Floating Anchor Diffusion Model for Multi-motif Scaffolding

Ke Liu [1 *] **Weian Mao** [2 1 *] **Shuaike Shen** [1 *] **Xiaoran Jiao** [1] **Zheng Sun** [3] **Hao Chen** [1] **Chunhua Shen** [1 4]

Abstract

Motif scaffolding seeks to design scaffold structures for constructing proteins with functions derived from the desired motif, which is crucial for the design of vaccines and enzymes. Previous works approach the problem by inpainting or conditional generation. Both of them can only scaffold motifs with fixed positions, and the conditional generation cannot guarantee the presence of motifs. However, prior knowledge of the relative motif positions in a protein is not readily available, and constructing a protein with multiple functions in one protein is more general and significant because of the synergies between functions. We propose a Floating Anchor Diffusion (FADiff) model. FADiff allows motifs to float rigidly and independently in the process of diffusion, which guarantees the presence of motifs and automates the motif position design. Our experiments demonstrate the efficacy of FADiff with high success rates and designable novel scaffolds. To the best of our knowledge, FADiff is the first work to tackle the challenge of scaffolding multiple motifs without relying on the expertise of relative motif positions in the protein. Code is available at https://github.com/aim-uofa/FADiff.

1. Introduction

The design of proteins with specific functions is significant for vaccines and enzymes (Correia et al., 2014; Linsky et al., 2020; Sesterhenn et al., 2020). One crucial way is to design stable *scaffolds* to support desired *motifs* (Watson et al., 2023; Trippe et al., 2023; Ingraham et al., 2023). Here motifs refer to protein structure fragments, which impart biological functions to proteins (Hutchinson & Thornton, 1996). Motif scaffolding has already proven to be significant in the wet experiment since drugs have been designed by solving specific instances of the motif-scaffolding problem (Procko et al., 2014; Siegel et al., 2010). The development of generative models, especially diffusion models, speeds up solving the motif scaffolding problem (Song et al., 2021; Yim et al., 2023; Huang et al., 2022). Scaffolding multiple motifs in one protein is more general and significant because of the synergies between functions. However, previous works focus on scaffolding one motif at a fixed location. Adapting these approaches to scaffolding multiple motifs requires prior knowledge of relative positions between multiple motifs which is not readily available.

Previous works approach the motif-scaffold problem via conditional generation or inpainting. Conditional generation methods like SMCdiff (Trippe et al., 2023) are only able to scaffold one motif while the presence of motifs is not guaranteed. For inpainting methods, like Chroma (Ingraham et al., 2023) and RFdiffusion (Watson et al., 2023), they fix both the structure position and sequence position of desired motifs. Consequently, to scaffold multiple motifs, the relationship between their positions must be supplied to the model in advance, necessitating domain knowledge that isn't always readily available. Even for a single motif, the sequence position is fixed.

To tackle the challenge of supporting multiple motifs, we propose a novel model dubbed **F**loating **A**nchor **Diff**usion model (**FADiff**), which not only ensures the existence of motifs but also automates motif position design without the need for prior domain expertise. The underlying principle of FADiff rests on treating the anchor motifs as rigid entities, thus permitting motifs to maintain their structures while floating. Given that a motif is composed of amino acids, it is likewise guided within the network alongside other amino acids. With the intent to preserve the structure of motifs, we treat them as rigid anchors during the diffusion process. The movement of motifs is dictated by their constituent amino acids, which further shapes the formation of the diffusion process with rigid movable substructures in this work.

Utilizing FADiff, we assure the presence of desired motifs, considering them not just as generation conditions, but as fundamental components of the generation results, analogous to the inpainting. Contrarily, while inpainting methods fixate the positions of the motifs, FADiff brings innovation to the table by independently determining the positions

[*]Equal contribution [1]Zhejiang University, China [2]The University of Adelaide, Australia [3]Swansea University, UK [4]Ant Group. Correspondence to: Hao Chen <haochen.cad@zju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

of each motif. Specifically, anchor motifs within FADiff maintain independent and rigid mobility. Guided by their internal amino acids, this property further enables the flexible movement of these anchor motifs towards rational positions. FAdiff encourages flexible motif scaffolding, negating the need for not readily available domain expertise to assign the structural or sequential arrangement within the generated protein structure.

To demonstrate the efficacy and generalization of our FADiff, we carried out a comprehensive series of experiments. The empirical findings indicate that given multiple motifs, FADiff can effectively position them while concurrently generating designable scaffolds to support them. It is worth noting that, once trained on the task of scaffolding two motifs, FADiff can be extrapolated to scaffold any other number of motifs. These observations indicate that FADiff potentially offers a general solution to the multi-motif scaffolding problem.

To the best of our knowledge, FADiff is the first work to tackle the problem of scaffolding multiple motifs without the need for prior knowledge of the relative positions of multiple motifs, which is often unobtainable. The main contributions of our work can be summarized as follows:

- We propose a practical and significant problem of scaffolding multiple motifs where the prior knowledge of their relative positions is not readily available.
- We propose a new diffusion model, floating anchor diffusion (FADiff) to tackle the problem of scaffolding multiple motifs. FADiff assures the existence of motifs and automates the design of motif position by facilitating the rigid movement of the motifs.
- Our experiments demonstrate that FADiff can float the anchor motifs to rational positions and generate designable scaffolds to support them. The generalization of FADiff indicates its potential to be a general solution to multiple motif scaffolding.

2. Related Works

2.1. Motif scaffolding problem

Multi-motif scaffolding is a central task in protein design. For example, a protein boasting high specificity can be fashioned by assimilating several recognized binding motifs (Pawson & Scott, 1997; Cao et al., 2022; Jiang et al., 2023). Furthermore, via expert knowledge, a pair of EF-hand motifs are effectively merged into the protein structure (Wang et al., 2022). Importantly, in many instances, either sequence or structure relative positions between motifs remain undetermined. While this situation permits the resolution of some issues, it persistently demands considerable experimentation, human intervention, and specialized knowledge (Roel-Touris et al., 2023; Davila-Hernandez et al., 2023; Roy et al., 2023). Additionally, these strategies display pronounced shortcomings, particularly when confronting conditions devoid of suitable templates and references in the Protein Data Bank (PDB) (Berman et al., 2000). FADiff provides a general solution to multi-motif scaffolding without any reliance on domain expertise.

2.2. Generative models for scaffolding motifs

The advent of generative protein models (De Bortoli et al., 2022; Lee et al., 2023; Madani et al., 2023; Trippe et al., 2023; Gruver et al., 2023; Lisanza et al., 2023) has instigated a dramatic evolution in protein design. Motif-scaffolding, a pivotal undertaking within protein design, has been consistently broached by diverse diffusion model techniques throughout the years. Generative models try to solve the motif scaffolding problem by conditional generation or inpainting. For example, SMCDiff (Trippe et al., 2023) and Chroma(Ingraham et al., 2023) take motifs as guidance for their pre-trained unconditional model to generate proteins with motifs in it. RFdiffusion (Watson et al., 2023) fixes the motifs in a protein and paints the scaffold. However, both two methods fail to scaffold multiple motifs since the motif positions in the protein are manually determined and fixed for them. The conditional generative methods even cannot guarantee the presence of motifs in the generated protein. FADiff solves the problem by enabling the motifs to float rigidly in the diffusion process, which leads to the automatic position design and the existence of motifs in the generated protein.

3. Preliminaries and Notation

3.1. The multi-motif scaffolding problem

A protein $\mathcal{P} = \{\mathcal{A}, \mathbf{X}\}$ is defined by its amino acid sequence $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ and backbone structure $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^T \in \mathbb{R}^{n,3}$, where n denotes the number of amino acids in a protein. $a_i \in \mathcal{C}^{20}$ denotes the type of i-th amino acids, where $\mathcal{C}$ is a set of 20 genetically-encoded amino acids. $\mathbf{x}_i \in \mathbb{R}^3$ is the i-th C-α residue backbone coordinates in 3D. The 3D structure of a protein can be determined by its corresponding amino acid sequence, *i.e.*, $\mathbf{X}(\mathcal{A})$. In addition, the order of the amino acids in the sequence, *i.e.*, the sequence position, is also an important piece of implicit information in $\mathcal{A}$, denoted by $\mathcal{D}$. Thus, the amino acid sequence consists of the sequence position and amino acid types, *i.e.*, $\mathcal{A} = \{\mathcal{C}^n, \mathcal{D}\}$. We can define a protein as:

Definition 3.1 (Protein structure)**.** A protein structure consists of amino acid sequence $\mathcal{A}$ and backbone structure $\mathbf{X}$, where $\mathcal{A}$ contains both the amino acid types $\mathcal{C}$ and index in sequence $\mathcal{D}$, *i.e.*, $\mathcal{P} = \{\mathcal{A}, \mathbf{X}\} = \{\mathcal{C}^n, \mathcal{D}, \mathbf{X}\}$

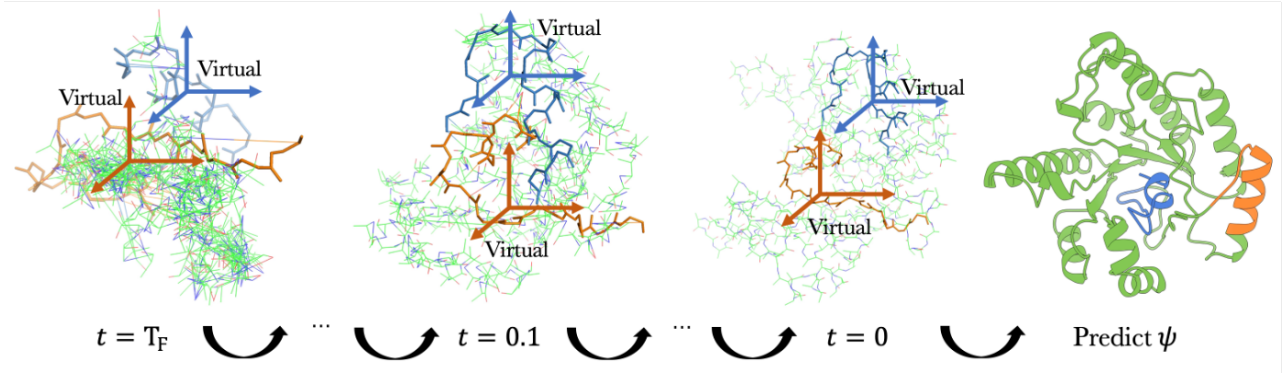

Figure 1. In the denoising process, we keep the motifs translating and rotating rigidly, which means the internal structure of motifs is maintained while their positions in the protein are flexible. The orange and blue colors indicate the anchor motifs that float rigidly. The green color indicates the scaffold residues. The coordinate system in color denotes the virtual coordinate system, which is the geometry center of each motif.

Given a protein $\mathcal{P}$, we can divide it into the functional motif $\mathcal{M}$ and the scaffold $\mathcal{S}$. Since multiple motifs exist in one protein, $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m\}$, the protein is denoted as $\mathcal{P} = \mathcal{M}_\mathcal{P} \cup \mathcal{S}_\mathcal{P} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m, \mathcal{S}\}$, where m is the number of motifs. Therefore, we can define the scaffolded motif and scaffolding as:

Definition 3.2 (Scaffolded motif and scaffold)**.** We can describe a scaffolded motif $\mathcal{M}_\mathcal{P}$ in a protein with its positions in the protein and its internal structure, specifically, the internal structure $\mathcal{M}_\mathrm{X}$, the internal sequence $\mathcal{M}_\mathcal{A}$, the position in the protein structure $\mathbf{X}_\mathcal{M}$, and the position in the protein sequence $\mathcal{A}_\mathcal{M}$ of motif, *i.e.*, $\mathcal{M}_\mathcal{P} = \{\mathcal{M}_\mathrm{X}, \mathcal{M}_\mathcal{A}, \mathbf{X}_\mathcal{M}, \mathcal{A}_\mathcal{M}\}$. Similarly, the scaffolding can be defined as $\mathcal{S}_\mathcal{P} = \{\mathcal{S}_\mathbf{X}, \mathcal{S}_\mathcal{A}, \mathbf{X}_\mathcal{S}, \mathcal{A}_\mathcal{S}\}$

The common motif-scaffolding settings focus on the backbone generation and the order of residues $\mathcal{D}$ in protein amino acids sequence $\mathcal{A}$ is considered. Therefore we ignore the $\mathcal{C}$ in $\mathcal{A}$ in the below as per the common setup. Inpainting methods (Ingraham et al., 2023; Watson et al., 2023) for motif scaffolding require the motif structure, its position in the protein structure, and its sequence position in the protein sequence prior, which can be remarked as:

Remark 3.3 (Motif-scaffolding by inpainting). Inpainting methods seek to predict the scaffolding structures $\mathcal{S}_\mathcal{P}$ given the motif structure $\mathcal{M}_\mathcal{P}$, *i.e.*, $\mathcal{S}_\mathcal{P} = f(\mathcal{M}_\mathrm{X}, \mathcal{M}_\mathcal{A}, \mathbf{X}_\mathcal{M}, \mathcal{A}_\mathcal{M})$.

The motif position in the protein $\mathbf{X}_\mathcal{M}, \mathcal{A}_\mathcal{M}$ is specified manually in inpainting methods, which is not readily available.

The conditional generation methods (Trippe et al., 2023) require only the motif internal structures to predict the structure of the whole protein, which can be remarked as:

Remark 3.4 (Motif-scaffolding by conditional generation). Conditional generation methods sought to predict the protein structures $\mathcal{P}$ given the motif's internal structure, *i.e.*, $\mathcal{P} = f(\mathcal{M}_\mathrm{X}, \mathcal{M}_\mathcal{A})$.

In the conditional generation, the presence of motifs is not guaranteed. For multiple motifs which are encoded together, the relative position between them is fixed.

To maintain the motif in the generated protein and enable the automatic position design, we formulate the multiple motif scaffolding problem as:

Definition 3.5 (Multiple motif scaffolding problem)**.** Given the internal structure of multiple motifs, the multiple motif scaffolding seeks to predict the scaffolding and the motif positions in the protein, *i.e.*, $\{\mathcal{S}_\mathcal{P}, \mathbf{X}_\mathcal{M}, \mathcal{A}_\mathcal{M}\} = f(\mathcal{M}_\mathrm{X}, \mathcal{M}_\mathcal{A})$

3.2. Backbone parameterization

We adopt the protein backbone parameterization and notations in FrameDiff (Yim et al., 2023). Each residue backbone is parameterized by an orientation preserving rigid transformation (*frame*) $\mathbf{T} \in \mathbb{R}^{4\times4}$ that maps from fixed coordinates $\mathrm{N}^*, \mathrm{C}_\alpha^*, \mathrm{C}^*, \mathrm{O}^* \in \mathbb{R}^3$ centers at $\mathrm{C}_\alpha^* = (0,0,0)$. Thus, the main atom coordinates of the i-th residue on the backbone are obtained as

$$[\mathrm{N}_i, \mathrm{C}_i, (\mathrm{C}_\alpha)_i] = \mathbf{T}_i \cdot [\mathrm{N}^*, \mathrm{C}_\alpha^*, \mathrm{C}^*],$$

where $\mathbf{T}_i$ is an operation of the special Euclidean (SE(3)) group. Each transformation $\mathbf{T}_i$ can be decomposed into rotation $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ and translation $\mathbf{X}_i \in \mathbb{R}^3$, *i.e.*, $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{X}_i)$, where $\mathbf{R}_i \in \mathrm{SO}(3)$. Therefore, given a coordinate $\mathbf{v} \in \mathbb{R}^3$ in the i-th frame, its location in the fixed coordinate is given as

$$\mathbf{T}_i\mathbf{v} = \mathbf{R}_i\mathbf{v} + \mathbf{X}_i. \tag{1}$$

Further, with an additional torsion angle ϕ, the coordinates of atom O in the residue can be determined. Different from FrameDiff, FADiff takes each motif as rigid and enables their movement in the diffusion process, as shown in Fig. 1, *i.e.*, the structure of each motif is preserved and can float

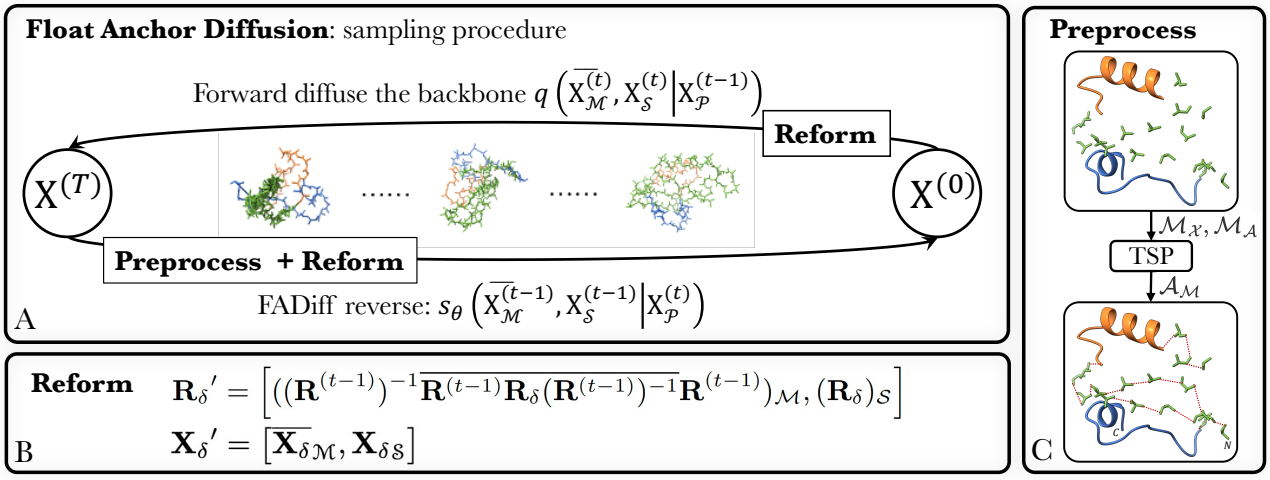

Figure 2. **A)** Given multiple motifs with their internal structure $\mathcal{M}_{\mathrm{X}}$ and $\mathcal{M}_{\mathcal{A}}$. We specify the sequence position of residues by finding the shortest chain with a greedy algorithm like the traveling salesman problem (TSP), where the distances are the gaps between the atoms C and N of two residues. In both the forward process and reverse process, we take the motifs as rigid and enable them to float rigidly. **B)** Generally, we reform the noise and updates for each motif. **C)** The preprocess of TSP. The orange and blue colors indicate the motifs. The residues in green are generated scaffolds.

rigidly. The movement of each motif is steered by the average of its constituent residues. To get the translation of each residue caused by the rigid anchor motif rotation, we define a virtual coordinate system with a rotation matrix of identity I and a translation of $\mathbf{X}_v = \overline{\mathbf{X}_{\mathcal{M}}}$.

3.3. Diffusion model for protein backbone generation

We follow the Riemannian score-based generative modeling of De Bortoli et al. (2022) and Yim et al. (2023). Denoising score matching (DSM) aims to approximate the Stein score $\nabla \log p_t(x)$, which is unavailable in practice, with a score network $s_\theta(t, \cdot)$ through minimizing the DSM loss:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\lambda_t \|\nabla \log p_{t|0}(\mathbf{X}^{(t)}|\mathbf{X}^{(0)}) - s_\theta(t, \mathbf{X}^{(t)})\|^2\right],$$

where $\mathbf{X}$, $p_{t|0}$, $\lambda_t > 0$, and θ denote the data distribution, the density of $\mathbf{X}^{(t)}$ given $\mathbf{X}^{(0)}$, a weight, and the network parameters, respectively. The expectation $\mathbb{E}$ is over the $t \sim \mathcal{U}([0, T_F])$ and $(\mathbf{X}^{(0)}, \mathbf{X}^{(t)})$.

3.4. Additional notations

The motif and scaffold parts of proteins are denoted by subscript $\mathcal{M}$ and $\mathcal{S}$ respectively. $\mathbf{R} = \{\mathbf{R}_{\mathcal{M}}, \mathbf{R}_{\mathcal{S}}\} = \{\mathbf{R}_{\mathcal{M}1}, \mathbf{R}_{\mathcal{M}2}, \cdots, \mathbf{R}_{\mathcal{M}m}, \mathbf{R}_{\mathcal{S}}\}$ and $\mathbf{X} = \{\mathbf{X}_{\mathcal{M}}, \mathbf{X}_{\mathcal{S}}\} = \{\mathbf{X}_{\mathcal{M}1}, \mathbf{X}_{\mathcal{M}2}, \cdots, \mathbf{X}_{\mathcal{M}m}, \mathbf{X}_{\mathcal{S}}\}$ denote the rotation and translation of all the residues in a protein. $\mathbf{T} = \{\mathbf{R}, \mathbf{X}\}$ denotes the position of residues. We denote noise, perturbation, and update by the notation with a δ subscript. For example, the noise to the rotation and translation is denoted as $\mathbf{R}_\delta$ and $\mathbf{X}_\delta$ respectively. The average operation $\bar{\cdot}$ over motifs indicates the average over each motif part, respectively. $[\cdot, \cdot]$ indicates the two elements, *i.e.* motif elements and scaffold elements.

4. Floating Anchor Diffusion

To tackle the challenge of generating scaffolds to support multiple motifs without prior knowledge of their relative positions, we propose a Floating Anchor Diffusion model (FADiff) as shown in Fig. 2. The core concept of FADiff lies in treating the motifs as rigid and enabling them to move rigidly. The motivation can be summarized as (1) For multiple motifs, their relative positions cannot be determined manually. Therefore, FADiff allows them to float independently in the diffusion process. (2) Since motif scaffolding needs the presence of motifs in the designed protein, FADiff preserves the structure of each motif as anchors rigidly. (3) Motifs are composed of amino acids, thus the movements are determined by their internal amino acids. Generally, we average the movement of residues inside each motif to steer it and make the whole process consistent with the diffusion process as shown in Fig. 1.

4.1. Forward diffuse the protein backbone

To model the forward diffusion process on protein backbone, $q(\mathbf{X}_{\mathcal{M}}^{(t)}, \mathbf{X}_{\mathcal{S}}^{(t)}|\mathbf{X}_{\mathcal{P}}^{(t-1)})$, we add noise to the frames following FrameDiff (Yim et al., 2023) but average the noise on motifs for treating the motif as rigid. We divide the transformation of frames into rotation and translation.

4.1.1. ROTATION

For a randomly sampled SO(3) rotation noise $\mathbf{R}_\delta$ to rotation $\mathbf{R}^{(t-1)}$, we estimate the movement of motifs with the average of their constituent residues, *i.e.*, the noise is reformed as:

$$\mathbf{R}_\delta{}' = \left[((\mathbf{R}^{(t-1)})^{-1}\overline{\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}}\mathbf{R}^{(t-1)})_\mathcal{M}, (\mathbf{R}_\delta)_\mathcal{S}\right] \quad (2)$$

The two items indicate the motif part and scaffold part respectively. The average operation $\bar{\cdot}$ over motifs indicates the average over each motif part respectively. The reformed noise in Eq. (2) is obtained as follows. With a randomly sampled noise $\mathbf{R}_\delta$, in order to maintain the rigidity of motifs, we estimate the movement of motifs with the average value of their constituent residues. Specifically, we define a virtual frame at the geometry center of motifs with a rotation matrix of identity I and a translation of $\mathbf{X}_v = \overline{\mathbf{X}_\mathcal{M}}$. Then we apply the noise to each residue to get the possible transformation of frames $\mathbf{R}^{(t)}{}' = \mathbf{R}^{(t-1)}\mathbf{R}_\delta$. The rotation transformation between the original and transformed motif in the virtual coordinate system is $\mathbf{R}^{(t)}{}'(\mathbf{R}^{(t-1)})^{-1} = \mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}$. To estimate the rotation of anchor motifs efficiently, we average the quaternion of its constituent residues, *i.e.*,

$$[\Delta\mathbf{R}_\mathcal{M}, \Delta\mathbf{R}_\mathcal{S}] = \left[(\overline{\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}})_\mathcal{M}, (\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1})_\mathcal{S}\right], \quad (3)$$

Without loss of generalization, any other rotation average methods can be applied here to estimate the rotation of motifs. Finally, we transform the rotation of the anchor motif under the coordinates of the virtual frame back to each residue as

$$\mathbf{R}^{(t)} = [\Delta\mathbf{R}_\mathcal{M}, \Delta\mathbf{R}_\mathcal{S}]\,\mathbf{R}^{(t-1)}. \quad (4)$$

The transform from $\mathbf{R}^{(t-1)}$ to $\mathbf{R}^{(t)}$ *i.e.*, the noise actually added to $\mathbf{R}^{(t-1)}$ is:

$$\mathbf{R}_\delta{}' = (\mathbf{R}^{(t-1)})^{-1}[\Delta\mathbf{R}_\mathcal{M}, \Delta\mathbf{R}_\mathcal{S}]\mathbf{R}^{(t-1)}, \quad (5)$$

which is used to calculate the rotation score. The details can be found in Appendix B.1. To get the translation of motifs' constituent residues, we first estimate the translation of their constituent residues caused by their rotations, which can be obtained as:

$$\Delta\mathbf{X}_\mathcal{M} = \Delta\mathbf{R}_\mathcal{M}(\mathbf{X}^{(t-1)}{}_\mathcal{M} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t-1)}{}_\mathcal{M} \quad (6)$$

4.1.2. TRANSLATION

Given a randomly sampled noise on translation $\mathbf{X}_\delta$, we also average the noise on each motif, respectively, as

$$\mathbf{X}_\delta{}' = \left[\overline{\mathbf{X}_\delta}_\mathcal{M}, \mathbf{X}_{\delta\mathcal{S}}\right], \quad (7)$$

which is used for translation score calculation. Then $\mathbf{X}^{(t)}$ is obtained through

$$\mathbf{X}^{(t)} = \mathbf{X}^{(t-1)} + \mathbf{X}_\delta{}'. \quad (8)$$

Finally, the noised data $\mathbf{T}$ for the score network to denoise is

$$\mathbf{T}^{(t)} = (\mathbf{R}^{(t)}, \mathbf{X}^{(t)} + [\Delta\mathbf{X}_\mathcal{M}, 0_\mathcal{S}]) \quad (9)$$

4.2. Denoising score matching

Given $\mathbf{T}^{(t)}$, the score network is designed to conduct iterative updates on the frames across a sequence of L layers, eventually yielding the predicted protein position $\hat{\mathbf{T}}^{(t-1)}$ (Jumper et al., 2021; Mao et al., 2024). Then the score is calculated with $\mathbf{T}^{(t)}$ and $\hat{\mathbf{T}}^{(t-1)}$. To keep the motifs rigid, we average the update for motifs like that in the forward diffusion process. When calculating the score, we remove the residue translations caused by the rigid rotation to keep consistent with the diffusion process.

4.2.1. FRAME UPDATE

Similar to the process in forward diffusion, we reform the predicted update $\hat{\mathbf{R}}_\delta$ on $\mathbf{R}^{(l-1)}$ as:

$$\hat{\mathbf{R}}'_\delta = \left[((\mathbf{R}^{(l-1)})^{-1}\overline{\mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(l-1)})^{-1}}\mathbf{R}^{(l-1)})_\mathcal{M}, (\hat{\mathbf{R}}_\delta)_\mathcal{S}\right] \quad (10)$$

we first estimate the rotation of each anchor motif with the average of its internal residue rotation. Then the rotation of each residue can be obtained as

$$\mathbf{R}^{(l)} = \left[\Delta\hat{\mathbf{R}}_\mathcal{M}, \Delta\hat{\mathbf{R}}_\mathcal{S}\right]\mathbf{R}^{(l-1)}, \quad (11)$$

$$[\Delta\hat{\mathbf{R}}_\mathcal{M}, \Delta\hat{\mathbf{R}}_\mathcal{S}] = \left[(\overline{\mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(l-1)})^{-1}})_\mathcal{M}, (\mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(l-1)})^{-1})_\mathcal{S}\right].$$

The translation of each residue caused by the rotation is:

$$\Delta\hat{\mathbf{X}}_\mathcal{M} = \Delta\hat{\mathbf{R}}_\mathcal{M}(\mathbf{X}^{(l-1)}{}_\mathcal{M} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(l-1)}{}_\mathcal{M}. \quad (12)$$

Then the $\mathbf{X}^{(l)}$ is derived with Eq. (1) as:

$$\mathbf{X}^{(l)} = \left[\overline{(\mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta)_\mathcal{M} - \Delta\hat{\mathbf{X}}_\mathcal{M}} + \Delta\hat{\mathbf{X}}_\mathcal{M}, (\mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta)_\mathcal{S}\right] + \mathbf{X}^{(l-1)} \quad (13)$$

The details of Eq. (13) can be found in the Appendix B.2. We adopt VFN-Diff (Mao et al., 2024), a SE(3) diffusion protein structure generation model as our score network in this work. Without loss of generalization, any other SE(3) diffusion model can be used as a score network here.

4.2.2. SCORE CALCULATION

To keep consistent with the score-based diffusion process (Song et al., 2021; Yim et al., 2023), we remove the residue translations caused by the rigid anchor motif rotation for the score calculation. Similar to the process above, we have the translation caused by rigid anchor motif rotation as:

$$\Delta \mathbf{X}'_{\mathcal{M}} = (\mathbf{R}^{(t)})^{-1}\hat{\mathbf{R}}^{(0)}(\mathbf{X}^{(t)} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t)}, \quad (14)$$

where $\hat{\mathbf{R}}^{(0)} = \mathbf{R}^{(l)}$ is the prediction of the network. Finally, the rotation and translation for score calculation are $\mathbf{R}^{(l)}$ and $\mathbf{X}^{(l)} - [\Delta \mathbf{X}'_{\mathcal{M}}, 0_{\mathbb{S}}]$. The details can be found at Appendix B.3.

4.3. Training loss

With the DSM loss in Eq. (4.3), the scheduler for rotation as $\lambda_t^r = 1/\mathbb{E}[\|\log p_{(t|0)}(\mathbf{R}_n^{(t)}|\mathbf{R}^{(0)})\|^2_{SO(3)}]$, and the scheduler for translation as $\lambda_t^x = (1 - e^{-t}/e^{-t/2})$ following Yim et al. (2023), we have the DSM loss as:

$$\mathcal{L}_{dsm} = \mathbb{E}\left[\lambda_t \|\nabla \log p_{t|0}(\mathbf{X}^{(t)}|\mathbf{X}^{(0)}) - s_\theta(t, \mathbf{X}^{(t)})\|^2\right],$$

which is consistent with the score-based diffusion model. More details can be found in the Appendix B.4.

Since the broken C-N bonds are found in early experiments, we have two auxiliary losses to get the distance between the atom C and N of two residues into the right range with Eq. (15) and to get the atoms in the right place with Eq. (16) following (Jumper et al., 2021; Yim et al., 2023) as follows:

$$\mathcal{L}_{c-n} = \frac{1}{4n}\sum_{i=1}^{n}\sum_{\mathbf{x}\in\Omega} \|\mathbf{x}_i^{(0)} - \hat{\mathbf{x}}_i^{(0)}\|^2, \quad (15)$$

$$\mathcal{L}_{bb} = \frac{1}{Z}\sum_{i,j=1}^{n}\sum_{a,b\in\Omega} \mathbb{1}\{d_{ab}^{ij} < 0.6\}\|d_{ab}^{ij} - \hat{d}_{ab}^{ij}\|^2, \quad (16)$$

$$Z = \left(\sum_{i,j=1}^{n}\sum_{a,b\in\Omega} \mathbb{1}\{d_{ab}^{ij} < 0.6\}\right) - n,$$

where Ω is the set of atoms $\{\mathrm{C}, \mathrm{C}_\alpha, \mathrm{O}, \mathrm{N}\}$. d_{ab}^{ij} and$\hat{d}_{ab}^{ij}$ indicate the ground truth and predicted distance between atom a and b in residue i and j. With $\mathbb{1}\{d_{ab}^{ij} < 0.6\}$, we leave alone the distances larger than 0.6Å. For more details in Appendix C.3

4.4. Sampling

Euler-Maruyama discretization with 500 steps implemented as a geodesic random walk is adopted in this work following De Bortoli et al. (2022); Yim et al. (2023); Watson et al. (2023). In this work, the sequence is constructed by finding the shortest chain, like the traveling salesman problem (TSP) where the distance is the gap between atoms C and N of two residues. More details are in the Appendix B.5.

5. Experiments

We trained FADiff on the task of scaffolding two motifs of lengths from 20 to 80 residues with the virtual motif (VM) dataset. We first analyze the performance of FADiff on the evaluation set of the VM dataset. Then we evaluate the performance and generalizability of FADiff on the multi-motif scaffolding (MS) Benchmark and analyze the generated samples in terms of designability. An ablation study is conducted to evaluate the effectiveness of TSP and noise scale. Finally, we compare our approaches with the conditional generation and inpainting methods which further demonstrate the efficacy of FADiff.

5.1. Setup

Dataset. Two datasets are utilized in this work, including the virtual motif dataset (**VM dataset**) from the PDB database (Berman et al., 2000) for training and the evaluation multi-motif scaffolding benchmark **MS Benchmark** that we collected from the PROSITE database(Sigrist et al., 2012). **VM dataset** contains 59,128 proteins with chain lengths from 60 to 512 residues extracted from the PDB database. For each entry, we randomly crop two fragments with lengths of 20 to 80 residues from the protein as virtual motifs for training. **MS Benchmark** contains 16,251 functional motifs with lengths between 10 and 20 residues (Xiong, 2006) that naturally exist.

Evaluation metrics. We mainly employ the self-consistence TM-score (**scTM**) to evaluate the *designablitity* of generated structures and the ***in silico* Success Rate** (**SR**) to evaluates the performance of the model following previous works (Trippe et al., 2023; Zhang & Skolnick, 2005; Ingraham et al., 2023). A higher TM-score or scTM indicates two structures are more similar. **scTM** is the TM-score between the generated structures and the reconstructed structure through ProteinMPNN (Dauparas et al., 2022) and ESMFold (Lin et al., 2023) as shown in Fig 7. The generated structure is *designable* if $\mathrm{scTM} > 0.5$. **SR** is the ratio of designable structures in the generation. Following previous work (Trippe et al., 2023; Zhang & Skolnick, 2005), for each group motif to be scaffolded, we generate 5 samples for each length of 160 to 410 residues and run ESMFold 8 times to get the highest scTM. **Motif RMSD**, *i.e.*, the difference between the desired motif and corresponding structure in the generated protein, used in conditional generation methods (Trippe et al., 2023) to evaluate the presence of motifs is not applicable here since it is *0* for our model consistently. Details can be found in Appendix D.2.

Compared approaches. We mainly compare FADiff with *conditional generation* methods (Trippe et al., 2023) and *inpainting* methods (Watson et al., 2023; Ingraham et al., 2023). We adapt inpainting methods to multiple motif scaf-

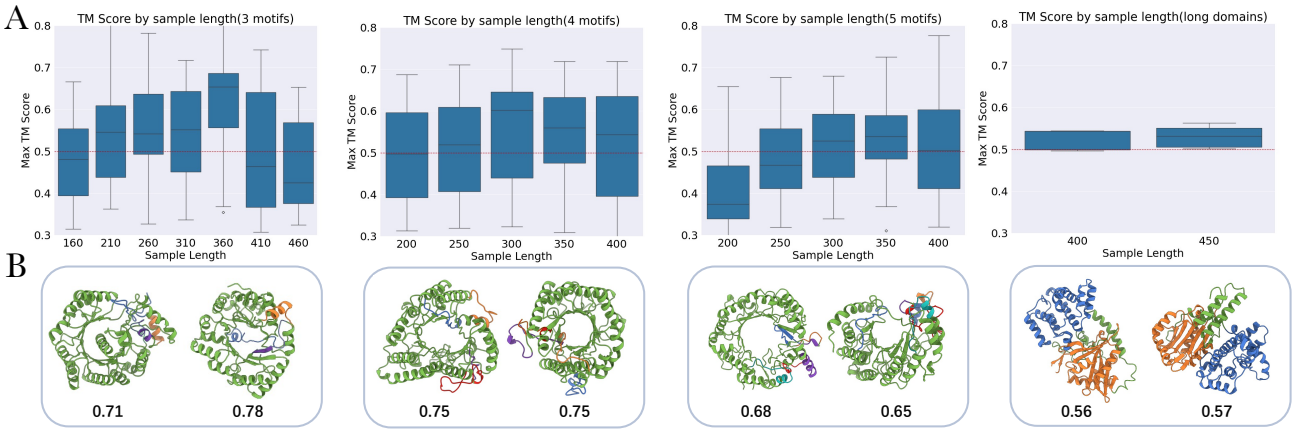

Figure 3. Statistic analysis and visualization of generation results for scaffolding 3, 4, 5, and two huge domains of length more than 100 residues. **A)** scTM distribution. The samples over the red dashed line are designable. 59.18%, 46.00%, 36.15%, and 60.00% generated protein structures are designable for scaffolding 3, 4, 5, and two huge domains. **B)** Generated protein structures. The green colors indicate the generated scaffolds and the other colors indicate the motifs. The numbers below each generated structure indicate the scTM score.

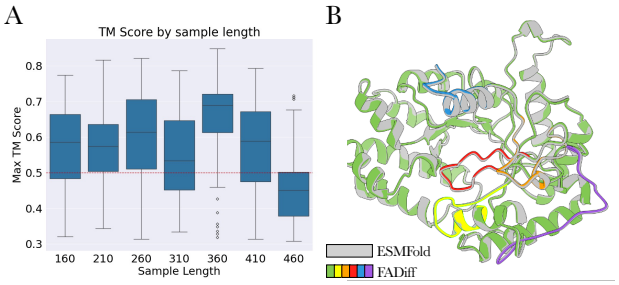

Figure 4. **A)** The distribution of scTM for inference results with varying lengths. The samples above the red dashed line are designable. **B)** visualization results were obtained by training the model on two motifs and testing on five motifs. 'ESMFold' denotes protein structures constructed through the ProteinMPNN and ESMFold, with a preference for closer structural resemblance. Our model demonstrates generalization capability.

folding by randomly assigning the relative structure position of two motifs since they require positions of motifs as input as *Remark* 3.3. Due to the probable absence of motifs in the conditional generation, one generation is successful if (1) $\mathrm{scTM} > 0.5$ and (2) the $\mathrm{motifRMSD} < 0.1$ following previous works (Trippe et al., 2023).

5.2. Experimental results

The evaluation of FADiff on the VM dataset demonstrates its ability to generate designable protein structures with a high tm-score. Experiments on the MS Dataset for two motifs demonstrate the efficacy of FADiff with a high ratio of designable structures in the generated samples. To evaluate the generalizability of FADiff, experiments on scaffolding more than two motifs are conducted with FADiff trained on the two-motif scaffolding task. Finally, an ablation study is conducted to demonstrate the effectiveness of the choice in implementation.

5.2.1. Evaluation on VM dataset

The TM-score between the protein structure where the virtual motifs are located and the structures generated by FADiff in the evaluation set of the VM dataset is shown in Fig. 8. The TM-score of 67.5% generated structures is above 0.5 which indicates that the generated structure and the original structures are similar, demonstrating the ability of FADiff to generate designable protein structures. Besides, the novelty of generated proteins provides insight into a way to design novel proteins, *i.e.*, scaffolding motifs, as shown in Table 4. More details can be found in Appendix D.3.1.

5.2.2. Evaluation on MS Benchmark

Scaffold two motifs. We evaluate FADiff on the task of scaffolding two functional motifs from the MS Benchmark. The ratio of designable protein structures generated by FADiff is 73.05% as shown in Fig. 4A. The performance of FADiff varies with the length of the generated protein due to the bias of training data. The distribution of proteins varies with their lengths in the VM dataset.FADiff also demonstrates commendable performance on scaffolding 5 motifs, as depicted in Fig. 4B.

Generalization of FADiff. To evaluate the generalization of FADiff, we apply FADiff trained on the task of scaffolding two virtual motifs to scaffold 3, 4, 5, and two huge domains as shown in Fig. 3. (1) **Scaffold more than two motifs**: The average *in silico* success rates of FADiff for scaffolding 3, 4, and 5 motifs achieve 62.38%, 58.40%,

Table 1. In silico success rate (%) for different lengths, with/without TSP, and different translation noise scales in sampling. TSP and Random indicate the preprocessing method. $/2$ and $\times 2$ indicate the translation noise scale in sampling.

Method	160	210	260	310	360
FADiff	**76.67**	71.67	**81.67**	65.00	**86.67**
Random	49.33	60.00	60.00	54.67	82.67
$/2$	69.23	**76.92**	76.92	60.00	83.08
$\times 2$	70.00	70.00	75.00	**78.33**	83.33

46.67%. (2) **Scaffold huge domains**: We further design scaffolds for two huge domains with lengths of more than 100 residues which are also never trained with. Domains are also functional parts of proteins. The average *in silico* success rate for two huge domains achieves 80.00%, which further demonstrates the generalization of FADiff. The generalization is achieved since FADiff treats all the residues equally. The decrease in success rate for scaffolding more motifs is caused by the increasing difficulty, especially with fewer scaffold residues.

5.3. Ablation study

Noise scale on translation. Since the motifs are much bigger than residues, it is straightforward to increase the translation noise to enable the residues of scaffolds to navigate in the scale of motifs. We train a model without increasing the noise scale on translation and test it on scaffolding two motifs. The average scTM is 0.213 and the average SR is close to 0% due to significant translation prediction errors. Please refer to Appendix C.1 for more details.

TSP for sequence construction. To evaluate the efficacy of TSP, we randomly connect the residues to construct the amino acid sequence. Although TSP outperforms the random connection consistently, the random connection also leads to a high average *in silico* success rate of 60.67% as shown in Table 1. More details are in Appendix D.3.3.

Noise scale in sampling. With different noise scales on translation in sampling, the performance of FADiff varies little and achieves a high average *in silico* success rate of 72.56% and 72.22% for the reducing by half ($/2$) and augmenting by twice the noise scale ($\times 2$).

5.4. Comparison

We compare FADiff with conditional generation and inpainting methods. The inpainting is adopted to scaffold multiple motifs by randomly assigning their relative positions. One generation is successful for the conditional generation method if the motif RMSD is less than 1 and $\mathrm{scTM} > 0.5$ following Trippe et al. (2023). FADiff outperforms inpainting and conditional generation consistently in the success rate of scaffolding two motifs over all different lengths of proteins as shown in Table 2. The generated structures by conditional generation have high scTM scores while the existence of desired motifs is not ensured. 87.25% of the generated structures' scTM scores are over 0.5, indicating they are designable. However, the motif RMSD of only 21.50% generated structures is under 0.1, which indicates the absence of desired motifs in the generation. Inpainting outperforms the conditional generation methods since the presence of motifs in the generation is guaranteed. However, the randomly assigned inappropriate relative positions of motifs lead to the failure in the generation. With FADiff, we ensure the existence and automate the design of motif relative positions by enabling the motifs to float rigidly.

Table 2. In silico success rate (%) for different lengths with different scaffolding methods, where Condition indicates the conditional generation method.

Method	160	210	260	310	360
FADiff	**76.67**	**71.67**	**81.67**	**65.00**	**86.67**
Inpainting	51.58	56.84	62.11	60.00	79.47
Condition	23.75	22.50	21.25	23.75	16.25

For more results on specific motifs and case studies, please refer to Appendix D.3.

6. Discussion

Why FADiff operate effectively without the expertise on the relative positions of multiple motifs? Previous works fail to scaffold multiple motifs due to the strong correlation between sequence position and structure position. In previous works, like RFDiffusion, the positions of multiple motifs on the amino sequence $\mathcal{A}$ and in the protein structure $\mathbf{X}$ should be specified manually. Since the protein structure $\mathbf{X}$ is determined by the amino sequence $\mathcal{A}$, *i.e.* $\mathbf{X}(\mathcal{A})$ and $\mathbf{X}_{\mathcal{M}}(\mathcal{A}_{\mathcal{M}})$, the manually assigned positions are almost unable to achieve the correlation. However, FADiff allows the anchor motifs to float to a rational position as a rigid, enabling the automatic design of relative positions of multiple motifs. Therefore, even with a random connection to construct the amino acid sequence, FADiff also achieves a high success rate by steering the motifs to a rational position determined by the sequence.

How does FADiff generalize to scaffold multiple motifs? In both the diffusion and reverse processes, all the residues of motifs and scaffolds are considered equally for FADiff. Only for the update of residue positions, we average the movement of motif residues to steer the motifs. The whole process can be considered consistent with the dif-

fusion model for protein backbone generation. A FADiff trained on scaffolding two virtual motifs of lengths from 20 to 80 can be applied to scaffolding more than two motifs and huge domains with a length of more than 100 residues.

7. Conclusion

To tackle the challenge of automatically designing relative positions and preserving the presence of multiple motifs. We propose a Floating Anchor Diffusion (FADiff) model for scaffolding multiple motifs for the first time. FADiff solves the problem by taking the anchor motifs as rigid respectively and allowing them to float flexibly. Our experiments on the benchmark demonstrate the efficacy and generalization of FADiff, providing insights for future wet experiments and a new way to construct novel protein structures. It is straightforward to apply FADiff to other generation tasks where multiple substructures should be preserved while their positions in the generation are flexible.

Impact Statement

The goal of this work is to advance the field of Machine Learning and Computational Biology. While there are many potential societal consequences of our work, we believe that none of which must be specifically highlighted here.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research*, 28(1): 235–242, 01 2000. ISSN 0305-1048.

Cao, L., Coventry, B., Goreshnik, I., Huang, B., Sheffler, W., Park, J. S., Jude, K. M., Marković, I., Kadam, R. U., Verschueren, K. H., et al. Design of protein-binding proteins from the target structure alone. *Nature*, 605 (7910):551–560, 2022.

Correia, B. E., Bates, J. T., Loomis, R. J., Baneyx, G., Carrico, C., Jardine, J. G., Rupert, P., Correnti, C., Kalyuzhniy, O., Vittal, V., et al. Proof of principle for epitope-focused vaccine design. *Nature*, 507(7491):201–206, 2014.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Davila-Hernandez, F. A., Jin, B., Pyles, H., Zhang, S., Wang, Z., Huddy, T. F., Bera, A. K., Kang, A., Chen, C.-L., De Yoreo, J. J., et al. Directing polymorph specific calcium carbonate formation with de novo protein templates. *Nature Communications*, 14(1):8191, 2023.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

Gruver, N., Stanton, S. D., Frey, N. C., Rudner, T. G. J., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Herbert, A. and Sternberg, M. Maxcluster: a tool for protein structure comparison and clustering, 2008.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35: 2750–2761, 2022.

Hutchinson, E. G. and Thornton, J. M. Promotif—a program to identify and analyze structural motifs in proteins. *Protein Science*, 5(2):212–220, 1996.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

Jiang, H., Jude, K. M., Wu, K., Fallas, J., Ueda, G., Brunette, T., Hicks, D., Pyles, H., Yang, A., Carter, L., et al. De novo design of buttressed loops for sculpting protein functions. *bioRxiv*, 2023.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Lee, J. S., Kim, J., and Kim, P. M. Score-based generative modeling for de novo protein design. *Nature Computational Science*, pp. 1–11, 2023.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Linsky, T. W., Vergara, R., Codina, N., Nelson, J. W., Walker, M. J., Su, W., Barnes, C. O., Hsiang, T.-Y., Esser-Nobis, K., Yu, K., et al. De novo design of potent and resilient hace2 decoys to neutralize sars-cov-2. *Science*, 370(6521):1208–1214, 2020.

Lisanza, S. L., Gershon, J. M., Tipps, S. W. K., Arnoldt, L., Hendel, S., Sims, J. N., Li, X., and Baker, D. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. *bioRxiv*, pp. 2023–05, 2023.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., and Naik, N. V. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pp. 1–8, 2023. URL https://api.semanticscholar.org/CorpusID:256304602.

Mao, W., Zhu, M., Sun, Z., Shen, S., Wu, L. Y., Chen, H., and Shen, C. De novo protein design using geometric vector field networks. *Proc. Int. Conf. Learning Representations*, 2024.

Pawson, T. and Scott, J. D. Signaling through scaffold, anchoring, and adaptor proteins. *Science*, 278(5346):2075–2080, 1997. URL https://www.science.org/doi/abs/10.1126/science.278.5346.2075.

Procko, E., Berguig, G. Y., Shen, B. W., Song, Y., Frayo, S., Convertine, A. J., Margineantu, D., Booth, G., Correia, B. E., Cheng, Y., et al. A computationally designed inhibitor of an epstein-barr viral bcl-2 protein induces apoptosis in infected cells. *Cell*, 157(7):1644–1656, 2014.

Roel-Touris, J., Nadal, M., and Marcos, E. Single-chain dimers from de novo immunoglobulins as robust scaffolds for multiple binding loops. *Nature Communications*, 14 (1):5939, 2023.

Roy, A., Shi, L., Chang, A., Dong, X., Fernandez, A., Kraft, J. C., Li, J., Le, V. Q., Winegar, R. V., Cherf, G. M., et al. De novo design of highly selective miniprotein inhibitors of integrins αvβ6 and αvβ8. *Nature Communications*, 14(1):5660, 2023.

Sesterhenn, F., Yang, C., Bonet, J., Cramer, J. T., Wen, X., Wang, Y., Chiang, C.-I., Abriata, L. A., Kucharska, I., Castoro, G., et al. De novo protein design enables the precise induction of rsv-neutralizing antibodies. *Science*, 368(6492):eaay5051, 2020.

Siegel, J. B., Zanghellini, A., Lovick, H. M., Kiss, G., Lambert, A. R., St. Clair, J. L., Gallaher, J. L., Hilvert, D., Gelb, M. H., Stoddard, B. L., et al. Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. *Science*, 329(5989):309–313, 2010.

Sigrist, C. J. A., de Castro, E., Cerutti, L., Cuche, B. A., Hulo, N., Bridge, A., Bougueleret, L., and Xenarios, I. New and continuing developments at PROSITE. *Nucleic Acids Research*, 41(D1):D344–D347, 11 2012. ISSN 0305-1048. doi: 10.1093/nar/gks1067. URL https://doi.org/10.1093/nar/gks1067.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6TxBxqNME1Y.

Wang, J., Lisanza, S., Juergens, D., Tischer, D., Watson, J. L., Castro, K. M., Ragotte, R., Saragovi, A., Milles, L. F., Baek, M., et al. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, 2022.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Xiong, J. *Essential bioinformatics*. Cambridge University Press, 2006.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Zhang, Y. and Skolnick, J. Tm-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids research*, 33(7):2302–2309, 2005.

A. Notations

The notations in this paper follow the principle that $\mathcal{M}$ with a subscript describes the structure of Motifs solely and $\mathcal{M}$ as a subscript describes the position or structure of motifs in the designed protein. The notations used in this paper are described in Table. 3 and Fig. 5.

Protein & Motif	
$\mathcal{P}$	A protein with both structure and sequence
$\mathcal{A}$	Protein amino acid sequence. $\mathcal{A} = \{\mathcal{C}^n, \mathcal{D}\}$
$\mathbf{X}$	Protein 3D structure
$a_i \in \mathcal{C}^{20}$	The i-th amino acid type
$\mathbf{x}_i \in \mathbb{R}^3$	The i-th C-α residue backbone coordinates in 3D
$\mathcal{D}$	Sequence position. The index of amino acids in the sequence
$\mathcal{M}_\mathcal{P}$	Scaffolded motif in a protein
$\mathcal{M}_\mathbf{X}$	Internal structure of motif without scaffolding
$\mathcal{M}_\mathcal{A}$	Internal sequence of motif without scaffolding
$\mathcal{A}_\mathcal{M}$	Scaffolded motif position in the protein sequence
$\mathbf{X}_\mathcal{M}$	Scaffolded motif position in the protein structure
$\mathcal{S}_\mathcal{P}$	Scaffolding in a protein
$\mathcal{S}_\mathbf{X}$	Internal structure of scaffolding without motif
$\mathcal{S}_\mathcal{A}$	Internal sequence of scaffolding without motif
$\mathcal{A}_\mathcal{S}$	Scaffolding position in the protein sequence with motif
$\mathbf{X}_\mathcal{S}$	Scaffolding position in the protein structure with motif
Parameterization	
$\mathbf{T} \in \mathbb{R}^{4\times4}$	Residue portions (transformation). Orientation preserving rigid transformation (*frame*)
$\mathbf{R}_i \in \mathbb{R}^{3\times3}$	Rotation
$\mathbf{X}_i \in \mathbb{R}^3$	Translation

Table 3. Notations for FADiff

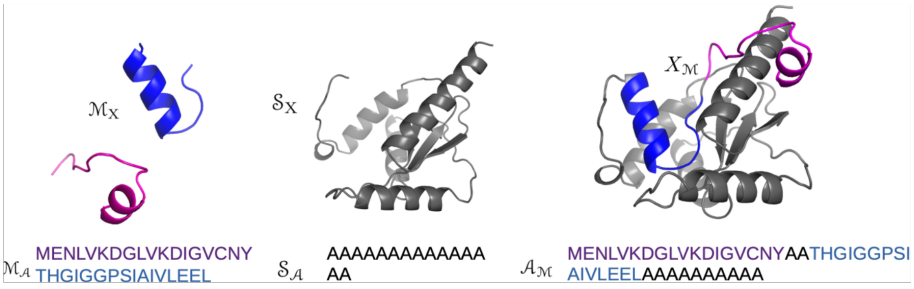

Figure 5. Illustration of notations

B. Method details

B.1. Noise on rotation

Given the rotation $\mathbf{R}^{(t)}$ and $\mathbf{R}^{(t-1)}$, we can calculate the transformation from $\mathbf{R}^{(t-1)}$ to $\mathbf{R}^{(t)}$ as:

$$\mathbf{R}_\delta{}' = (\mathbf{R}^{(t-1)})^{-1}\mathbf{R}^{(t)}. \tag{17}$$

With Eq. (4), the equation above is further derived as:

$$\mathbf{R}_\delta{}' = (\mathbf{R}^{(t-1)})^{-1}\left[\Delta\mathbf{R}_\mathcal{M}, \Delta\mathbf{R}_\mathcal{S}\right]\mathbf{R}^{(t-1)} \tag{18}$$

The transformation is consistent with the diffusion process of FrameDiff. With Eq. (3), we have:

$$\begin{aligned}
\mathbf{R}_\delta{}' &= (\mathbf{R}^{(t-1)})^{-1}\left[(\overline{\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}})_{\mathcal{M}}, (\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1})_{\mathcal{S}}\right]\mathbf{R}^{(t-1)} \\
&= \left[\left((\mathbf{R}^{(t-1)})^{-1}\overline{\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}}\mathbf{R}^{(t-1)}\right)_{\mathcal{M}}, ((\mathbf{R}^{(t-1)})^{-1}\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1})\mathbf{R}^{(t-1)}{}_{\mathcal{S}}\right] \\
&= \left[\left((\mathbf{R}^{(t-1)})^{-1}\overline{\mathbf{R}^{(t-1)}\mathbf{R}_\delta(\mathbf{R}^{(t-1)})^{-1}}\mathbf{R}^{(t-1)}\right)_{\mathcal{M}}, (\mathbf{R}_\delta)_{\mathcal{S}}\right]
\end{aligned}$$

B.2. Translation update

Eq. (13) is derived as follows: Since the model gives an update under the coordinate system of the local frame. We first get the update of translation under the fixed coordinate system through Eq. (1) as:

$$\hat{\mathbf{X}}_\delta^{world} = \mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta$$

There are two parts of translation updates: (1) the model expected translation update $\hat{\mathbf{X}}_\delta^{world}$ (Eq. (B.2)), and (2) the rigid motif rotation caused translation $\Delta\hat{\mathbf{X}}_{\mathcal{M}}$ (Eq. (12)). We believe the model's expected translation update is translating the residues to rational positions. In contrast, the translation from the rigid anchor motif rotation interferes with the movement to the rational position. Therefore, we remove the translation update caused by rotation, then average them on the motif residues as:

$$\hat{\mathbf{X}}_\delta^{update} = [(\overline{\hat{\mathbf{X}}_\delta^{world} - \Delta\hat{\mathbf{X}}_{\mathcal{M}}})_{\mathcal{M}}, (\hat{\mathbf{X}}_\delta^{world})_{\mathcal{S}}] \tag{19}$$

Finally, we add the translation update to $\mathbf{X}^{(l-1)}$ to get $\mathbf{X}^{(l)}$ as:

$$\begin{aligned}
\mathbf{X}^{(l)} &= \hat{\mathbf{X}}_\delta^{update} + \mathbf{X}^{(l-1)} \\
&\overset{Eq.\ (19)}{=} [(\overline{\hat{\mathbf{X}}_\delta^{world} - \Delta\hat{\mathbf{X}}_{\mathcal{M}}})_{\mathcal{M}}, (\mathbf{X}_\delta^{world})_{\mathcal{S}}] + \mathbf{X}^{(l-1)} \\
&\overset{Eq.\ (B.2)}{=} \left[\overline{(\mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta)_{\mathcal{M}} - \Delta\hat{\mathbf{X}}_{\mathcal{M}}} + \Delta\hat{\mathbf{X}}_{\mathcal{M}}, (\mathbf{R}^{(t-1)}\hat{\mathbf{X}}_\delta)_{\mathcal{S}}\right] + \mathbf{X}^{(l-1)}.
\end{aligned}$$

B.3. Translation for score calculation

Given the predicted rotation matrix $\hat{\mathbf{R}}^{(0)}$ from the model, the noised rotation matrix $\mathbf{R}^{(t)}$, and the noised translation $\mathbf{X}^{(t)}$, we can derive the translation caused by the rigid anchor motif as follows: The rotation from $t = 0$ to $t = t$ is:

$$\mathbf{R}_{0\to t} = (\hat{\mathbf{R}}^{(0)})^{-1}\mathbf{R}^{(t)} \tag{20}$$

Then the translation caused by the rotation is:

$$\begin{aligned}
\Delta\mathbf{X}_{\mathcal{M}}{}' &= \mathbf{R}_{0\to t}^{-1}(\mathbf{X}^{(t)} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t)} \\
&\overset{Eq.\ (20)}{=} ((\hat{\mathbf{R}}^{(0)})^{-1}\mathbf{R}^{(t)})^{-1}(\mathbf{X}^{(t)} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t)} \\
&= (\mathbf{R}^{(t)})^{-1}\hat{\mathbf{R}}^{(0)}(\mathbf{X}^{(t)} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t)}
\end{aligned}$$

B.4. Relationship between FADiff and other score-based diffusion models

In this work, we adopt the FrameDiff (Yim et al., 2023) as the pipeline for the diffusion process. Without loss of generalization, any other score-based diffusion models can be used here. We explain that FADiff is consistent with FrameDiff below First, we review the FrameDiff. In the forward diffusion process, the noise is added to translation and rotation as:

$$\begin{aligned}
\mathbf{R}^{(t)} &= \mathbf{R}^{(0)}\mathbf{R}_\delta(t) \\
\mathbf{X}^{(t)} &= \mathbf{X}^{(0)} + \mathbf{X}_\delta(t),
\end{aligned} \tag{21}$$

where the score s for translation and rotation can be calculated as:

$$\begin{aligned}
s_{\mathbf{r}}^{Frame} &= \text{score}_{\mathbf{r}}(\mathbf{R}_\delta(t), t) \\
s_{\mathbf{x}}^{Frame} &= \text{score}_{\mathbf{x}}(\mathbf{X}^{(t)}, \mathbf{X}^{(0)}, t).
\end{aligned} \tag{22}$$

The score network gives the predicted denoised rotation and translation $\hat{\mathbf{R}}^{(0)}$ and $\hat{\mathbf{X}}^{(0)}$, with which the score can be calculated as:

$$s_{\mathbf{r}}'^{Frame} = \text{score}_{\mathbf{r}}((\hat{\mathbf{R}}^{(0)})^{-1}\mathbf{R}^{(t)}, t)$$
$$s_{\mathbf{x}}'^{Frame} = \text{score}_{\mathbf{x}}(\mathbf{X}^{(t)}, \hat{\mathbf{X}}^{(0)}, t).$$

Since all the layers for the update are the same, we take the last layer for example, and then $\hat{\mathbf{R}}^{(t-1)}$ can be derived as:

$$\mathbf{R}^{(l)} = \mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta \tag{23}$$
$$\hat{\mathbf{R}}^{(0)} = \mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta \tag{24}$$

Then the score can be calculated as:

$$s_{\mathbf{r}}'^{Frame} = \text{score}_{\mathbf{r}}(\hat{\mathbf{R}}_\delta^{-1}, t)$$
$$s_{\mathbf{x}}'^{Frame} = \text{score}_{\mathbf{x}}(\mathbf{X}^{(t)}, \hat{\mathbf{X}}^{(0)}, t). \tag{25}$$

FrameDiff seeks to match $s_{\mathbf{r}}$ and $s_{\mathbf{x}}$ to $s_{\mathbf{r}}'$ and $s_{\mathbf{x}}'$. The object to optimize is:

$$\begin{aligned} &\min \quad \|s_{\mathbf{r}} - s_{\mathbf{r}}'\|^2 + \|s_{\mathbf{x}} - s_{\mathbf{x}}'\|^2 \\ \min \quad &\|\text{score}_{\mathbf{r}}(\mathbf{R}_\delta(t), t) - \text{score}_{\mathbf{r}}(\hat{\mathbf{R}}_\delta^{-1}, t)\|^2 + \|\text{score}_{\mathbf{x}}(\mathbf{X}^{(t)}, \mathbf{X}^{(0)}, t) - \text{score}_{\mathbf{x}}(\mathbf{X}^{(t)}, \hat{\mathbf{X}}^{(0)}, t)\|^2 \end{aligned} \tag{26}$$

For an ideal FrameDiff, $\hat{\mathbf{R}}_\delta = {\mathbf{R}_\delta}^{-1}$ and $\hat{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$. To keep consistent with FrameDiff, the score network should predict $\hat{\mathbf{R}}_\delta = {\mathbf{R}_\delta}^{-1}$ and the whole model should predict the $\hat{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$.

Generally, the model should predict the noise added to the rotation and translation in the forward diffusion process. In FADiff, we get consistency by making the model predict the noise actually added to the original data, *i.e.*, the ${\mathbf{R}_\delta}'$ and ${\mathbf{X}_\delta}'$. The whole process is listed below. In the forward process, the rotation and translation noise for each residue of motifs are:

$$\begin{aligned} {\mathbf{R}_\delta}' &= \left[\left((\mathbf{R}^{(0)})^{-1}\overline{\mathbf{R}^{(0)}\mathbf{R}_\delta(\mathbf{R}^{(0)})^{-1}}\mathbf{R}^{(0)}\right)_{\mathcal{M}}, (\mathbf{R}_\delta)_{\mathcal{S}}\right], \\ {\mathbf{X}_\delta}' &= \left[\overline{\mathbf{X}_\delta}_{\mathcal{M}}, \mathbf{X}_{\delta\,\mathcal{S}}\right]. \end{aligned} \tag{27}$$

Then the noised data is derived as:

$$\begin{aligned} \mathbf{R}^{(t)} &= \left[(\overline{\mathbf{R}^{(0)}\mathbf{R}_\delta(\mathbf{R}^{(0)})^{-1}}\mathbf{R}^{(0)})_{\mathcal{M}}, (\mathbf{R}^{(0)}\mathbf{R}_\delta)_{\mathcal{S}}\right], \\ \mathbf{X}^{(t)} &= \left[\left(\mathbf{X}^{(0)} + \Delta\mathbf{X}_{\mathcal{M}} + \overline{\mathbf{X}_\delta}\right)_{\mathcal{M}}, \left(\mathbf{X}^{(0)} + \mathbf{X}_\delta\right)_{\mathcal{S}}\right], \end{aligned} \tag{28}$$

where $\Delta\mathbf{X}_{\mathcal{M}}$ is the translation caused by the rotation of rigid anchor motif as Eq. (6) and Eq. (2):

$$\Delta\mathbf{X}_{\mathcal{M}} = (\overline{\mathbf{R}^{(0)}\mathbf{R}_\delta(\mathbf{R}^{(0)})^{-1}})_{\mathcal{M}}({\mathbf{X}^{(0)}}_{\mathcal{M}} - \mathbf{X}_v) + \mathbf{X}_v - {\mathbf{X}^{(0)}}_{\mathcal{M}}$$

The score is calculated as:

$$\begin{aligned} s_{\mathbf{r}}^{FA} &= \text{score}_{\mathbf{r}}(\left[\left((\mathbf{R}^{(0)})^{-1}\overline{\mathbf{R}^{(0)}\mathbf{R}_\delta(\mathbf{R}^{(0)})^{-1}}\mathbf{R}^{(0)}\right)_{\mathcal{M}}, (\mathbf{R}_\delta)_{\mathcal{S}}\right], t), \\ s_{\mathbf{x}}^{FA} &= \text{score}_{\mathbf{x}}(\left[\left(\mathbf{X}^{(0)} + \overline{\mathbf{X}_\delta}\right)_{\mathcal{M}}, \left(\mathbf{X}^{(0)} + \mathbf{X}_\delta\right)_{\mathcal{S}}\right], \mathbf{X}^{(0)}, t), \end{aligned} \tag{29}$$

without the translation caused by the rotation. The score network gives the predicted denoised rotation and translation $\hat{\mathbf{R}}^{(t-1)}$ and $\hat{\mathbf{X}}^{(t-1)}$.

The predicted rotation and translation is the updated frame from the last layer as Eq. (11) and Eq. (13):

$$\begin{aligned} \mathbf{R}^{(l)} &= \left[(\overline{\mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(l-1)})^{-1}})_{\mathcal{M}}, (\mathbf{R}^{(l-1)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(l-1)})^{-1})_{\mathcal{S}}\right]\mathbf{R}^{(l-1)} \\ \mathbf{X}^{(l)} &= \left[\overline{(\mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta)_{\mathcal{M}} - \Delta\hat{\mathbf{X}}_{\mathcal{M}}} + \Delta\hat{\mathbf{X}}_{\mathcal{M}}, (\mathbf{R}^{(l-1)}\hat{\mathbf{X}}_\delta)_{\mathcal{S}}\right] + \mathbf{X}^{(l-1)} \end{aligned} \tag{30}$$

Since all the layers for the update are the same, we take the last layer for example, and then the equation above is derived as:

$$\hat{\mathbf{R}}^{(0)} = \left[\left(\overline{\mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(t)})^{-1}}\mathbf{R}^{(t)}\right)_{\mathcal{M}}, (\mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta)_{\mathcal{S}}\right] \tag{31}$$

$$\hat{\mathbf{X}}^{(0)} = \left[\overline{(\mathbf{R}^{(t)}\hat{\mathbf{X}}_\delta)_{\mathcal{M}} - \Delta\hat{\mathbf{X}}_{\mathcal{M}}} + \Delta\hat{\mathbf{X}}_{\mathcal{M}}, (\mathbf{R}^{(t)}\hat{\mathbf{X}}_\delta)_{\mathcal{S}}\right] + \mathbf{X}^{(t)} \tag{32}$$

With Eq. (14), the translation of each residue caused by the rotation is:

$$\Delta\hat{\mathbf{X}}_{\mathcal{M}} = (\mathbf{R}^{(t)})^{-1}\hat{\mathbf{R}}^{(0)}(\mathbf{X}^{(t)} - \mathbf{X}_v) + \mathbf{X}_v - \mathbf{X}^{(t)}. \tag{33}$$

Finally, the predicted scores from the model are :

$$\begin{aligned} s_{\mathbf{r}}'^{FA} &= \text{score}_{\mathbf{r}}\left(\left[\left(\left(\overline{\mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(t)})^{-1}}\mathbf{R}^{(t)}\right)^{-1}_{\mathcal{M}}\mathbf{R}^{(t)}\right)_{\mathcal{M}}, (((\mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta)_{\mathcal{S}})^{-1}\mathbf{R}^{(t)})_{\mathcal{S}}\right], t\right), \\ &= \text{score}_{\mathbf{r}}\left(\left[\left(\left(\overline{\mathbf{R}^{(t)}\hat{\mathbf{R}}_\delta(\mathbf{R}^{(t)})^{-1}}\mathbf{R}^{(t)}\right)^{-1}_{\mathcal{M}}\mathbf{R}^{(t)}\right)_{\mathcal{M}}, (\hat{\mathbf{R}}_\delta^{-1})_{\mathcal{S}}\right], t\right), \end{aligned} \tag{34}$$

$$\begin{aligned} s_{\mathbf{x}}'^{FA} &= \text{score}_{\mathbf{x}}\left(\left[\left(\mathbf{X}^{(0)} + \overline{\mathbf{X}_\delta}\right)_{\mathcal{M}}, \left(\mathbf{X}^{(0)} + \mathbf{X}_\delta\right)_{\mathcal{S}}\right], \hat{\mathbf{X}}^{(0)} - [\Delta\hat{\mathbf{X}}_{\mathcal{M}}, 0_{\mathcal{S}}], t\right) \\ &= \text{score}_{\mathbf{x}}\left(\left[\left(\mathbf{X}^{(0)} + \overline{\mathbf{X}_\delta}\right)_{\mathcal{M}}, \left(\mathbf{X}^{(0)} + \mathbf{X}_\delta\right)_{\mathcal{S}}\right], [(\hat{\mathbf{X}}^{(0)} - \Delta\hat{\mathbf{X}})_{\mathcal{M}}, \hat{\mathbf{X}}^{(0)}_{\mathcal{S}}], t\right) \end{aligned} \tag{35}$$

We can explain the derivation in two parts. The scaffolding part is consistent with FrameDiff obviously with Eq. (34) and Eq. (35), which is the same as FrameDiff. For the motif part, substituting Eq. (28), Eq. (31) and Eq. (33) into Eq. (34) and Eq. (35) yields the same form as Eq. (29). When we relax the equation with the average to be themselves, and further substitute $\hat{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$ and $\hat{\mathbf{R}}_\delta = \mathbf{R}_\delta{}^{-1}$ into the equation, the equation gets the same as Eq. (29).

B.5. Sampling

B.5.1. TSP FOR SEQUENCE CONSTRUCTION

In sampling, we randomly sample residues of the scaffold with a Gaussian distribution to decide their translation and rotation. The motifs are put at the origin of the fixed coordinate. Then we construct a distance map by calculating the distance between the atoms C and N of two residues, which should be the length of the peptide bond in a naturally existing protein. Besides, to maintain the motif structure, we change the distance between two residues connected in the motif to *0* and the distance from the scaffold residues to the connected motif residues to be infinite. Finally, with a greedy algorithm to find the shortest chain like the TSP, a sequence is obtained.

B.5.2. RANDOM CONNECTION FOR SEQUENCE CONSTRUCTION

In sampling, we randomly sample residues of the scaffold with a Gaussian distribution to decide their translation and rotation. The motifs are put at the origin of the fixed coordinate. Then we construct a distance map where we set the distance between two residues connected in the motif to *0* and the distance from the scaffold residues to the connected motif residues to be infinite to maintain the motif structure. And the distances between other residues are given randomly. Finally, with a greedy algorithm to find the shortest chain like the TSP, a sequence is obtained.

C. Training details

C.1. Hyper-parameters

We follow the FrameDiff (Yim et al., 2023) for all the parameters except the coordinate scale. We train FADiff for 90,000 steps with a coordinate scale of 0.1 and 0.02 based on the pre-trained VFN-Diff (Mao et al., 2024). Here the coordinate scale c is used in the adding noise stage:

$$\mathbf{x}^t = c \cdot (\mathbf{x}^{(t-1)} \cdot c + \mathbf{x}_\delta),$$

where x_δ is the Gaussian noise. The coordinate scale of 0.1 follows Yim et al. (2023); however, the model trained with the coordinate scale of 0.02 works much better since the coordinate scale is much larger than a single amino acid.

C.2. Hardware

We train FAdiff for 90,000 steps in around 20 hours. All our experiments are conducted on a computing cluster with 8 GPUs of NVIDIA GeForce RTX 4090 24GB and CPUs of AMD EPYC 7763 64-Core of 3.52GHz. All the inferences are conducted on a single GPU of NVIDIA GeForce RTX 4090 24GB.

C.3. Training loss

Since the method of FADiff is general, any other score-based diffusion model can be adopted as our backbone. All the training loss weights and other settings remain the same except the coordinate scale as explained in the Appendix C.1

D. Experiment

D.1. Dataset Detail

D.1.1. Distribution of sequence length of the VM dataset

The number of proteins in the VM dataset varies with the sequence length as shown in Fig. 6.

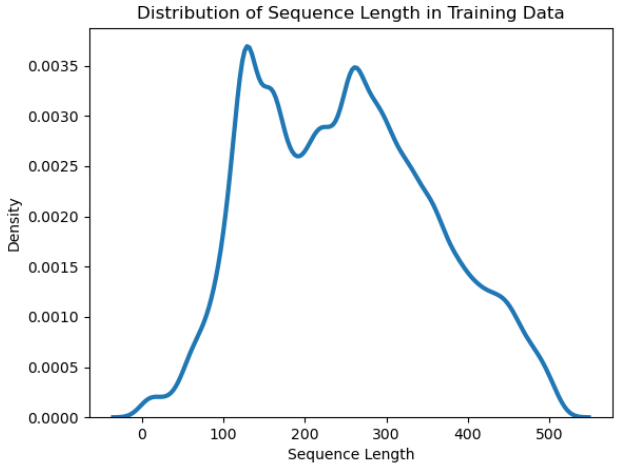

Figure 6. The distribution of sequence length for the VM dataset.

D.1.2. Motif dataset

The motif dataset is extracted from PROSITE, a database maintained by Swiss Institute of Bioinformatics (SIB), which contains 1942 documentation entries, 1311 patterns, and 1400 ProRules (dated January 24, 2024). It contains patterns, profiles, and rules for recognizing specific motifs in protein sequences.

The dataset consists of 16,251 motif fragments, based on their representation in the Protein Data Bank (PDB), which involved aligning protein sequences and atom coordinates with known motifs.

D.2. Evaluation Metrics

Following Trippe et al. (2023), we calculate the **scTM** of one generated structure as follows: (1) we utilize the ProteinMPNN (Dauparas et al., 2022) to design the amino acid sequence. (2) The designed sequence from the ProteinMPNN is input into the ESMFold (Lin et al., 2023) to get the structure. (3) The TM-score between the structures from the ESMFold and generated from our model is calculated as the scTM. The workflow is illustrated in Fig. 7.

We calculate the *in silico* Success Rate following Trippe et al. (2023) as follows: (1) for each group of motifs, we generate 5 samples for each length. (2) each generated sample is input into the ProteinMPNN to design the sequence. (3) each sequence is input into the ESMfold 8 times to get the folding protein structures. (4) We calculate the TM score between

our generation and the 8 folding results from ESMFold and take the highest TM score as scTM from this generation. One generation is successful if the $\mathrm{scTM} > 0.5$

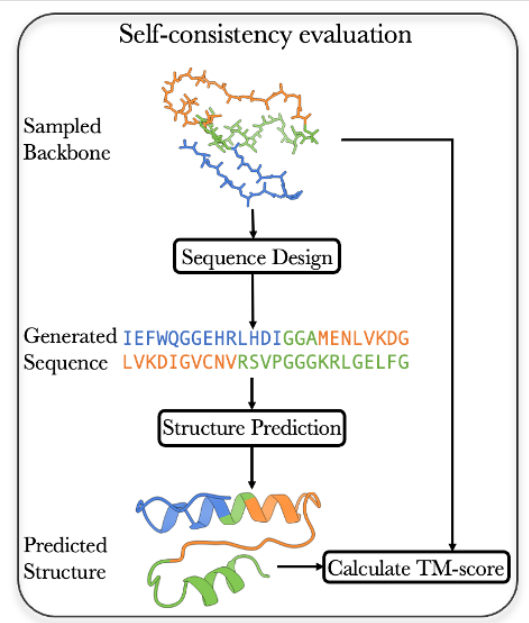

Figure 7. For self-consistency evaluation, we utilize a pre-trained fixed-backbone sequence-design model, namely ProteinMPNN, to design the scaffold sequence from the generated protein structure. Then we put the designed sequence to ESMFold to obtain the full protein structure. Finally, we calculate the TM-score between the predicted structure and the original backbone structure. In the figure, the orange, blue, and green colors indicate the motif `2BGS`, `1G79`, and the scaffold.

We have also calculated the **diversity** and **pdbTM** for each generation. **Diversity** is the ratio of unique clusters in the number of generated samples where the clusters are produced by MaxCluster (Herbert & Sternberg, 2008) following previous works. pdbTM indicates the novelty of generated protein structures. Each generated protein structure is compared with the structures in PDB (Berman et al., 2000) to get the TM-score between two structures as pdbTM score, one generation is novel if the $\mathrm{pdbTM} < 0.7$.

D.3. Experimental Results

D.3.1. Evaluation on VM Dataset

The distribution of TM-scores on the VM dataset is shown in Fig. 8A. Hign TM-score indicates the generated protein structure is similar to the native structures where the desired motifs are located.

D.3.2. Evaluation on MS Dataset

The diversity of all the generations is 1, so we just mention it in the appendix. The pdbTM which indicates the Novelty of generation for each generation, is shown in Table 4. Following previous works Yim et al. (2023); Mao et al. (2024); Watson et al. (2023), one generated structure is novel if the pdbTM< 0.7, here we show the ratio of novel protein structures in the generation comparing with FrameDiff and VFNDiff.

Method		FADiff				VFN-Diff	FrameDiff
# Motif		2	3	4	5		
$\mathrm{pdbTM}_{<0.7}$	overall	84.60%	89.23%	90.33%	93.75%	41.67%	54.67%
	designable	82.94%	85.66%	86.88%	89.91%	1.67%	1.33%

Table 4. pdbTM for each generation. The numbers indicate the ratio of novel protein structures in the generated samples.

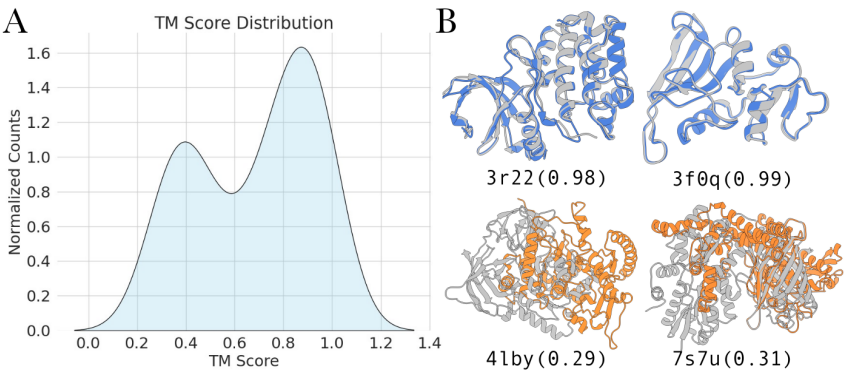

Figure 8. A. TM-score distribution on evaluation dataset of VM dataset. B. The generated structure with a high tm-score closely matches the native structure, while a low tm-score indicates the generated structure differs from the native structure (grey). The four-digit number below each structure indicates the protein identifier and the number in the brackets denotes the TM score.

D.3.3. Ablation Study

The whole results of the ablation study on all the lengths are shown in Table 5.

Method	160	210	260	310	360	410	Avg
FADiff	76.67	71.67	81.67	65.00	86.67	56.67	73.05
Random	49.33	60.00	60.00	54.67	82.67	57.33	60.67
$/2$	69.23	76.92	76.92	60.00	83.08	69.23	72.56
$\times 2$	70.00	70.00	75.00	78.33	83.33	56.67	72.22

Table 5. In silico success rate (SR%) for different lengths, with/without TSP, and with different translation noise scales in sampling. TSP and Random indicate the sequence construction method. $/2$ and $\times 2$ indicate the noise scale on translation in sampling. The Avg indicates the Average success rate for one method.

Noise scale on translation in training The failed cases from the FADiff trained without increasing the noise scale on translation are shown in Fig. 9 and Fig. 10.

Figure 9. Failed case 1

Figure 10. Failed case 2

TSP for sequence construction TSP is performed in the first 100 steps while random is performed at the first step only. Here, we show the results of performing TSP with different settings in Fig. 11 and Table 6. If we conduct TSP throughout the entire sampling process, this will cause the model to be unable to converge because the position of amino acids on the sequence is constantly changing. However, if we conduct TSP in the first 100 steps, this will steer the motifs to the appropriate position because the noise is gradually decreasing during the first 100 steps. Besides, conducting TSP only in the first 100 steps will not cause the model to fail to converge. Performing TSP multiple times results in a better scTM than only performing TSP once in the first 100 steps.

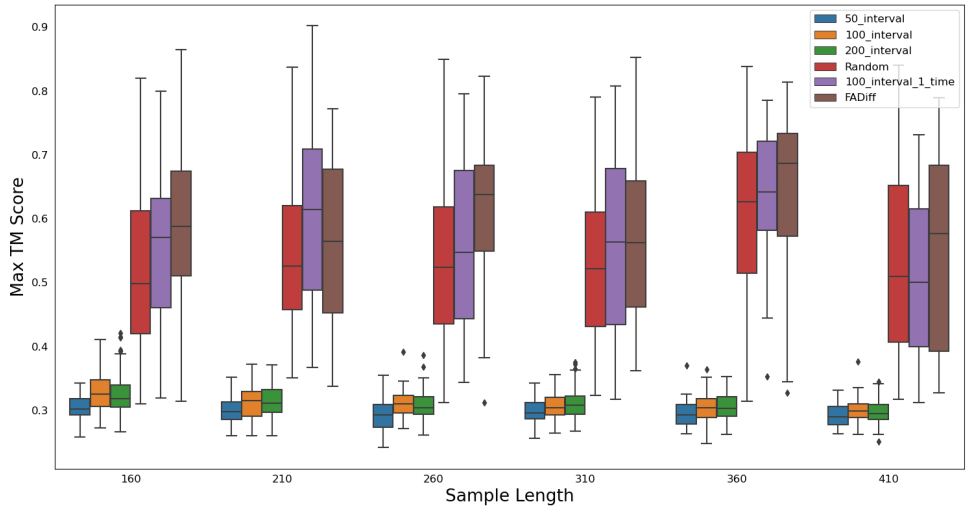

Figure 11. 50_interval, 100_interval, and 200_interval indicate that we perform TSP every 50, 100, and 200 steps in sampling. Random indicates TSP is not performed. 100_interval_1_time and FADiff indicate that we only perform TSP in the first 100 steps two times or multiple times (every step) in the first 100 steps.

Method	160	210	260	310	360	410
Random	49.33	60.00	60.00	54.67	82.67	57.33
100_interval_1_time	67.50	67.50	60.00	62.50	90.00	50.00
FADiff	76.67	71.67	81.67	65.00	86.67	56.67

Table 6. In silico success rate (SR%) for different TSP settings.

D.3.4. Comparison

We also compare FADiff with inpainting and conditional generation methods. FADiff outperforms inpainting consistently on the SCTM score as shown in Fig. 12.

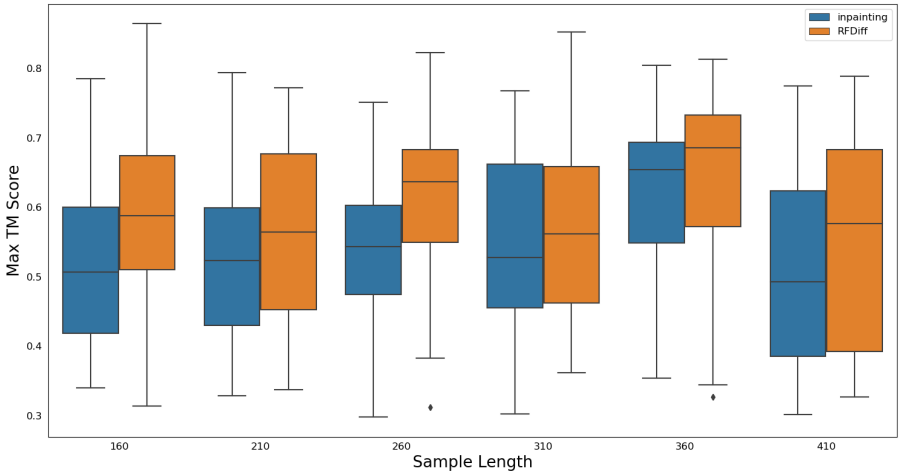

Figure 12. TM-score of inpainting methods and FADiff on generation different lengths of scaffoldings.