ablation study.

	Protein-DNA		Protein-RNA		Protein-Protein	
	F1	PRAUC	F1	PRAUC	F1	PRAUC
FAFormer	$0.1457_{.005}$	$0.1279_{.006}$	$0.1284_{.003}$	$0.1113_{.004}$	$0.1596_{.002}$	$0.1463_{.003}$
w/o Edge	$0.1171_{.004}$	$0.1225_{.002}$	$0.1048_{.007}$	$0.0983_{.008}$	$0.1100_{.005}$	$0.0972_{.002}$
w/o Attention	$0.1401_{.001}$	$0.1250_{.006}$	$0.1059_{.002}$	$0.0973_{.001}$	$0.1325_{.001}$	$0.1121_{.001}$
w/o FAFFN	$0.1332_{.001}$	$0.1211_{.002}$	$0.1078_{.005}$	$0.0958_{.001}$	$0.1474_{.001}$	$0.1334_{.001}$

https://doi.org/10.52202/079017-4038

Removing any core module results in a significant performance decline. Notably, the model degrades to either an attention-only or message-passing-only architecture when the edge or attention module is removed, both of which lead to significant declines in performance. This demonstrates the advantage of combining these two architectures with FA.

E.3 Aptamer Screening with AlphaFold3

AlphaFold3 [1] represents the latest advancement in biomolecular complex structure prediction and is accessible through an online server[18]. Due to its limited quota (20 jobs per day), we evaluated it on two of the smallest sampled aptamer datasets, GFP and HNRNPC. The statistics for these datasets are presented in Table 8. We used the contact probability produced by the server and the maximum probability as the estimated affinity. The results are shown in Table 10. Although AlphaFold3 performs well in predicting the complex structure, it fails to identify promising aptamers in our cases. We attribute this to fine-grained optimization and overfitting on molecule interaction patterns in the structure prediction task, which have biased its screening performance on specific protein targets.

Table 10: Comparison results with AlphaFold3.

	Metric	AlphaFold3	RoseTTAFoldNA	FAFormer
GFP	Top10 Prec.	0.3000	0.4000	$0.4000_{.0}$
	Top50 Prec.	0.3199	0.3600	$0.3800_{.018}$
	PRAUC	0.3132	0.3926	$0.4027_{.022}$
HNRNPC	Top10 Prec.	0.1000	0.1000	$0.1666_{.124}$
	Top50 Prec.	0.0799	0.0599	$0.0800_{.024}$
	PRAUC	0.1355	0.1452	$0.1781_{.089}$

[18] https://golgi.sandbox.google.com/

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As presented in the abstract and introduction, we focus on learning the interactions of protein complexes and extending it to the zero-shot aptamer screening application. Besides, we propose FAFormer, which is an equivariant Transformer architecture based on frame averaging, achieves the best performance on all the tasks.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We've included a section in the paper specifically focusing on modeling the interactions of protein complexes and zero-shot aptamer screening without evaluating FAFormer on other tasks. We will continue to work on this and aim to benchmark the method more comprehensively.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

https://doi.org/10.52202/079017-4038

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide proof verifying the equivariance of the gate function, including a detailed derivation demonstrating how the gate function maintains equivariance with orthogonal matrices and translation vectors.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We've included a code link in the paper for review, which contains the pipelines for all tasks and datasets. Additionally, we present the data sources and processing methods for all datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A code link is included in the paper, containing the training pipelines for all tasks. Additionally, we have detailed the dataset preprocessing methods and provided the corresponding scripts in the code link.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: As shown in Appendix A and Appendix D, we have introduced the dataset splitting strategies based on protein sequence identity, the search range of each hyperparameter, and the optimal hyperparameters for each model.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

https://doi.org/10.52202/079017-4038

Justification: We report the variance for all results by repeating the experiments with three different random seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As mentioned in the Appendix A, the experiments are conducted on a single Linux server with The AMD EPYC 7513-32 Core Processor, 1024G RAM, and 4 Tesla A40-48GB. Our method is implemented on PyTorch 1.13.1 and Python 3.9.6. The computational complexity and running time comparison are shown in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We've followed the instructions in the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper presents work whose goal is to advance the field of geometric deep learning on protein complex modeling. There are some potential societal consequences of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The datasets and models used in this study don't have such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

https://doi.org/10.52202/079017-4038

Justification: All the involved models and datasets, including pretrained protein/RNA sequence models, pretrained structure prediction models (RoseTTAFoldNA and AlphaFold3), complex datasets, and aptamer datasets, are properly credited and cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets; it only uses existing models and datasets that are properly credited and cited.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-4038