Department of Life Sciences

Imperial College of Science, Technology and Medicine

MRes Bioinformatics Project

Diffusion Posterior Sampling via Sequential Monte Carlo for Zero-Shot Scaffolding of Protein Motifs

Author:
James Matthew Uygongco Young

Supervisor:
Dr Omer Deniz Akyildiz

August 2024

Word Count: 9628

Submitted in partial fulfilment of the requirements for the
MRes Bioinformatics and Theoretical Systems Biology of Imperial College London

Abstract

With the advent of protein diffusion models, new proteins can be generated at an unprecedented rate. The *motif scaffolding problem* requires steering this generation process to produce proteins with a desirable functional substructure—a motif. While models have been trained on this conditional task through classifier-free guidance, recent techniques in diffusion posterior sampling can be leveraged as zero-shot alternatives. In addition, their approximations can be corrected with sequential Monte Carlo algorithms to asymptotically target the exact posterior distribution. In this work, we formalise the single-motif, multi-motif, and symmetric motif scaffolding tasks as inverse problems. We then solve them by adapting diffusion posterior samplers with an unconditional model, Genie, acting as a prior. Against established benchmarks, we find some success in scaffolding single motifs and nearly designable scaffolds in the multi-motif case. The latter is possible by comparing motifs with the predicted fully-denoised proteins in an SE(3)-invariant likelihood measure involving pairwise distances between each residue's rigid body frame representation. This setup performs comparably to conventional masking approaches in the single motif case but further generalises to multiple motifs. Moreover, we also produce designable monomers with cyclic and dihedral internal symmetries. This work demonstrates the capabilities and areas for improvement of zero-shot posterior samplers in motif scaffolding tasks.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

Proteins are fundamental to many biological systems. Naturally occurring and defined by an amino acid sequence, they fold to structural conformations that determine their function. Nature, however, has explored but a tiny fraction of the entire protein universe, furthering its reach through evolution at the scale of millions of years. De novo protein design aims to accelerate this process to hours or days. Recent works [1, 2, 3, 4] use diffusion models [5], a class of generative models, to learn the diverse distribution of protein structures, permitting the sampling of new, potentially novel proteins. Consequently, a natural task is to steer the generation process to produce proteins containing a particular functional substructure—a *motif*. While works have demonstrated their ability on this task, the **motif scaffolding problem**, along with its variants, remains a challenge.

The most successful approach to date involves training an unconditional diffusion model to condition upon the existence of the motif [1, 6]. However, additional training for each design task may be expensive amid variable design requirements, prompting the need for a generalisable method. Concurrent efforts [7, 8, 9] propose posterior sampling techniques to solve numerous inverse problems with a diffusion model as the prior. When paired with sequential Monte Carlo algorithms, these methods can guarantee asymptotically exact sampling. By formalising the motif scaffolding problem and its variants into inverse problems, they become compatible with posterior samplers and can be solved in a zero-shot fashion, i.e. without explicitly training for the task. Wu *et al.* [9] and Trippe *et al.* [2] have done this for the classic motif scaffolding problem by conditioning on a partial view of the protein backbone. However, while they have laid the foundation, their methods are not immediately generalisable in the case of multiple motifs. Moreover, diffusion posterior samplers have not been sufficiently compared in the context of motif scaffolding. Hence, a general framework for different motif scaffolding tasks and the subsequent evaluation of compatible posterior samplers are needed.

1.1 Objectives

Motivated by the success of generative models in de-novo protein design, we seek to extend conditional methods to support variants of the motif scaffolding problem, none of which have been performed before without conditional training. Additionally, we aim to establish which posterior sampling techniques are appropriate for motif scaffolding.

1.2 Contributions

In meeting our objectives, we made the following contributions through the study.

- **Formalisation of the motif scaffolding problem and its variants:** We define general inverse problems for motif scaffolding, multi-motif scaffolding, and symmetric motif scaffolding for monomers. These representations make it possible to condition on any of the tasks without additional training to an unconditional protein backbone diffusion model.

- **Frame-based distance conditioning:** We propose alternative motif definitions that provide comparable performance with existing approaches yet are generalisable to other scaffolding tasks. In particular, we represent a motif free of its orientation and location but preserving its chirality. To our knowledge, this is the first instance in which such a representation is used.

- **Adapting diffusion posterior samplers for scaffolding tasks**: Several diffusion posterior sampling techniques have either been applied to the single-motif scaffolding task or none at all. We adapt them to work with formalised scaffolding problems and evaluate their performance across several established benchmarks.

- **Ablation and hyperparameter studies**: We perform experiments that explore how generated backbones are affected by different parameters of the sampler's proposal and likelihood, together with hyperparameters of the unconditional model.

- **Public code repository:** We publish our source code on GitHub with a unified interface to different scaffolding tasks under several methods. The supported experiments are designed to be easily configurable and modular enough to use other unconditional models, posterior samplers, and protein motifs. It is available at github.com/matsagad/mres-project.

1.3 Ethical Considerations

The problems addressed in the study revolve around protein motifs, the structures of which were retrieved from the Protein Data Bank. When scaffolds were generated, designability metrics were quantified purely in silico. No real proteins were handled or synthesised as part of the project.

During the study, we used several GPUs for model inference and evaluations. While these devices generally exhibit a non-negligible carbon footprint, we remark that by opting for specialised hardware accelerators such as GPUs over general-purpose CPUs, we minimise our overall computation times and maximise the number of FLOPs per carbon emitted.

Chapter 2

Preliminaries

This chapter covers preliminary material to make our methods accessible to a broad audience. We begin by briefly summarising sequential Monte Carlo algorithms for posterior sampling in state-space models. We then outline the machinery of diffusion models, which are the probabilistic models considered. Lastly, we focus our attention on protein backbones—our distribution of interest.

2.1 Sequential Monte Carlo

Often, we have sequential measurements $\mathbf{y}_{1:t}$ and attribute their variations to latent variables $\mathbf{x}_{1:t}$. To understand the system's underlying mechanism, we may wish to model the posterior distribution $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$ and, as further measurements are taken, update our model accordingly. This is known as the *online filtering problem*. While broader in scope now, **sequential Monte Carlo** (**SMC**) algorithms were originally intended to solve this problem. In this overview, we precisely focus on this application and follow the introductory texts by Naesseth, Lindsten, & Schön [10] and Doucet & Johansen [11].

To begin with, we have a so-called *target* distribution π_t whose density is given by

$$\pi_t(\mathbf{x}_{1:t}) = \frac{\gamma_t(\mathbf{x}_{1:t})}{Z_t},$$

for some positive function γ_t and its normalising constant Z_t. Typically, we are only concerned with π_t at the final time step $t = T$, and those at $t = 1, \ldots, T-1$ are merely intermediaries. The first step to SMC is to choose a suitable target that matches our desired distribution. For example, in the filtering problem, we can choose $\gamma_t(\mathbf{x}_{1:t}) = p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ so that $Z_t = p(\mathbf{y}_{1:t})$ and $\pi_t(\mathbf{x}_{1:t}) = p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$. Of course, if we have access to the posterior, we can immediately assign it to γ_t. How these sequential distributions behave largely depends on the probabilistic model at hand.

2.1.1 State-Space Models

One example of a probabilistic model is a **state-space model (SSM)**. Here, we have access to a *prior* μ on the initial latent state $\mathbf{x}_1$, a Markovian *transition kernel* f, and a *likelihood* g,

$$\begin{aligned} \mathbf{x}_1 &\sim \mu(\cdot), \\ \mathbf{x}_t \mid \mathbf{x}_{t-1} &\sim f(\cdot \mid \mathbf{x}_{t-1}) \quad \text{for } t \geq 2, \\ \mathbf{y}_t \mid \mathbf{x}_t &\sim g(\cdot \mid \mathbf{x}_t) \quad \text{for } t \geq 1. \end{aligned}$$

We have then the unnormalised joint distribution of $\mathbf{x}_{1:t}$ as

$$\gamma_t(\mathbf{x}_{1:t}) \coloneqq p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \mu(\mathbf{x}_1) g(\mathbf{y}_1 \mid \mathbf{x}_1) \prod_{i=2}^{t} f(\mathbf{x}_i \mid \mathbf{x}_{i-1}) g(\mathbf{y}_i \mid \mathbf{x}_i).$$

However, as in most cases, how we may sample from γ_t is not straightforward. SMC alleviates this issue with a Monte Carlo approximation by way of *particle filtering*.

2.1.2 Importance Sampling

The key idea to particle filtering is a technique called **importance sampling (IS)**. Given the difficulty of sampling from the target distribution, we can sample from an easier *proposal* distribution and assign weights to the samples. Formally, we can compute the expectation of some function h_t under our target distribution π_t

$$\pi_t(h_t) = \mathbb{E}_{\pi_t}[h_t(\mathbf{x}_{1:t})],$$

by introducing a proposal q_t and rearranging the expression in terms of expectations under the proposal

$$\begin{aligned} \mathbb{E}_{\pi_t}[h_t(\mathbf{x}_{1:t})] &= \int_{\mathcal{X}} h_t(\mathbf{x}_{t:t}) \pi_t(\mathbf{x}_{1:t}) d\mathbf{x}_{1:t} = \frac{1}{Z_t} \int_{\mathcal{X}} \frac{\gamma_t(\mathbf{x}_{1:t})}{q_t(\mathbf{x}_{1:t})} h_t(\mathbf{x}_{t:t}) q_t(\mathbf{x}_{t:t}) d\mathbf{x}_{1:t} \\ &= \frac{1}{Z_t} \mathbb{E}_{q_t}\left[\frac{\gamma_t(\mathbf{x}_{1:t})}{q_t(\mathbf{x}_{1:t})} h_t(\mathbf{x}_{1:t})\right] = \frac{\mathbb{E}_{q_t}\left[\frac{\gamma_t(\mathbf{x}_{1:t})}{q_t(\mathbf{x}_{1:t})} h_t(\mathbf{x}_{1:t})\right]}{\mathbb{E}_{q_t}\left[\frac{\gamma_t(\mathbf{x}_{1:t})}{q_t(\mathbf{x}_{1:t})}\right]}. \end{aligned}$$

Then, we can independently sample $\mathbf{x}^i_{1:t} \sim q_t(\cdot)$ and estimate the original expectation as

$$\mathbb{E}_{\pi_t}[h_t(\mathbf{x}_{1:t})] \approx \frac{\sum_{i=1}^{K} \tilde{w}_t(\mathbf{x}^i_{1:t}) h_t(\mathbf{x}^i_{1:t})}{\frac{1}{K} \sum_{i=1}^{K} \tilde{w}_t(\mathbf{x}^i_{t:t})},$$

where $\tilde{w}_t(\mathbf{x}^i_{1:t}) = \gamma_t(\mathbf{x}^i_{1:t})/q_t(\mathbf{x}^i_{1:t})$ act as weights, for $i = 1, \dots, K$. We denote these weights as $\tilde{w}^i_t$ and their normalised counterparts as w^i_t for short. A nice property of the unnormalised weights is they form a consistent estimator $\hat{Z}_t$ for the normalising constant Z_t,

$$\hat{Z}_t = \frac{1}{K}\sum_{i=1}^{K} \tilde{w}^i_t.$$

This is useful, particularly when we nest SMC applications. When $h_t = \delta_{\mathbf{x}_{1:t}}$, the Dirac measure, we effectively approximate the target distribution π_t.

For every new measurement, however, we need to sample and evaluate the expressions again for all time steps. To do this efficiently, we can choose an autoregressive proposal

$$q_t(\mathbf{x}_{1:t}) = q_t(\mathbf{x}_{1:t-1})q_t(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}).$$

This forms the basis of **sequential importance sampling** (**SIS**). We omit the details here, but a recursive update for the unnormalised weights $\tilde{w}^i_t$ can be derived

$$\tilde{w}_t(\mathbf{x}_{1:t}) = \frac{\gamma_t(\mathbf{x}_{1:t})}{\gamma_{t-1}(\mathbf{x}_{1:t-1})q_t(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})}\tilde{w}_{t-1}(\mathbf{x}_{1:t-1}), \tag{2.1}$$

where $q_1(\mathbf{x}_1 \mid \mathbf{x}_{1:0}) = q_1(\mathbf{x}_1)$ and $\tilde{w}_0, \gamma_0 = 1$. As such, the weights need not be computed from scratch but are simply updated by a multiplicative factor.

However, with repeated multiplications, one weight is bound to dominate while the rest go to zero. This phenomenon, called *weight degeneracy*, can be mitigated by resampling each $\mathbf{x}^i_t$ according to their weights w^i_t at each step, practically filtering out samples with extremely low weights. This resampling step transforms SIS into a **particle filter**.

2.1.3 Particle Filtering

The general particle filter, outlined in Algorithm 2.1, combines SIS and resampling. A small difference in our derivations is that the weights are set to the multiplicative factor found in Equation 2.1 as resampling already accounts for the weights in the previous step. Also, we define $q_1(\mathbf{x}_1 \mid \mathbf{x}_0) = q_1(\mathbf{x}_1)$.

Again, particle filtering aims to approximate the target distribution at the final time step π_T. Generally, we can make two main choices that determine our sampler's efficiency: the proposal q_t and the intermediate unnormalised target functions γ_t. For example, we look at the simplest filter within SSMs, the **bootstrap particle filter** (**BPF**).

Algorithm 2.1: The General Particle Filter

```
input  : Measurements $\mathbf{y}_{1:T}$, proposals $q_t$, unnormalised target functions $\gamma_t$, no. of
         particles $K$
output : Approximate samples from the target $\mathbf{x}_T^{1:K}$ and their weights $w_T^{1:K}$
for $t = 1, \ldots, T$ do
    for $i = 1, \ldots, K$ do
        Sample $\bar{\mathbf{x}}_t^i \sim q_t(\cdot \mid x_{t-1}^i)$                          # Sample from proposal
        Set $\tilde{w}_t^i \leftarrow \gamma_t(\mathbf{x}_{1:t}) / \gamma_{t-1}(\mathbf{x}_{1:t-1}) q_t(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})$     # Update weights
    end
    Set $w_t^i \leftarrow \tilde{w}_t^i / \sum_{j=1}^K \tilde{w}_t^j$, for $i = 1, \ldots, K$
    Resample $\mathbf{x}_t^{1:K} \sim$ Multinomial($w_t^{1:K}$, $\bar{\mathbf{x}}_t^{1:K}$)          # Resample particles
end
```

Bootstrap Particle Filter

Recall that, in an SSM, we can choose γ_t as the joint distribution to match the posterior. Under this choice, the update rule is given by

$$\tilde{w}_t(\mathbf{x}_{1:t}) = \frac{f(\mathbf{x}_t \mid \mathbf{x}_{t-1}) g(\mathbf{y}_t \mid \mathbf{x}_t)}{q_t(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})} \tilde{w}_{t-1}(\mathbf{x}_{1:t-1}),$$

and, if we choose the proposal $q_t := f$, we find an even simpler result

$$\tilde{w}_t(\mathbf{x}_{1:t}) = g(\mathbf{y}_t \mid \mathbf{x}_t) \tilde{w}_{t-1}(\mathbf{x}_{1:t-1}),$$

depending only on the likelihood. The BPF algorithm is, therefore, a simpler instance of the general particle filter and is often the baseline in SSMs.

Path Degeneracy

As particle filtering calls for repeated resampling, a common occurrence is for all particles to be identical after a number of iterations. This is known as *path degeneracy*. Certain strategies can be employed to encourage diversity amongst the particles in the resampling procedure.

Low-Variance Resampling Normally, resampling takes the form of multinomial sampling. However, in practice, techniques such as residual, systematic, and stratified resampling are often used, as they reduce variance at the cost of some added correlation. They effectively allow particles with substantial weights to make it through resampling without the risk of them being excluded.

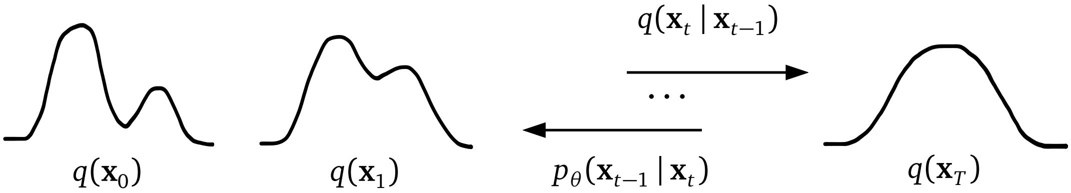
$q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$
$\cdots$
$q(\mathbf{x}_0)$
$q(\mathbf{x}_1)$
$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$
$q(\mathbf{x}_T)$

Figure 2.1: The diffusion process on probability densities. The original distribution $q(\mathbf{x}_0)$ is approximately Gaussian after a large number of noising steps T. Diffusion models learn a reverse process $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ to undo the added noise.

Adaptive Resampling Another strategy is to reserve the resampling step only for when the weights start becoming degenerate. One measure used to determine degeneracy is the *Effective Sample Size* (*ESS*) defined by

$$\mathrm{ESS}_t = \frac{1}{\sum_{i=0}^{K} \left(w_t^i\right)^2},$$

which is equal to one when all weights except for one are zero and K when the weights are all equal. Typically, resampling is only triggered when $\mathrm{ESS}_t \leq K/2$. In iterations where no resampling occurs, the weights are updated multiplicatively to account for the weights in the previous step. For example, in BPF, we set $\tilde{w}_t^i = g(\mathbf{y}_t \mid \bar{x}_t^i) w_{t-1}^i$ and, after resamping, set $w_t^i = 1/K$.

2.2 Diffusion Models

Generative modelling aims to learn some data distribution $q(\mathbf{x})$. Here, we focus on a particular class called **diffusion models**. Introduced by Sohl-Dickstein *et al.* [5], diffusion models generate samples by learning to undo a diffusion process. It relies on the insight that, for $\mathbf{x}_0 \sim q(\cdot)$, the sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T$, constructed by progressively adding isotropic Gaussian noise over a large number of steps T, will result in $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{0},\ \mathbf{I})$. This is illustrated in Figure 2.1. By learning to reverse the noise process, samples from Gaussian noise are precisely mapped to those in $q(\mathbf{x})$. Below, we discuss two equivalent diffusion model frameworks under discrete denoising steps.

2.2.1 Denoising Diffusion Probabilistic Models

A reformulation by Ho *et al.* [12], **denoising diffusion probabilistic models** (**DDPMs**) learn to measure the total noise accumulated by the data distribution from the *forward* (*noising*) *process*—a fixed Markov process with the transition kernel

$$q(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t;\ \sqrt{1-\beta_t}\mathbf{x}_{t-1},\ \beta_t\mathbf{I}\right), \tag{2.2}$$

for a static variance schedule $\beta_1, \dots, \beta_T \in (0,1)$ that decreases as $t \to 0$. In fact, denoting $\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, the reparameterisation trick allows the direct sampling of $\mathbf{x}_t$ given $\mathbf{x}_0$

$$q(\mathbf{x}_t \,|\, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\right). \tag{2.3}$$

Notice then that we exactly have $q(\mathbf{x}_T \,|\, \mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T;\ \mathbf{0},\ \mathbf{I})$ for large enough T since $\bar{\alpha}_t \to 0$. From above, we can also derive an expression for $\mathbf{x}_0$

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_t\right), \tag{2.4}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$. Hence, while we progressively noise the samples, we can predict the total noise ϵ_t in one step to retrieve an estimate for $\mathbf{x}_0$. The *reverse* (*denoising*) *process* is analogously chosen to be a Markov process with the transition kernel

$$p_\theta(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1};\ \mu_\theta(\mathbf{x}_t, t),\ \Sigma_\theta(t)), \tag{2.5}$$

parameterised by

$$\begin{aligned}\mu_\theta(\mathbf{x}_t, t) &:= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t, t) \qquad\qquad \Sigma_\theta(t) := \beta_t\mathbf{I},\\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right),\end{aligned}$$

where $\epsilon_\theta(\mathbf{x}_t, t)$ is the added noise as predicted by a neural network and $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ is the predicted fully-denoised sample by substituting ϵ_θ in Equation 2.4.

The procedure to generate samples $\mathbf{x}_0 \sim q(\cdot)$ then involves the reverse process

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t),$$

first sampling from $\mathbf{x}_T \sim \mathcal{N}(\cdot;\ \mathbf{0},\ \mathbf{I})$ then gradually denoising the samples for T steps.

2.2.2 Score-Based Generative Models

An alternative view proposed by Song & Ermon [13] is to model the data distribution's *score* function $s(\mathbf{x}) = \nabla_{\mathbf{x}} \log q(\mathbf{x})$, a vector that points in the direction of areas with high density, which is effectively where noisy samples need to move towards to undo the forward process. We quickly discuss these **score-based generative models** (**SBGMs**) in relation to DDPMs.

First, a *score network* s_θ is trained under a score matching objective to approximate $s_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log q(\mathbf{x})$. A Markov chain Monte Carlo procedure called *Langevin dynamics* is then used to sample from $q(\mathbf{x})$. The process starts by sampling from a prior distribution $\mathbf{x}_T \sim \pi(\cdot)$ then proceeds to iteratively compute

$$\mathbf{x}_{t-1} \leftarrow \mathbf{x}_t + \frac{\epsilon}{2} s_\theta(\mathbf{x}_t) + \sqrt{\epsilon}\mathbf{z}_t, \quad \text{for } t = T, \ldots, 1,$$

where $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0},\, \mathbf{I})$ and $\epsilon > 0$ is a fixed step size. When $\epsilon \to 0$ and $T \to \infty, p(\mathbf{x}_0) = q(\mathbf{x})$.

Clearly, parallels can be drawn between DDPMs and the score-based approach in their sampling procedures. We conclude by making their relationship more explicit. Taking the gradient of the DDPM formulation from Equation 2.3, we find

$$\begin{aligned} s(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0) &= \nabla_{\mathbf{x}_t} \left(-\frac{1}{2(1-\bar{\alpha}_t)} \left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)^2 \right) \\ &= -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{1-\bar{\alpha}_t} = -\frac{\epsilon_t}{1-\bar{\alpha}_t}, \end{aligned}$$

which establishes the connection between the two frameworks' learning objectives as

$$s_\theta(\mathbf{x}_t, t) = -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}. \tag{2.6}$$

When working with de-noising networks, we use this relationship to compute the score when required.

2.3 Protein Structure

Protein molecules are made up of linear chains of amino acid residues. Those consisting of a single chain are called *monomers*, and those with multiple chains, i.e. made up of several monomers, are called *oligomers*. Remarkably, they fold to structural conformations that are non-trivially defined by their amino-acid sequence. This structure is commonly described in varying levels of complexity, but, like most generative modelling efforts, we focus on a protein's *tertiary* structure—its three-dimensional arrangement in space. A key reason is that while the sequence defines the protein, its structure more clearly determines its function. Additionally, we limit our scope to monomeric proteins, as most generative efforts have.

2.3.1 Geometric Priors

Protein geometries empirically follow some rules. Hence, we can efficiently capture their features by combining geometric constraints into their representation. To begin with, we briefly introduce the notion of geometric priors and expand on those relevant to proteins.

Loosely, the group of symmetries $\mathcal{G}$ of an object $\mathbf{x} \in \mathcal{X}$ are transformations on it that produce the same object, i.e. $T(\mathbf{x}) = \mathbf{x}$ for $T \in \mathcal{G}$. In the three-dimensional domain on which proteins lie, Euclidean transformations take the form of rotations, translations, and reflections. These transformations form what is known as the *three-dimensional Euclidean group* $E(3)$. However, while proteins are treated the same under rotations and translations, they are generally different when reflected. *Chirality* in proteins can be observed, for example, in its strictly right-handed α-helices. The **three-dimensional special Euclidean group** $SE(3)$, containing rotations and translations but not reflections, is, therefore, the symmetry group associated with proteins.

Often, we want to learn some function $f : \mathcal{X} \to \mathcal{Y}$, whether it be a classifier, a regressor, or, as we will look at later, a de-noising network. However, the learning problem may demand learning symmetries within $\mathcal{X}$ from scratch. Instead, we may bake these symmetries within f itself to make learning more tractable. Typically, we want f to possess *invariance* or *equivariance* under a symmetry group $\mathcal{G}$. Take any $\mathbf{x} \in \mathcal{X}$ and $T \in \mathcal{G}$. We say f is **invariant** under $\mathcal{G}$ if we have $f(\mathbf{x}) = f(T(\mathbf{x}))$. Whereas, for $\mathcal{Y} = \mathcal{X}$, we say f is **equivariant** under $\mathcal{G}$ if we have $f(T(\mathbf{x})) = T(f(\mathbf{x}))$. For example, in classifying whether proteins are fluorescent, f must be $SE(3)$-invariant, as a protein's location or orientation has no bearing on this trait. On the other hand, a de-noising network on proteins is ideally $SE(3)$-equivariant, so proteins are de-noised the same way regardless of their position.

2.3.2 Backbone Representations

Proteins can be partitioned into their backbone and side chains, which are often modelled separately. Here, we cover different attempts to represent protein backbones with $SE(3)$-invariance, as illustrated in Figure 2.2. Protein backbones are linear and have an alternating $\mathrm{N} - \mathrm{C}\alpha - \mathrm{C}$ atomic structure. One representation fixes the bond lengths between atoms and focuses on their **torsion angles** [14], removing the need for reasoning with coordinates. However, while $SE(3)$-invariant, errors in predicted angles accumulate throughout the chain, leading to inaccurate global conformations. A more globally aware alternative is modelling the **pairwise distances between C-α atoms** [15], capturing enough of the structure to recover the rest of the atomic positions accurately. It

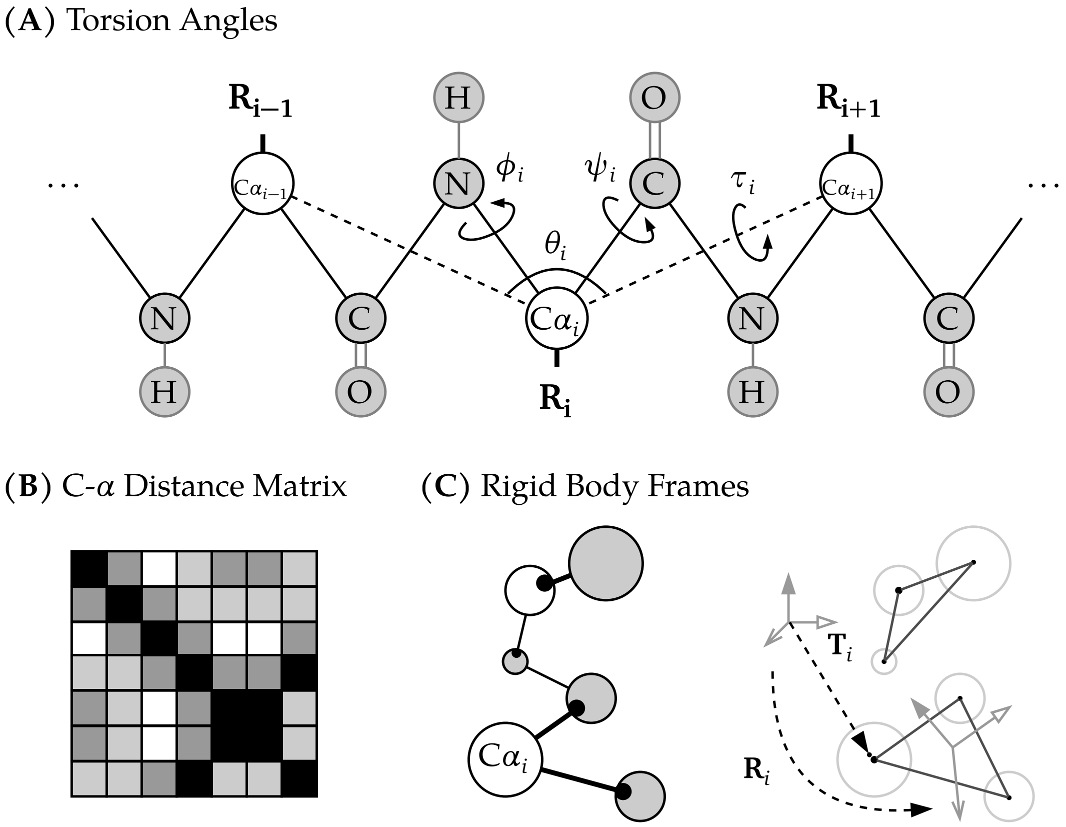

Figure 2.2: ***SE*(3)-invariant protein backbone representations**. (**A**) Torsion angles are sufficient to define the entire protein backbone. Here, $\mathbf{R_i}$ is the side-chain of the ith residue, $C\alpha_i$ is the ith C-α atom, and $\phi_i, \theta_i, \psi_i, \tau_i$ are the torsion angles defining the orientations of other atoms with respect to the ith C-α atom. (**B**) Pairwise C-α distances can also be modelled, and the rest of the backbone is predicted afterwards. (**C**) Fixing $\mathrm{N} - \mathrm{C}\alpha - \mathrm{C}$ substructures as rigid bodies, residues can be represented as (triangular) frames, defined by a rotation matrix $\mathbf{R}_i$ and a translation vector $\mathbf{T}_i$ with respect to a global reference frame.

is $SE(3)$-invariant but additionally reflection-invariant, making it incapable of accounting for chirality. Furthermore, generative modelling poses the need to convert the distance matrix back to three-dimensional space, which is a mapping that is not always possible. Last in this non-exhaustive review is the **frame** representation [16, 3], where the $\mathrm{N}{-}\mathrm{C}\alpha{-}\mathrm{C}$ substructures in each residue are assumed to be rigid bodies with idealised bond lengths and angles. Here, each residue frame is represented by a rotation matrix $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{T}_i \in \mathbb{R}^3$, for its orientation and position with respect to a global reference frame. These representations can be derived from the three-dimensional C-α coordinates using the Gram-Schmidt process as in Appendix A.1. This $SE(3)$-invariant, but not reflection-invariant, formulation is used in several current state-of-the-art protein backbone models [1, 6] and the Invariant Point Attention module [16] found in recent deep learning pipelines.

Chapter 3

Background

This chapter provides an overview of research topics we build upon and related works that attempt to solve similar problems. Specifically, we cover diffusion posterior sampling and works within de novo protein design, particularly for motif scaffolding.

3.1 Posterior Sampling in Diffusion Models

With the ability to sample from a complex data distribution $q(\mathbf{x})$, a natural query is to sample conditioned on some label $\mathbf{y}$, i.e. from the posterior $q(\mathbf{x} \mid \mathbf{y})$. For example, we may want to generate proteins $\mathbf{x}$ with some internal symmetry defined by $\mathbf{y}$. Formally, we wish to solve the inverse problem $\mathbf{y} = \mathscr{A}(\mathbf{x}) + \mathbf{n}$, for $\mathbf{n} \sim \mathscr{N}(\cdot;\ \mathbf{0},\ \sigma^2\mathbf{I})$, with a diffusion model acting as a prior distribution on $\mathbf{x}$. Below, we review several recent works.

3.1.1 Classifier-Free Guidance

Recall that SBGMs and, equivalently, DDPMs are capable of reversing a diffusion process to generate samples from $q(\mathbf{x})$ by modelling the score $s(\mathbf{x}) = \nabla_{\mathbf{x}} \log q(\mathbf{x})$—a quantity that, broadly speaking, determines the direction noisy samples need to diffuse towards to approximate $q(\mathbf{x})$. To sample from the posterior, we may opt to move in the direction of the conditional score at de-noising step t given by

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\mathbf{y} \mid \mathbf{x}_t), \tag{3.1}$$

which is the sum of the score and the gradient of the log-likelihood, which we will call the *guidance term*. In fact, we may even scale the guidance term to magnify the conditional signal. In theory, then, a classifier trained to label samples from the data distribution $q_\theta(\mathbf{y} \mid \mathbf{x})$ can be used to perform conditional sampling. However, classifiers are usually trained on non-noisy data, and the conditional score needs to be evaluated for noised samples. Ho & Salimans [17] sidestep this by conducting joint conditional and unconditional training of a diffusion model in what is known as **classifier-free guidance**. Here, the model learns to take input labels and de-noise samples conditioned on them. To then magnify the conditional signal, they expand the likelihood in Equation 3.1 using Bayes rule and place a **guidance scale** γ to arrive at

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log q_\gamma(\mathbf{x}_t \mid \mathbf{y}) &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \gamma \left(\nabla_{\mathbf{x}_t} q(\mathbf{x}_t \mid \mathbf{y}) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \right) \\ &= (1-\gamma) \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} q(\mathbf{x}_t \mid \mathbf{y}).\end{aligned}$$

When $\gamma = 0$, the unconditional model is retrieved. For large γ, the modes of the distribution are magnified while its troughs shrink, resulting in samples better satisfying the label $\mathbf{y}$ at the cost of diversity [17]. However, it may be desirable to omit the need for conditional training to leverage unconditional models in various conditional settings. This is the case with the subsequent methods.

3.1.2 Projections as Guidance

We loosely refer to the following two methods as *projection methods* despite not being conventional terminology. One idea is to project the intermediate latent variables $\mathbf{x}_t$ onto an observation subspace at each denoising step to maintain sample consistency with the label $\mathbf{y}$. Yang *et al.* [18] make the approximation

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log \int q(\mathbf{x}_t \mid \mathbf{y}_t, \mathbf{y}) \psi(\mathbf{y}_t \mid \mathbf{y}) \mathrm{d}\mathbf{y}_t \approx \nabla_{\mathbf{x}_t} \log \int q(\mathbf{x}_t \mid \mathbf{y}_t) \psi(\mathbf{y}_t \mid \mathbf{y}) \mathrm{d}\mathbf{y}_t \\ \approx \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \hat{\mathbf{y}}_t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\hat{\mathbf{y}}_t \mid \mathbf{x}_t)\end{aligned}$$

where we assume $q(\mathbf{x}_t \mid \mathbf{y}_t, \mathbf{y}) \approx q(\mathbf{x}_t \mid \mathbf{y}_t)$ and $\hat{\mathbf{y}}_t$ is a sample from $\psi(\mathbf{y}_t \mid \mathbf{y})$, the distribution that models how observations are affected by the addition of noise. Provided ψ is tractable, which is in linear inverse problems, then conditional sampling can be performed. We refer to this as **observation-projection**. With linear inverse problems, the sequence $\{\hat{\mathbf{y}}_t\}_{t=1}^T$ is essentially constructed by noising the label $\mathbf{y}$. In fact, if the label is a masked view of the latent variable, we can maximise the guidance term by effectively replacing the masked segment of $\mathbf{x}_t$ with $\hat{\mathbf{y}}_t$ as done in some works [2, 19], also known as the *replacement method*.

However, when the label itself is noisy or the inverse problem at hand is non-linear, the above approach is unsuitable. Chung *et al.* [7] instead project the noisy latent variables $\mathbf{x}_t$ to their predicted de-noised versions $\hat{\mathbf{x}}_0$ which is available through Equation 2.4 applied on the de-noising network's output. The likelihood $q(\mathbf{y} \mid \mathbf{x}_t)$ can then be approximated to yield the conditional score

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) \approx \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\mathbf{y} \mid \hat{\mathbf{x}}_0).$$

Note that the likelihood is precisely $q(\mathbf{y} \mid \hat{\mathbf{x}}_0) = \mathcal{N}(\mathbf{y};\ \mathcal{A}(\hat{\mathbf{x}}_0),\ \sigma^2\mathbf{I})$, whose logarithm's gradient can be computed via backpropagation. We refer to this as **latent-projection**.

Still, errors remain in both projection methods due to their approximations [20, 21]. The next set of methods compounds these projection ideas with particle filtering to provide an asymptotically exact sampling from the posterior.

3.1.3 Sequential Monte Carlo for Diffusion Posterior Sampling

As SMC permits exact posterior sampling with added liberties in the proposal and intermediate target choices, it is well-suited to correct for errors in the approximations of the above methods. Trippe *et al.* [2] performed filtering with the replacement method as part of their SMCDiff algorithm for conditioning on masked regions of latent variables. More generally, Dou & Song [8] extended the observation-projection method by computing the optimal proposal within linear inverse problems together with filtering. Their proposed method, **Filter Posterior Sampling with Sequential Monte Carlo** (**FPS-SMC**), showed competitive performance in several image in-painting and de-blurring tasks among other posterior samplers. On the other hand, Wu *et al.* [9] combined an SMC technique called *twisting* with the approximate optimal proposal from the latent-projection method to solve general inverse problems. Their method called **Twisted Diffusion Sampler** (**TDS**) provides state-of-the-art performance in motif-scaffolding, an in-painting problem within proteins, among methods requiring no conditional training.

A unique feature of these filtering methods is their increasingly accurate posterior sampling at the controlled expense of a larger number of particles.

3.2 De Novo Protein Design

A recent wave of studies has employed generative modelling in the space of protein structures. The success of diffusion models in image generation presents them as suitable candidates for likewise generating new and novel proteins. Unlike images, however, the Euclidean symmetries of proteins need to be accounted for to make the modelling process tractable. We highlight existing protein diffusion models and discuss motif scaffolding as a conditional task.

3.2.1 Protein Backbone Diffusion Models

Given the difficulty of modelling protein side chains, i.e. the sequence must be known beforehand to determine side chains to be modelled [22], works in de novo protein design have historically represented proteins through their backbone. Only after the backbone is fixed are the side chains predicted. Until recently, all-atom models have not been in the picture, and therefore, we focus on recent backbone models.

Using an $E(3)$-equivariant graph neural network (EGNN) setup, Trippe *et al.* [2] trained a diffusion model called ProtDiff for generating diverse protein backbones, represented by the three-dimensional coordinates of C-α atoms. In addition, ProtDiff is used in tandem with the conditional algorithm, SMCDiff, to design scaffolds for protein motifs. Expanding to a broader set of design challenges, Watson *et al.* [1] fine-tune a protein structure prediction network to act as a de-noising network. Their model, RFDiffusion, was conditionally trained on a variety of tasks and produced successful designs verified experimentally. However, in using EGNNs, ProtDiff lacks reflection-invariance and occasionally produces left-hand helices, and RFDiffusion, while $SE(3)$-equivariant, was not trained end-to-end for generative tasks. Yim *et al.* [3] developed a principled framework for training an $SE(3)$-diffusion model and applied it to learning protein structures. Their model, FrameDiff, produces successful designs less frequently than RFDiffusion but is a quarter of its size. In the forward process, FrameDiff noises the translation vector and rotation matrix of each frame separately. Lin & AlQuraishi [4] propose simply noising the three-dimensional coordinates of C-α atoms and training an $SE(3)$-equivariant de-noising network, **Genie**, to asymmetrically de-noise backbones using both elements of the frame representation. Under the simpler scheme, Genie performs superior to FrameDiff on both designability and diversity at a further quarter reduction in size. Building on top of this work, Lin *et al.* [6] introduced Genie2—a conditionally trained Genie model for motif scaffolding problems. With the additional augmentations to training data, including training on a database of AlphaFold [16] predicted structures, Genie2 achieves comparable performance with RFDiffusion in unconditional generation but with state-of-the-art motif scaffolding results.

The culmination of these backbone model developments is their use in generating novel proteins that fit a list of design elements.

3.2.2 Motif Scaffolding Tasks

Often, we want to generate proteins with some functional or conformational properties. The most common task is the **motif scaffolding** problem, where new proteins are de-

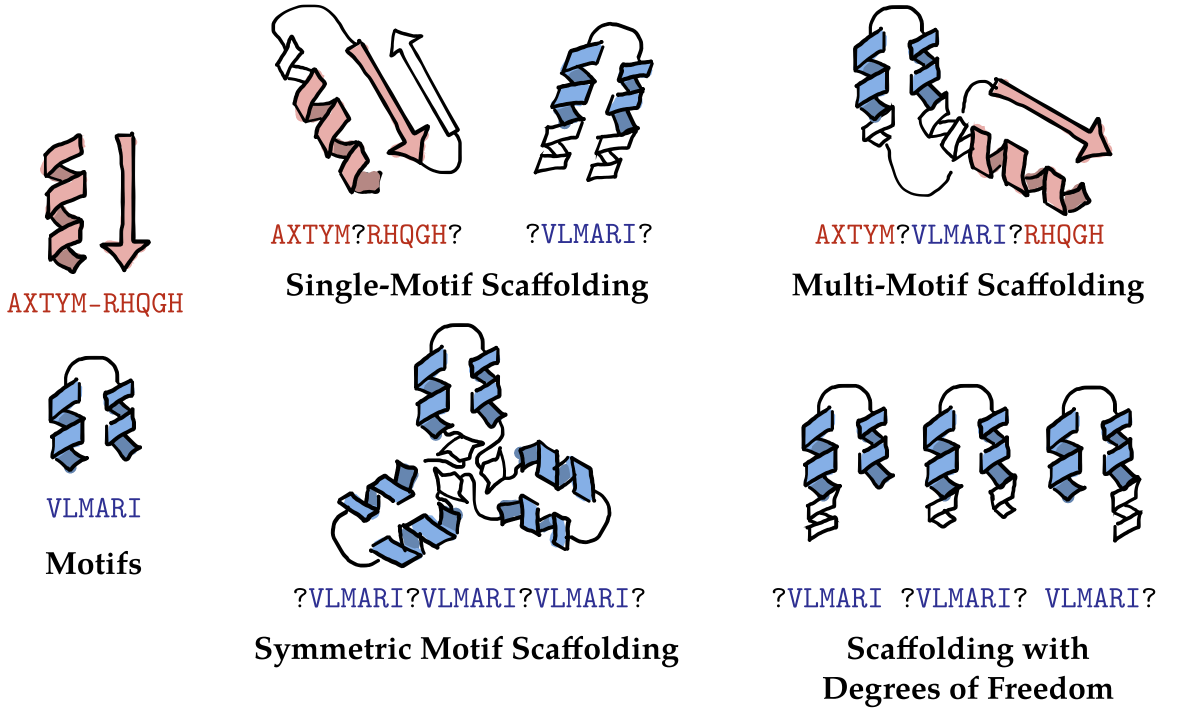

Figure 3.1: Different motif scaffolding tasks. Contiguous and discontiguous motifs are shown in blue and red, respectively. Single-motif scaffolding is concerned with one motif. Multi-motif scaffolding considers two or more motifs, each having an orientation irrespective of the other. Symmetric motif scaffolding produces symmetric proteins, with each subunit containing the motif. Finally, in scaffolding with degrees of freedom, the motif placement is not fixed and can be in several locations on the protein backbone.

signed to include a motif—a segment of an existing protein possessing some functional significance—in some fixed region of the backbone. It is akin to in-painting within images. Several of the previously mentioned backbone models have been adapted in some form to support this task. They are tested on the suite of benchmark problems curated by Watson *et al.* [1] that covers a wide range of motifs, from small molecule binding sites to viral epitopes. Recently, together with Genie2, Lin *et al.* [6] further proposed a set of benchmark problems for **multi-motif scaffolding**, where several motifs may be conditioned upon. This differs from single-motif scaffolding with a discontiguous motif as each motif's orientation is independent of the other. Another task demonstrated by RFDiffusion is the generation of oligomers symmetric under some point symmetry. They further compound this with a motif to have **symmetric motif scaffolding**, where each monomer contains a copy of the motif. A different generalisation to the original problem, in-painting with degrees of freedom was introduced as a conditional task supported by the diffusion posterior sampler TDS [9]. Instead of fixing the motif, Wu *et al.* consider allowing the motif to be located anywhere on the backbone. We will refer to this as **scaffolding with degrees of freedom**. All the tasks are depicted in Figure 3.1

3.3 Related Work

Our focus lies in solving various motif scaffolding problems through diffusion posterior sampling with SMC. Several works [1, 6, 23] attempt to solve the same problems by conditionally training their models on specific tasks. We differentiate our work in being generalisable to many tasks by simply modifying our formulations, thereby requiring no additional training. A similar work, Chroma [24], provides an extensive suite of composable conditioners built on top of an unconditional model but does not use filtering and differs in their conditional formulations.

In terms of posterior sampling, existing methods such as BPF and FPS-SMC [8] have not been explored in the motif scaffolding setting. Most similar to our work is TDS [9]—which uses SMC in conjunction with the latent projection technique for conditional guidance. Wu *et al.* use TDS together with FrameDiff for motif scaffolding. However, like SMCDiff [2], TDS has not been applied to other scaffolding tasks. We precisely formalise such tasks to be compatible with existing diffusion posterior samplers and provide non-linear alternatives to the masking approach performed by all of the above methods. While Genie2 [6] uses distance matrices to represent motifs, we additionally break reflection-invariance by considering relative angles between frames.

Chapter 4

Scaffolding by Posterior Sampling

In this chapter, we outline our strategy for scaffolding protein motifs. We formalise each scaffolding task as an inverse problem and then present compatible SMC samplers.

Formally, we define an inverse problem as $\mathbf{y} = \mathscr{A}(\mathbf{x}) + \mathbf{n}$, where $\mathbf{y} \in \mathbb{R}^d$, $\mathbf{x} \in \mathbb{R}^D$, and $\mathbf{n} \sim \mathcal{N}(\cdot;\ 0,\ \sigma^2\mathbf{I}_d)$. Most of the scaffolding tasks are intrinsically linear, i.e. $\mathscr{A} = \mathbf{A} \in \mathbb{R}^{d\times D}$, but we tackle its weaknesses through non-linear extensions. Hence, our general strategy is to seek appropriate expressions for $\mathscr{A}$. Adapting this formulation allows diffusion posterior samplers to transform latent variables into a sequence of observations. We work with the flattened representation $\mathbf{x} \in \mathbb{R}^{3L}$, a protein backbone with L residues, each represented by its three-dimensional C-α coordinates.

4.1 Motif Scaffolding

Given a protein backbone $\mathbf{x} \in \mathbb{R}^{3L}$, we define the motif and scaffold index sets $\{\mathscr{M}, \mathscr{S}\}$ as a partition over all backbone coordinates $\{1, \ldots, 3L\}$, with all three coordinates of each residue belonging to the same index set. Furthermore, we assume $\{\mathscr{M}_i\}_{i=0}^{|\mathscr{M}|}$ and $\{\mathscr{S}_i\}_{i=0}^{|\mathscr{S}|}$ are ordered according to residue number and coordinate axis. Presented with a motif $\mathbf{m} \in \mathbb{R}^{|\mathscr{M}|}$, the motif scaffolding problem involves sampling from the distribution $p(\mathbf{x}_{\mathscr{S}} \mid \mathbf{x}_{\mathscr{M}} = \mathbf{m})$. This is analogous to inferring the entire protein backbone $\mathbf{x}$ given a partial observation of it $\mathbf{x}_{\mathscr{M}}$.

In setting $\mathscr{A} := \mathbf{A}$ to be a masking operation over $\mathbf{x}$, we can frame this as a linear inverse problem. Here, we have the observation $\mathbf{y} := \mathbf{m}$, imposing the motif's position, and the linear transformation $\mathbf{A} = \mathbf{A}_{\mathscr{M}} \in \mathbb{R}^{|\mathscr{M}|\times 3L}$, given by $(\mathbf{A}_{\mathscr{M}})_{ij} := \delta_{\mathscr{M}_i, j}$. To see this, note that

$$(\mathbf{A}_{\mathscr{M}}\mathbf{x})_i = \sum_{j=1}^{3L} \mathbf{A}_{ij}\mathbf{x}_j = \sum_{j=1}^{3L} \delta_{\mathscr{M}_i, j}\mathbf{x}_j = \mathbf{x}_{\mathscr{M}_i}.$$

We refer to this as the **masking** approach.

Notably, Wu *et al.* [9] have shown improved performance by accounting for the (rotational) orientation of the motif. A possible explanation is that we impose a narrower path towards the posterior distribution without the additional degree of freedom. Subsequently, Lin *et al.* [6] used distance matrices as input to their conditionally-trained protein diffusion model. Motivated by this, we propose a **distance** approach. As a start, we express a quadratic transformation as

$$\mathscr{A}(\mathbf{x}) = \sum_{i=1}^{d} (\mathscr{A}(\mathbf{x}))_i \mathbf{e}_i := \sum_{i=1}^{d} \mathbf{x}^\top \mathbf{A}_i \mathbf{x} \mathbf{e}_i, \tag{4.1}$$

where $\{\mathbf{A}\}_{i=1}^{d} \in \mathbb{R}^{D \times D}$ is a sequence of matrices and $\mathbf{e}_i \in \mathbb{R}^d$ is the ith standard basis vector. To condition on distances, we first declare an ordering $o : \mathbb{N} \to \{(j,k) \mid 1 \leq j \leq k \leq |\mathscr{M}|\}$ over all unique pair-combinations of motif residues. Then, for $o(i) = (j,k)$, we set

$$\mathbf{y}_i = \mathbf{y}_{\mathscr{M},i} := \sum_{l=1}^{3} (\mathbf{m}_{3\mathscr{M}_j - l + 1} - \mathbf{m}_{3\mathscr{M}_k - l + 1})^2,$$

$$(\mathbf{A}_i)_{m,n} = (\mathbf{A}_{\mathscr{M},i})_{m,n} := \sum_{l=1}^{3} (\delta_{m,3\mathscr{M}_j - l + 1} - \delta_{m,3\mathscr{M}_k - l + 1})(\delta_{n,3\mathscr{M}_j - l + 1} - \delta_{n,3\mathscr{M}_k - l + 1}).$$

Essentially, we compute all $d = \binom{|\mathscr{M}|}{2}$ pairwise distances between the C-α atoms in the motif region $\mathbf{x}_{\mathscr{M}}$ and match them against the true distances within the motif. Contrary to masking, this approach yields an $E(3)$-invariant motif representation, treating the motif as unique up to translations, rotations, and reflections. This indifference to reflections, however, violates the chirality of proteins. This drawback is explored in our experiments.

To address this, we break the reflection-invariance by further conditioning on the backbone's pairwise orientation deviations. Recall that the frame representation of a protein backbone can be derived from its three-dimensional coordinates $\mathbf{x}$. We denote the ith residue then as the pair $(\mathbf{R}_i(\mathbf{x}), \mathbf{T}_i(\mathbf{x}))$ of a rotation matrix and a translation vector with respect to the global frame. The distance approach has effectively conditioned on the pairwise distances between the translation vectors $\mathbf{T}_i(\mathbf{x})$. Likewise, we define distances between the rotation matrices $\mathbf{R}_i(\mathbf{x})$. First, we remove the dependence on the global frame by computing the relative rotations $\mathbf{R}_j(\mathbf{x})^\top \mathbf{R}_k(\mathbf{x})$ for every residue pair (j,k). Appendix A.1 explains this choice. Then, we use the result measuring the cosine of the angular difference between two rotation matrices

$$d_{\cos}(\mathbf{R}_1,\ \mathbf{R}_2) = \frac{\operatorname{trace}\left(\mathbf{R}_1 \mathbf{R}_2^\top\right) - 1}{2},$$

to compare the sample's relative rotations with those of the motif. Finally, appending this to the previous formulation, for an ordering $o(i) = (j,k)$, we set

$$\mathbf{y}_{i+\binom{|\mathscr{M}|}{2}} := \cos(0) = 1, \quad (\mathscr{A}(\mathbf{x}))_{i+\binom{|\mathscr{M}|}{2}} = \mathscr{A}_{\mathscr{M},i+\binom{|\mathscr{M}|}{2}}(\mathbf{x}) := d_{\cos}(\mathbf{R}_{\mathscr{M}_j}(\mathbf{x})^\top \mathbf{R}_{\mathscr{M}_k}(\mathbf{x}),\, \mathbf{R}_j(\mathbf{m})^\top \mathbf{R}_k(\mathbf{m})).$$

We will refer to this as the **frame-based distance** approach. Note that we have a *product-of-experts* likelihood by appending the additional set of constraints. As each of the multiplicands, i.e. the coordinate and the rotation distances, have differing magnitudes, we raise the rotation matrix contribution to a power and make this a hyperparameter η.

4.1.1 Multi-Motif Scaffolding

Similarly, as before, suppose the index sets $\{\mathscr{M}^1, \ldots, \mathscr{M}^N, \mathscr{S}\}$ form a partition over the backbone coordinates and are each ordered according to residue number and coordinate axis. Given motifs $\mathbf{m}_1, \ldots, \mathbf{m}_N$, the multi-motif scaffolding problem requires sampling from the distribution $p(\mathbf{x}_{\mathscr{S}} \mid \mathbf{x}_{\mathscr{M}^1} = \mathbf{m}_1, \ldots, \mathbf{x}_{\mathscr{M}^N} = \mathbf{m}_N)$.

Unlike in the single motif case, the masking approach fixes the motif-to-motif orientations and severely underrepresents the posterior distribution. We can adapt the distance approach to keep each motif's orientation free by only conditioning on inter-residue distances within each motif. This can be achieved by concatenating the observations and their corresponding transformation matrices across all motifs

$$\mathbf{y} := \left[\mathbf{y}_{\mathscr{M}^1}^\top \ldots \mathbf{y}_{\mathscr{M}^N}^\top\right]^\top,$$
$$\{\mathbf{A}_i\}_{i=1}^d := \left\{\mathbf{A}_{\mathscr{M}^1,1}, \ldots, \mathbf{A}_{\mathscr{M}^1,\binom{|\mathscr{M}^1|}{2}},\ \ldots,\ \mathbf{A}_{\mathscr{M}^N,1}, \ldots, \mathbf{A}_{\mathscr{M}^N,\binom{|\mathscr{M}^N|}{2}}\right\},$$

where $\mathbf{y} \in \mathbb{R}^d$ for $d = \sum_{n=1}^N \binom{|\mathscr{M}^i|}{2}$ and $\mathscr{A}$ is defined in terms of each $\mathbf{A}_i$ as in Equation 4.1. To adapt the frame-based distance approach, we similarly append the additional constraints for all $n \in [1, N]$ and $i \in [1, |\mathscr{M}^n|]$,

$$\mathbf{y}_{i+d+\sum_{j=1}^{n-1}\binom{|\mathscr{M}^j|}{2}} = 1, \quad (\mathscr{A}(\mathbf{x}))_{i+d+\sum_{j=1}^{n-1}\binom{|\mathscr{M}^j|}{2}} = \mathscr{A}_{\mathscr{M}^n,i+\binom{|\mathscr{M}^n|}{2}}(\mathbf{x}).$$

4.1.2 Scaffolding with Degrees of Freedom

So far, we assume the motif is located in a specific region $\mathscr{M}$ of the protein. However, this choice may require careful judgement and domain expertise. Instead, we may be interested in allowing the motif to be placed anywhere on the protein.

Let $\mathbb{M}$ be the set of all possible contiguous motif placements. Wu *et al.* [9] parameterised over the likelihood and placed a uniform prior $p(\mathscr{M}) = 1/|\mathbb{M}|$ on the motif placements

$$p(\mathbf{y}\,|\,\mathbf{x}) = \sum_{\mathscr{M}\in\mathbb{M}} p(\mathbf{y}\,|\,\mathbf{x},\mathscr{M})p(\mathscr{M}) = \frac{1}{|\mathbb{M}|}\sum_{\mathscr{M}\in\mathbb{M}} p(\mathbf{y}\,|\,\mathbf{x},\mathscr{M}). \tag{4.2}$$

As $\mathbf{x}$, the computational bottleneck, is reused throughout the summation, the above mixture likelihood incurs minimal overheads. In Appendix A.2, we propose alternative methods for this scaffolding task that we additionally explored but did not evaluate extensively. Note that the inverse problem formulation remains unchanged, meaning our previous results are compatible with this extension.

4.2 Symmetric Generation

For some point symmetry group in $\mathbb{R}^3$, define $\mathscr{G} = \{\mathbf{g}_k\}_{k=0}^{n-1}$ as the set of all its symmetry operations. We consider designing internally symmetric monomers as we focus on diffusion models that produce a single chain. However, our formulation can analogously be applied to models supporting multiple chains to design symmetric oligomers by treating each subunit as a monomer. Suppose the chain is composed of n identical subunits with L divisible by n. In dealing with 3D atom coordinates, $\mathscr{G}$ is a set of transformation matrices in $\mathbb{R}^{3\times 3}$. Without loss of generality, we order $\mathscr{G}$ such that $\mathbf{g}_0$ is the identity matrix. We can then construct the inverse problem by setting $\mathbf{y} := \mathbf{0}$ and $\mathscr{A} := \mathbf{A}_{\mathscr{G}} - \mathbf{I}_{3L}$, where $\mathbf{A}_{\mathscr{G}} \in \mathbb{R}^{3L\times 3L}$ is given by

$$\mathbf{A}_G = \left[\begin{array}{c|c} \operatorname{diag}(\mathbf{g}_0,\ldots,\mathbf{g}_0) & \\ \vdots & 0 \\ \operatorname{diag}(\mathbf{g}_{n-1},\ldots,\mathbf{g}_{n-1}) & \end{array}\right],$$

composed of block diagonals $\operatorname{diag}(\mathbf{g}_k,\ldots,\mathbf{g}_k) \in \mathbb{R}^{3L/n\times 3L/n}$. Effectively, this constrains the generated protein to be identical to several symmetric projections of its first subunit. This implicitly partitions the chain into n contiguous segments representing each subunit. However, one can shuffle the block diagonals between group operations to render the subunits discontiguous. While the formulation above superfluously subtracts $\mathbf{g}_0$ from the identity matrix, we keep it to simplify our upcoming expressions but truncate the matrix in practice. We demonstrate this process for cyclic and dihedral symmetries.

4.2.1 Cyclic and Dihedral Symmetries

Proteins with cyclic symmetry C_n are invariant to any integer multiple rotations of $2\pi/n$ with respect to a given axis. Without loss of generality, we choose to work with the z-axis and accordingly translate $\mathbf{x}$ so its centre-of-mass (CoM) lies on it. Denote $\mathbf{R}_{a,\theta} \in \mathbb{R}^{3\times3}$ as the rotation matrix that rotates a vector anti-clockwise about the a-axis by an angle of θ. As such, we have the set of rotations $\mathcal{G}_{C_n} = \{\mathbf{R}_{z,2\pi k/n}\}_{k=0}^{n-1}$, and the inverse problem for C_n can be defined by $\mathbf{y} := \mathbf{0}$ and $\mathcal{A} := \mathbf{A}_{\mathcal{G}_{C_n}} - \mathbf{I}_{3L}$. While any ordering of the set $\mathcal{G}$ is conducive to producing a cyclic protein, it may be favourable to have adjacent angles for adjacent sub-sequences, e.g. $\mathbf{g}_k = \mathbf{R}_{z,2\pi k/n}$.

Proteins with dihedral symmetry D_n similarly have C_n symmetry in one axis but have C_2 symmetry in another axis orthogonal to the first. We choose the z- and y-axes as primary and secondary axes of symmetry and translate $\mathbf{x}$ to have its CoM at the origin. Thus, we have $\mathcal{G}_{D_n} = \mathcal{G}_{C_n} \cup \{\mathbf{R}_{z,2\pi k/n}\mathbf{R}_{y,2\pi}\}_{k=0}^{n-1}$ and likewise set $\mathbf{y} := \mathbf{0}$ and $\mathcal{A} := \mathbf{A}_{\mathcal{G}_{D_n}} - \mathbf{I}_{3L}$.

4.2.2 Symmetric Motif Scaffolding

In addition to symmetric constraints, we may also condition the existence of a motif. Note that we can assume the motif is local to exactly one subunit and $n-1$ copies exist in the others. Otherwise, if the motif lies on the boundary between subunits, we can redefine the motif to be the residues entirely situated in one subunit. Suppose then that the motif is in the first subunit, i.e. $\mathcal{M}_i \leq 3L/n$ for all i. Hence, we set $\mathbf{y} := \begin{bmatrix}\mathbf{e}_{\mathcal{M}_1} & \dots & \mathbf{e}_{\mathcal{M}_{|\mathcal{M}|}}\end{bmatrix}\mathbf{m}$, where $\mathbf{e}_i \in \mathbb{R}^L$ is the ith standard basis vector, and $\mathcal{A} := \mathbf{A}_{\mathcal{G}} - \mathbf{I}_{3L} + \operatorname{diag}(\mathbb{1}_{\mathcal{M}})$ to define the inverse problem. The additional term unmasks the motif indices and asserts it is equal to the chosen motif $\mathbf{m}$.

4.3 SMC Diffusion Posterior Samplers

With the inverse problems formalised, we can now lay out the general algorithms for each of the chosen posterior samplers in the context of de-noising protein backbones. With SMC, we reiterate that two main design choices are available: the proposals q_t and the intermediate targets γ_t. The samplers we describe differ in their choices for these distributions. Recall that diffusion models are Markovian in their reverse processes and, therefore, define an SSM. We begin by considering BPF, the baseline sampler for SSMs.

4.3.1 Bootstrap Particle Filter

In BPF, we set the proposal q_t to match the DDPM's reverse process and the target γ_t to be the joint distribution of both latent and observed variables. With this choice, the likelihood $g(\mathbf{y}_t \mid \mathbf{x}_t)$ becomes the fitness criteria for filtering proteins. While BPF is general

Algorithm 4.1: Bootstrap Particle Filter for Motif Scaffolding Problems

```
input  : final observation $\mathbf{y}_0$, observation distribution $\psi$, likelihood $g$, no. of
         particles $K$
output : protein backbones $\mathbf{x}_0$ containing the motif
# Create sequence of observations
Sample $\mathbf{y}_{1:T} \sim \psi(\cdot \mid \mathbf{y}_0)$

# Generate protein backbones
Sample $\mathbf{x}_T^{1:K} \sim \mathcal{N}(\cdot;\ \mathbf{0},\ \mathbf{I})$
for $t = T, \dots, 1$ do
    for $i = 1, \dots, K$ do
        Sample $\bar{\mathbf{x}}_{t-1}^i \sim \mathcal{N}\left(\cdot;\ \mu_\theta(\mathbf{x}_t^i, t),\ \Sigma_\theta(t)\right)$   # Reverse diffuse particles
        Set $\tilde{w}_{t-1}^i \leftarrow g(\mathbf{y}_{t-1} \mid \bar{\mathbf{x}}_{t-1}^i)$                  # Evaluate their likelihood
    end
    Set $w_{t-1}^i \leftarrow \tilde{w}_{t-1}^i / \sum_{j=1}^K \tilde{w}_{t-1}^j$, for $i = 1, \dots, K$
    Resample $\mathbf{x}_{t-1}^{1:K} \sim \text{Multinomial}(w_{t-1}^{1:K},\ \bar{\mathbf{x}}_{t-1}^{1:K})$     # Resample particles
end
```

enough to admit any likelihood, we first consider linear inverse problems and revisit the non-linear case in the TDS section.

Our inverse problem states $\mathbf{y}_0 = \mathscr{A}(\mathbf{x}_0) + \mathbf{n}$, with $n \sim \mathcal{N}(\cdot;\ \mathbf{0},\ \sigma^2\mathbf{I})$ and $\mathscr{A} = \mathbf{A}$ in the linear setting. However, our scaffolding formulations only define $\mathbf{y}_0$, e.g. as the motif. To construct the rest of the observations $\mathbf{y}_{1:T}$, we sample them from what we will call the *observation* distribution $\psi(\cdot \mid \mathbf{y}_0)$. We may define ψ by applying the transformation $\mathbf{A}$ onto the DDPM's forward process to get

$$\psi(\mathbf{y}_t \mid \mathbf{y}_{t-1}) = \mathcal{N}(\mathbf{y}_t;\ \sqrt{1-\beta_t}\mathbf{y}_{t-1},\ \beta_t \mathbf{A}\mathbf{A}^\top), \quad \psi(\mathbf{y}_{1:T} \mid \mathbf{y}_0) = \prod_{t=1}^{T} \psi(\mathbf{y}_t \mid \mathbf{y}_{t-1}). \tag{4.3}$$

The likelihood can then be derived using the above $\mathbf{y}_t$ sequence as

$$g_{mask}(\mathbf{y}_t \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t;\ \mathbf{A}\mathbf{x}_t,\ \sigma^2 \bar{\alpha}_t \mathbf{I}). \tag{4.4}$$

This technique is what we previously referred to as observation projection. To illustrate in the case of regular motif scaffolding, when $\mathbf{y}_0 = \mathbf{m}$, the motif is forward diffused to build the sequence of observations, and the protein backbones that de-noise most similarly to the motif are favoured for resampling. The algorithm is summarised more generally in Algorithm 4.1 for any ψ and g.

Algorithm 4.2: Filtering Posterior Sampling for Motif Scaffolding Problems

```
input  : final observation $\mathbf{y}_0$, masking matrix $\mathbf{A}$, no. of particles $K$
output : protein backbones $\mathbf{x}_0$ containing the motif
# Create sequence of observations
Sample $\epsilon_T \sim \mathcal{N}(\cdot;\ \mathbf{0},\ \mathbf{I})$
Set $\mathbf{y}_T \leftarrow \mathbf{A}\epsilon_T$                        # Share noise with initial state
Sample $\mathbf{y}_{1:T-1} \sim \psi_{FPS}(\cdot \mid \mathbf{y}_T, \mathbf{y}_0, \mathbf{A})$

# Generate protein backbones
Set $\mathbf{x}_T^i \leftarrow \epsilon_T$, for $i = 1, \ldots, K$
for $t = T, \ldots, 1$ do
    for $i = 1, \ldots, K$ do
        Sample $\bar{\mathbf{x}}_{t-1}^i \sim \mathcal{N}(\cdot;\ \mu_{FPS}\left(\mathbf{x}_t^i, \mathbf{y}_{t-1}, t\right),\ \Sigma_{FPS}(t))$      # Optimal proposal
        Set $\tilde{w}_{t-1}^i \leftarrow g_{mask}(\mathbf{y}_{t-1} \mid \bar{\mathbf{x}}_{t-1}^i) p_\theta(\bar{\mathbf{x}}_{t-1}^i \mid \mathbf{x}_t^i) / p_\theta(\bar{\mathbf{x}}_{t-1}^i \mid \mathbf{x}_t^i, \mathbf{y}_{t-1})$   # Compute weights
    end
    Set $w_{t-1}^i \leftarrow \tilde{w}_{t-1}^i / \sum_{j=1}^K \tilde{w}_{t-1}^j$, for $i = 1, \ldots, K$
    Resample $\mathbf{x}_{t-1}^{1:K} \sim \text{Multinomial}(w_{t-1}^{1:K},\ \bar{\mathbf{x}}_{t-1}^{1:K})$         # Resample particles
end
```

4.3.2 Filtering Posterior Sampling

To improve the efficiency of BPF for linear inverse problems, we can choose q_t to be the locally **optimal proposal** q_t^*, better estimating the target with fewer particles. This notion is formalised by minimising the KL-divergence between $\gamma_{t-1}(\mathbf{x}_{1:t-1})q_t(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})$ and $\gamma_t(\mathbf{x}_{1:t})$. For SSMs, this is a known result with

$$q_t^*(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{y}_t) = f(\mathbf{x}_t \mid \mathbf{x}_{t-1}) g(\mathbf{y}_t \mid \mathbf{x}_t).$$

Dou and Song [8] analytically derived the optimal proposal in the diffusion context using Equations 4.3, 4.4 as the observation sequence and likelihood. It is given by

$$\begin{aligned} q^*(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{t-1}) &= \mathcal{N}(\mathbf{x}_{t-1};\ \mu_{FPS}(\mathbf{x}_t, \mathbf{y}_{t-1}, t),\ \Sigma_{FPS}(t)), \\ \Sigma_{FPS}(t) &= \left(\Sigma_\theta(t)^{-1} + \frac{1}{\sigma^2 \bar{\alpha}_{t-1}} \mathbf{A}^\top \mathbf{A}\right)^{-1}, \\ \mu_{FPS}(\mathbf{x}_t, \mathbf{y}_{t-1}, t) &= \Sigma_{FPS}(t) \left(\Sigma_\theta(t)^{-1} \mu_\theta(\mathbf{x}_t, t) + \frac{1}{\sigma^2 \bar{\alpha}_{t-1}} \mathbf{A}^\top \mathbf{y}_{t-1}\right). \end{aligned}$$

However, instead of forward-noising $\mathbf{y}_0$, they perform a noise-sharing technique by setting $\mathbf{y}_T = \mathbf{A}\mathbf{x}_T$ and building the sequence backwards with

$$\psi_{FPS}(\mathbf{y}_{t-1} \mid \mathbf{y}_t, \mathbf{y}_0, \mathbf{A}) = \mathcal{N}\left(\mathbf{y}_{t-1};\ \sqrt{\bar{\alpha}_{t-1}} y + \sqrt{\tfrac{(1-c)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}} (\mathbf{y}_t - \sqrt{\bar{\alpha}_t} y),\ c(1-\bar{\alpha}_{t-1}) \mathbf{A}^\top \mathbf{A}\right)$$

Algorithm 4.3: Twisted Diffusion Sampler for Motif Scaffolding Problems

```
input  : final observation $\mathbf{y}_0$, likelihood $g$, guidance scale $\gamma$, no. of particles $K$
output : protein backbones $\mathbf{x}_0$ containing the motif
Sample $\mathbf{x}_T^{1:K} \sim \mathcal{N}(\cdot;\ \mathbf{0},\ \mathbf{I})$
Set $w_T^i \leftarrow g(\mathbf{y}_0 \mid \mathbf{x}_T^i)$ for $i = 1, \dots, K$
Resample $\mathbf{x}_{T-1}^{1:K} \sim \text{Multinomial}(w_T^{1:K},\ \mathbf{x}_T^{1:K})$
for $t = T-1, \dots, 1$ do
    for $i = 1, \dots, K$ do
        Set $s_t^i = s_\theta(\mathbf{x}_t^i, t) + \gamma \nabla_{\mathbf{x}_t^i} \log g(\mathbf{y}_0 \mid \hat{\mathbf{x}}_0(\mathbf{x}_t^i, t))$   # Conditional score
        Sample $\bar{\mathbf{x}}_{t-1}^i \sim \mathcal{N}\left(\cdot;\ \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t + (1-\alpha_t)s_t^i\right),\ \Sigma_\theta(t)\right)$   # Optimal Proposal
        Set $\tilde{w}_{t-1}^i \leftarrow g(\mathbf{y}_0 \mid \bar{\mathbf{x}}_{t-1}^i) p_\theta(\bar{\mathbf{x}}_{t-1}^i \mid \mathbf{x}_t^i) / \left[g(\mathbf{y}_0 \mid \bar{\mathbf{x}}_t^i) p_\theta(\bar{\mathbf{x}}_{t-1}^i \mid \mathbf{x}_t^i, \mathbf{y}_0)\right]$ # Compute weights
    end
    Set $w_{t-1}^i \leftarrow \tilde{w}_{t-1}^i / \sum_{j=1}^K \tilde{w}_{t-1}^j$, for $i = 1, \dots, K$
    Resample $\mathbf{x}_{t-1}^{1:K} \sim \text{Multinomial}(w_{t-1}^{1:K},\ \bar{x}_{t-1}^{1:K})$   # Resample particles
end
```

for some tunable parameter $c \in [0, 1]$. We choose $c = \beta_t/(1-\bar{\alpha}_{t-1})$ to match our DDPM variance $\Sigma_\theta(t) = \beta_t \mathbf{I}$. The weight computation is also updated when using the optimal proposal. We omit the details but present the algorithm fully in Algorithm 4.2.

4.3.3 Twisted Diffusion Sampler

One challenge in extending our methods to non-linear inverse problems is the infeasibility of constructing the sequence of observations $\mathbf{y}_t$. To circumvent this, we can keep our observations fixed, i.e. $\mathbf{y}_t = \mathbf{y}_0$, and simply use the predicted fully-denoised protein

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t)\right),$$

to approximate $\mathbf{x}_0$. Then, given any likelihood g, we have

$$g(\mathbf{y}_0 \mid \mathbf{x}_t) \approx g\left(\mathbf{y}_0 \mid \hat{\mathbf{x}}_0(\mathbf{x}_t, t)\right) = \mathcal{N}\left(\mathbf{y}_0;\ \mathscr{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t, t)),\ \tilde{\sigma}_t^2 \mathbf{I}\right).$$

This technique is what we previously referred to as latent projection. Wu *et al.* [9] recommend setting $\tilde{\sigma}_t^2 := \text{Var}[\mathbf{x}_t \mid \mathbf{x}_0]$, making the filtering criteria lenient in the initial parts of the reverse process, when the projection $\hat{\mathbf{x}}_0$ is still unreliable, and tightening the filter at the end to satisfy the conditions. To keep it non-zero we set $\tilde{\sigma}_t^2 = (1-\bar{\alpha}_t) + \sigma^2$, so we precisely have our inverse problem formulation at $t = 0$. Now, we can adapt our distance-based approach to have the likelihood

$$g_{dist}(\mathbf{y}_0 \mid \mathbf{x}_t) \coloneqq \mathcal{N}\left(\mathbf{y}_0;\ \sum_{i=0}^{d} \hat{\mathbf{x}}_0(\mathbf{x}_t, t)^\top \mathbf{A}_i \hat{\mathbf{x}}_0(\mathbf{x}_t, t)\mathbf{e}_i,\ \tilde{\sigma}_t^2 \mathbf{I}\right).$$

Similarly, we have the frame-based distance likelihood

$$g_{frame\_dist}(\mathbf{y}_0 \mid \mathbf{x}_t) \coloneqq g_{dist}\left(\mathbf{y}_{0,1:d/2} \mid \mathbf{x}_t\right) g_{rot\_dist}\left(\mathbf{y}_{0,d/2+1:d} \mid \mathbf{x}_t\right)^\eta,$$

where, for an ordering $o(i) = (j, k)$, motif $\mathbf{m}$, and motif index set $\mathcal{M}$, we have

$$g_{rot\_dist}(\cdot \mid \mathbf{x}_t) \coloneqq \mathcal{N}\left(\cdot;\ \sum_{i=1}^{|\cdot|} d_{\cos}\left(\mathbf{R}_{\mathcal{M}_j}(\hat{\mathbf{x}}_0(\mathbf{x}_t, t))^\top \mathbf{R}_{\mathcal{M}_k}(\hat{\mathbf{x}}_0(\mathbf{x}_t, t)),\ \mathbf{R}_j(\mathbf{m})^\top \mathbf{R}_k(\mathbf{m})\right)\mathbf{e}_i,\ \tilde{\sigma}_t^2 \mathbf{I}\right).$$

To compute the optimal proposal $q^*(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_0)$, we can alternatively find the conditional score. While intractable, we can approximate it with the score and likelihood

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}_0) &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log g(\mathbf{y}_0 \mid \mathbf{x}_t) \\ &\approx s_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log g(\mathbf{y}_0 \mid \hat{\mathbf{x}}_0(\mathbf{x}_t, t)).\end{aligned}$$

Now, the proposal is effectively reverse-diffusing the particles but using the conditional score instead of the score. Similar to guidance, we can magnify the conditional signal by scaling the guidance term by some γ. The full algorithm is given in Algorithm 4.3. We remark that, in practice, the gradient is computed via automatic differentiation and thereby requires a differentiable $\mathcal{A}$. For more information, refer to Appendix A.3.

Chapter 5

Experimental Setup

In this chapter, we discuss our setup for solving the different scaffolding tasks and evaluating the generated protein backbones. Our methods are summarised in Figure 5.1.

5.1 In Silico Evaluation Strategy

Similar to existing works [2, 1, 4], we use an in silico **self-consistency pipeline** for measuring the designability of protein backbones. The procedure involves an inverse-folding network and a structure prediction network. We use ProteinMPNN [25] and ESMFold [26], respectively. Each generated backbone is first processed by the inverse-folding network and is predicted by its representative amino-acid sequences. The structure prediction network then folds eight of these sequences to produce the *predicted structures*. The **self-consistency root mean squared deviation (scRMSD)** between the generated and predicted backbones are then computed and the smallest is reported together with

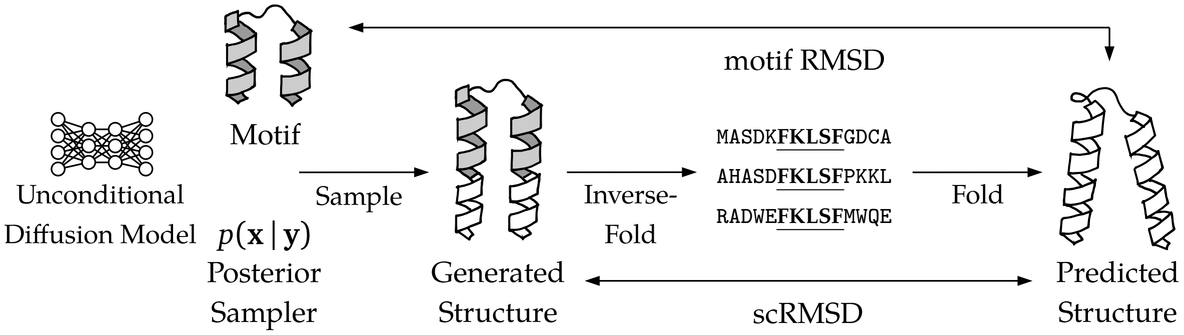

Figure 5.1: An overview of the motif scaffolding experimental setup. Protein backbones are first sampled from the conditional setup with the motif as an observation. These generated structures are then inverse-folded with the motif sequence fixed and folded back into structures. Finally, metrics such as self-consistency RMSD and motif RMSD are computed between the predicted structure and both the generated structure and the motif.

the predicted design's **predicted local distance difference test (pLDDT)**—a measure of confidence by the structure prediction network. The premise of this technique is that generated backbones possessing natural structures are likely to be represented consistently across orthogonal methods. We use the available in-silico design pipeline provided by the authors of Genie2[1].

5.1.1 Designability and Diversity Metrics

We adopt a similar designability criterion as Lin *et al.* [6]. We consider a protein backbone as **designable** if it deviates with the most similar predicted design by at most two Angstroms ($\text{scRMSD} \leq 2\text{A}$) and if the designs are confidently predicted ($\text{pLDDT} \geq 70$). For motif scaffolding tasks, we consider a scaffold to be successful if it is designable as above, the motif is present in the predicted backbone within one Angstrom in alignment deviation (motif $\text{RMSD} \leq 1\text{A}$), and there is a low predicted alignment error (pAE) between residues ($\text{pAE} \leq 5$)—another confidence metric of the structure prediction network. For multi-motif scaffolding, every motif must be within one Angstrom. We remark that scRMSD in motif scaffolding differs from the unconditional setting, as inverse-folded sequences are conditioned to have the motif's sequence.

Furthermore, as success rates can be misleading with identically designed structures, we also measure sample **diversity**. Again, similar to Lin *et al.*, we group designs using single-linkage hierarchical clustering with a distance threshold given by a TM-score of 0.6. We report the number of unique successful scaffolds for motif scaffolding tasks.

5.1.2 Benchmark Problems

The 25 motif problems curated by Watson *et al.* [1] cover a diverse range of motifs and are standard for motif scaffolding evaluation. We test our methods here but exclude problem 6VW1, which involves scaffolding multiple chains. For multi-motif scaffolding, we test against the six problems curated by Lin *et al.* [6], one of which requires up to four motifs to be scaffolded. In both benchmarks, we fix the motif placement by taking the median of scaffold length ranges in their specifications like Wu *et al.* [9]. We isolate this variability to provide a less stochastic comparison over the methods.

Given our intention of using an unconditional model that only produces single chains, we focus on generating internally symmetric monomers for symmetric scaffolding. We test our symmetric formulation by measuring the designability of generated monomers up to 128 and 256 residues long under various point symmetries. Because of time, we

[1] The repository is available at https://github.com/aqlaboratory/insilico_design_pipeline.

were unable to sufficiently test our symmetric motif scaffolding formulation and leave this for future work.

Detailed information on the specifications of each problem is given in Appendix B.1

5.2 Conditional Setup

We pair an unconditional model with an SMC sampler to produce protein backbones under a conditional signal without conditional training.

5.2.1 Unconditional Diffusion Model

In our analyses, we use Genie [4], an unconditional protein backbone diffusion model. We make this choice for its simplicity in its forward noising process and its relatively small model size. In particular, we use Genie-SCOPe-128 and Genie-SCOPe-256, models trained on proteins from the SCOPe dataset [27], capable of generating proteins of up to 128 and 256 residues, respectively. For motif scaffolding problems, where the overall length is at most 128 residues, we use Genie-SCOPe-128 due to quicker inference times. For the multi-motif case, which contains problems requiring longer samples, we use Genie-SCOPe-256. We use both models to test our symmetric formulations and limit generated proteins to 128 and 256 residues. We test with two lengths to see the impact of having short and long asymmetric subunits. In each of the models, we de-noise samples for $T = 1000$ steps.

An important factor to consider is the model's temperature scale $\zeta \in [0, 1]$ that controls the amount of noise added to each step in the reverse process. High ζ leads to more diverse samples, and low ζ typically yields higher-quality samples. The Genie models have been shown to attain the best F1 designability-diversity score at around $\zeta = 0.4$. However, how this affects conditional samplers wrapped around the model is not well understood. We, therefore, additionally test the effects of ζ across our samplers.

5.2.2 Diffusion Posterior Samplers

We choose to examine the performance of six posterior samplers across the different motif scaffolding benchmarks. They are:

1. (**BPF-FW**) BPF with forward noising $\psi(\cdot \mid \mathbf{y})$,

2. (**BPF-BW**) BPF with backward noising $\psi_{FPS}(\cdot \mid \mathbf{y})$,

3. (**FPSSMC**) FPS-SMC,

4. (**TDS-MASK**) TDS under a masking approach,

5. (**TDS-DIST**) TDS under a distance approach,

6. (**TDS-FRAME-DIST**) TDS under a frame-based distance approach.

Here, we have an equal split of observation and latent projection methods and a comparison between the three motif representations. While BPF can be modified to have a latent-projection likelihood, we retrieve a similar formulation but with twisting when TDS has guidance scale $\gamma = 0$, which is already considered in our hyperparameter tuning. BPF and FPSSMC are configured to solve linear inverse problems and thereby operate under the masking approach.

For multi-motif benchmarks, we use TDS-DIST and TDS-FRAME-DIST, as they are the only methods suitable for independently modelling the orientations of different motifs. For symmetric generation, we use FPSSMC and TDS-MASK given the linearity of the inverse problem. While we can swap masking for distances in our symmetric motif scaffolding framework, we choose the more straightforward approach.

We fix the likelihood standard deviation $\sigma = 0.05$ to define the same inverse problem in all methods. We also fix the number of particles to be $K = 8$. We perform adaptive (residual) resampling and resample only when $\text{ESS}_t \leq K/2$. With each sampler having hyperparameters, we search the most performant based on two motif problems: 3IXT and 1PRW—a high and a low success rate problem, respectively. Moreover, 3IXT is contiguous, whereas 1PRW is not. Appendix B.2 further accounts optimisations made to speed up inference.

Chapter 6

Results and Discussion

In this chapter, we present the results of our experiments. We proceed through each scaffolding task in order and, where relevant, highlight supporting experiments.

6.1 Motif Scaffolding

We sampled proteins conditioned on motifs from the RFDiffusion motif scaffolding benchmark [1]. Ten of the 23 motif problems had at least one successful solution among the 32 designs generated for each problem. Figure 6.1 summarises the performance of samplers across the benchmarks. Among the methods, TDS-MASK and TDS-DIST solved the most, with eight problems each. Furthermore, TDS-DIST and TDS-FRAME-DIST showed a comparable, if not higher, number of unique successes over TDS-MASK for several problems. On the one hand, this shows the viability of masking for single-motif scaffolding, maintaining the linearity of the inverse problem and being broadly applicable to many posterior samplers. On the other hand, the non-linear distance approaches, with their comparable performance and generalisability to the multi-motif case, present a case for being a drop-in replacement.

BPF-FW and BPF-BW were two methods that used the reverse process as their proposal. They yielded low scRMSD but high motif RMSD, resulting in low success rates. This was because the overall de-noising process mostly remained unchanged, with the conditional signal only present during resampling. For this reason, we explored FPSSMC to provide a stronger signal through its optimal proposal. However, while it attained a lower motif RMSD, it did not improve success due to its higher scRMSD, prioritising the motif's appearance over the global integrity of the protein. We suspect a larger likelihood variance would reduce the resampling frequency and help mitigate this issue, but remark that this value has been fixed in all the methods.

Overall, the methods performed better when projecting latent variables instead of

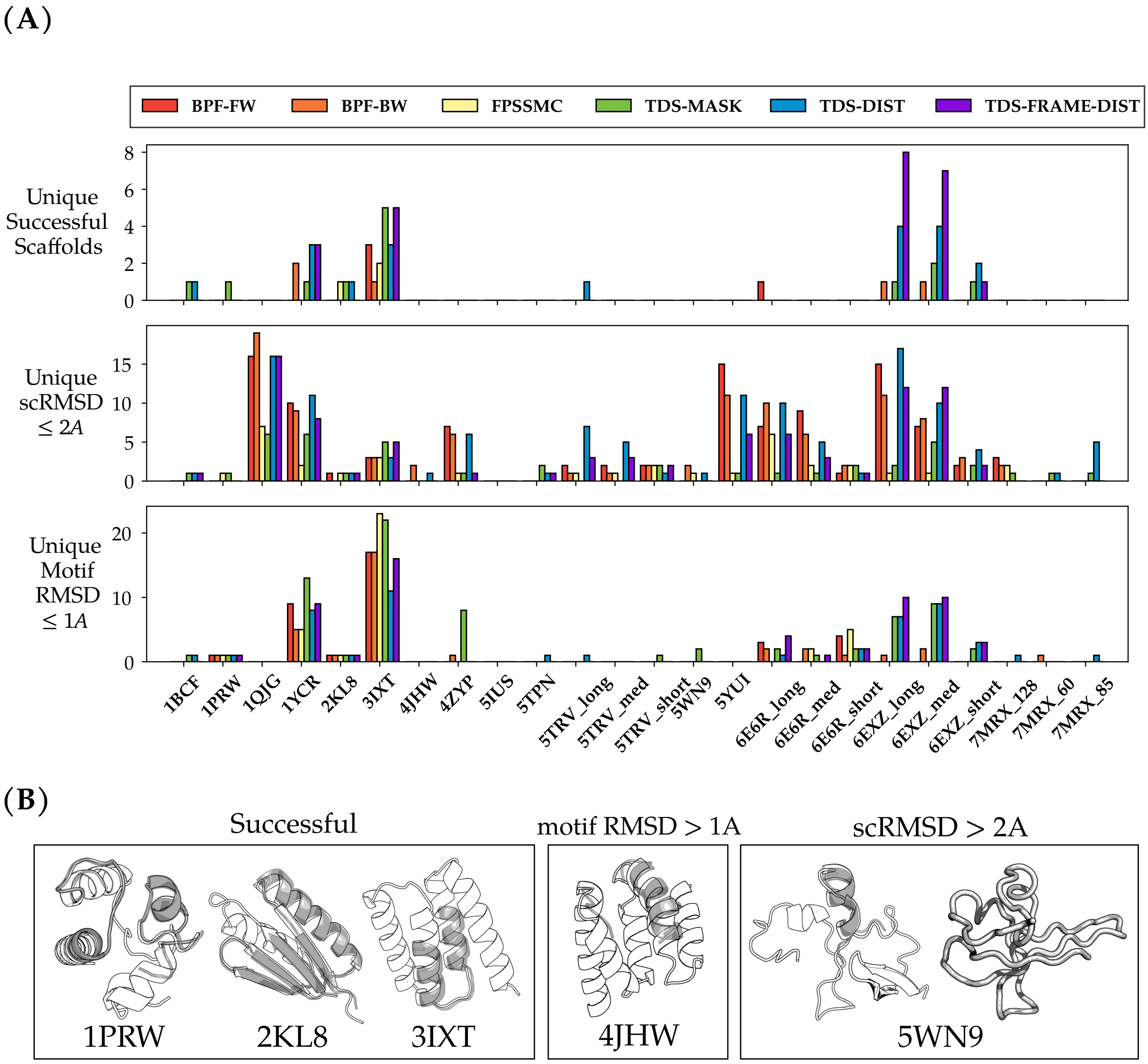

Figure 6.1: (A) Performance of sampling methods on the 24 motif scaffolding benchmarks. Thirty-two backbones are sampled from each method across all the motif problems. Scaffolds that are successful and those which meet at least one of the main success criteria are reported according to their unique count. **(B) Examples of the designed scaffolds.** The motif, in grey, is aligned with the scaffold, in white. Most unsuccessful scaffolds either do not possess the motif in full or have poor self-consistency.

observations. The asymmetry between the forward and reverse processes may have made it unlikely for a sequence of observations generated through the forward process to be matched by the backbone while it was being de-noised. Latent-projections were less sensitive to this asymmetry, given they do not use the forward process.

As the motif placements were fixed for each problem, it is possible that some had placements that were too restrictive to satisfy. In this scenario, it is sensible to compound our methods with the mixture likelihood in Equation 4.2 to consider multiple place-

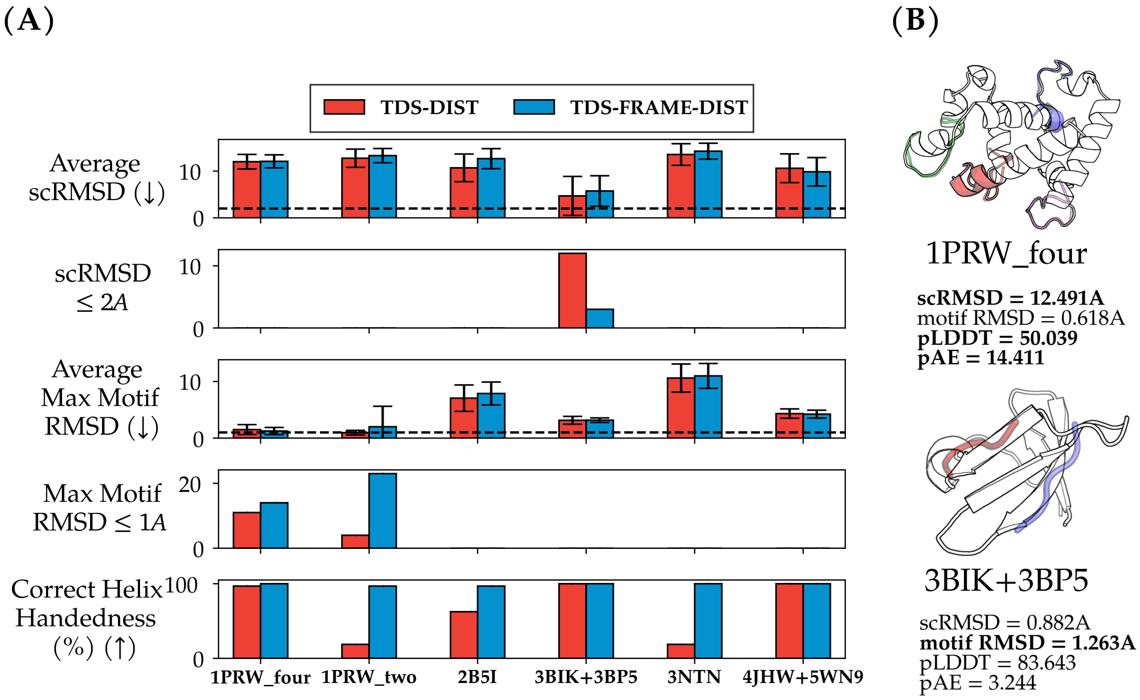

Figure 6.2: (A) Success metrics of sampling methods on the six multi-motif scaffolding benchmarks. Thirty-two scaffolds are sampled for each problem. Values for a pass in each criterion are denoted by the dashed line. Error bars shown are one standard deviation from the mean. Only samples with correct handedness were considered. **(B) Examples of the designed scaffolds.** The motifs, in colour, are aligned with the scaffold, in white. Metrics which violate the success criteria are in bold.

ments simultaneously. However, we could not pursue this path extensively due to time. Appendix C.1.3 documents our preliminary results for several unsolved motif problems.

Altogether, a larger number of samples need to be performed to provide more conclusive results. We remark that state-of-the-art methods yield as little as one unique success in some motif problems for over a thousand samples [6].

6.2 Multi-Motif Scaffolding

We tested our methods on the six multi-motif benchmark problems from Genie2. While none of the methods succeeded, some designs narrowly missed the success criteria. Figure 6.2 summarises the results.

Here, the dynamic between the distance and frame-based distance approaches are shown in full display. For example, in problem 3BIK+3BP5, we found that TDS-DIST had more designable scaffolds than TDS-FRAME-DIST. This was because the motifs were sufficiently flat in one dimension, identical to their reflections. Yet, TDS-FRAME-DIST still allots a substantial percentage of guidance to correct for handedness and impedes progress on meeting the distance constraints. In all the problems, however, TDS-FRAME-DIST maintained correct helix handedness, whereas TDS-DIST did not.

While it is reasonable to expect that TDS-DIST is more likely to produce at least one reflected motif with an increasing number of motifs, we found that discontiguity in motifs was the main hindrance to its performance. This is especially apparent in problems 1PRW_four and 1PRW_two. Although the first has four contiguous motifs, the second has two discontiguous motifs, prompting a worse performance for TDS-DIST. The case is the same for problems 2B5I and 3NTN, which also have discontiguous motifs. Hence, η, TDS-FRAME-DIST's likelihood contribution scale for handedness, could be adjusted more appropriately to account for this fact.

We hypothesise that problems such as 3BIK+3BP5 can already be solved with more samples. However, feasible design specifications are more important than ever, as additional motifs greatly restrict the allowable conformations of the protein. General improvements can potentially be made by sampling motif placements or compounding the formulation with the mixture likelihood to consider multiple placements at once. To our knowledge, this is the first attempt to scaffold multiple motifs without conditional training, and while unsuccessful, it shows some room for improvement.

6.3 Symmetric Generation

We generated internally symmetric monomers for cyclic and dihedral point symmetries. Several met the designability criteria as shown in Figure 6.3. Designs with at most 128 residues were unsuccessful at higher orders of symmetry as the asymmetric subunits became increasingly short. Those with at most 256 residues had a similar trend but were more successful altogether. Of note, higher orders of cyclic symmetry often had β-sheets surrounding the region closest to the axis of symmetry, with several C_8 designs resembling TIM barrels.

While, unlike RFDiffusion, our method implicitly imposes symmetry, the generated backbones were all symmetric. We attribute this to the tight inverse problem variance $\sigma^2 = 0.0025$ and the nature of FPSSMC and TDS samplers to guide the backbone in sufficiently meeting the inverse problem formulation. This implicit approach has some advantages over explicitly symmetrising backbones at each step. First, it allows for control over looser symmetries by increasing the variance σ^2. This widens the target sample space to include commonly observed monomeric proteins with non-exact internal symmetries. This, however, may not be as applicable to symmetric oligomer generation. Second, it

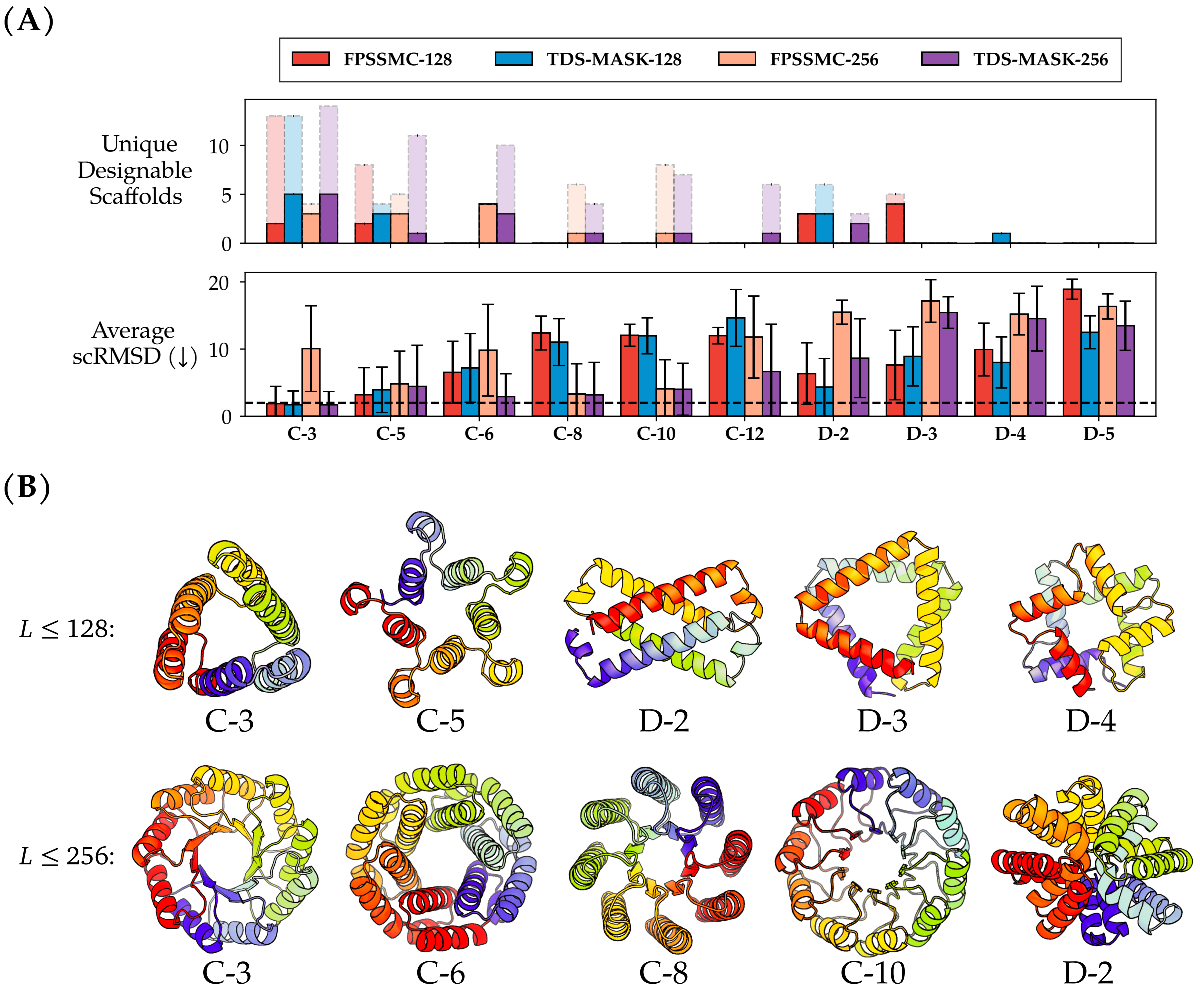

Figure 6.3: **(A) Designability of symmetric designs across several point symmetries**. Sixteen scaffolds with a maximum of 128 and 256 residues were sampled for each symmetry through FPSSMC and TDS-MASK. The total number of designable scaffolds is dashed atop the unique count. The success threshold for scRMSD is indicated by the dashed line. **(B) Examples of the successfully designed scaffolds.** The first and second rows show designs with a maximum of 128 and 256 residues, respectively. The primary axis of symmetry points directly outwards of the page.

is composable with other constraints without having to orient the asymmetric subunits at each step. This enables symmetries to form without fixing distances from the axes of symmetry.

Chapter 7

Conclusion

In summary, we formalised various motif scaffolding tasks and adapted diffusion posterior samplers to work with them. Our setup was able to produce successful scaffolds for several motif problems. In addition, our frame-based distance representation of the motif provided comparable performance with the conventional masking approach but further generalised to the case of multiple motifs. While our setup was unsuccessful there, we believe scaling the number of samples or sampling motif placements is enough to solve some of the current multi-motif problems. Internally symmetric monomers were also successfully designed by our setup. This work has demonstrated generation capabilities for several scaffolding tasks without conditional training. We believe our setup's performance can be significantly improved through further work.

7.1 Limitations and Future Work

Due to time, we were unable to produce a large number of replicates for the scaffolding benchmarks. This hurts our performance, as some motif problems may demand hundreds of samples for a single success. But, at the same time, this artificially inflates sample diversity, as the metric is best stress-tested with sufficiently many designs, with diversity tending to zero as the sample size increases. We point out, however, that our fixing of the motif placement means diversity is also negatively affected. Furthermore, it is difficult to compare against other methods as they either have different, often less stringent, designability criteria, do not report sample diversity, or use a different unconditional model.

To further assess the different formulations, it would be beneficial to compare existing works on three fronts. First, the masking approach can be parameterised by considering finitely many random orientations of the motif with improved performance [9]. However, whether these benefits outperform the distance approaches has not been tested. Second, with motif placements, it is unclear whether it is better to randomly sample a configuration and fix the motif or to consider all possible configurations as it is being de-noised. Third, with symmetric generation, a comparison between an implicit and an explicit symmetric constraint has yet to be performed. While the implicit method easily stacks with other constraints, explicitly symmetrising the backbone provides guarantees and does not require optimising for symmetry.

Hyperparameters were also a challenge in this work. The massive combinatorial space of parameters was not fully explored and can be investigated further.

We additionally point out that the likelihood measures have a weakness—they only act upon the motif regions within the backbone. The guidance term, therefore, has zero contribution to the rest of the scaffold. For large guidance scales, it is hence possible for motifs to be at an unnaturally large distance away from the scaffold. A more "globally-acting" likelihood through heuristics or distance constraints on adjacent residues should help anchor the motif in place. Additionally, the frame-based distance approach for conditioning motifs may be improved by exploring variations in the likelihood.

Another interesting extension is to fix the computations available and determine which ratio between the number of particles and unique samples will yield the best success rate. In our case, we have fixed the number of particles.

Finally, recent trends include a quicker generative modelling paradigm in flow-matching, all-atom models for proteins, and joint sequence and structure modelling. Our methods may be extended to support design tasks in any of these directions.

Bibliography

[1] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[2] Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi S Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2022.

[3] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pages 40001–40039. PMLR, 2023.

[4] Yeqing Lin and Mohammed Alquraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. In *International Conference on Machine Learning*, pages 20978–21002. PMLR, 2023.

[5] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[6] Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

[7] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2022.

[8] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem

solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2023.

[9] Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2023.

[10] Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

[11] Arnaud Doucet, Adam M Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[14] Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 15(1):1059, 2024.

[15] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in neural information processing systems*, 31, 2018.

[16] 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.

[17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Nineth International Conference on Learning Representations*, 2020.

[19] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *The Tenth International Conference on Learning Representations*, 2021.

[20] Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023.

[21] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *The Twelfth International Conference on Learning Representations*, 2023.

[22] Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

[23] Kieran Didi, Francisco Vargas, Simon V Mathis, Vincent Dutordoir, Emile Mathieu, Urszula J Komorowska, and Pietro Lio. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design. *arXiv preprint arXiv:2312.09236*, 2023.

[24] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[25] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

[26] 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.

[27] Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. Scope: Structural classification of proteins—extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2014.

[28] Andrew C Hunt, James Brett Case, Young-Jun Park, Longxing Cao, Kejia Wu, Alexandra C Walls, Zhuoming Liu, John E Bowen, Hsien-Wei Yeh, Shally Saini, et al. Multivalent designed proteins neutralize SARS-CoV-2 variants of concern and confer protection against infection in mice. *Science translational medicine*, 14(646):eabn1252, 2022.

Appendix A

Inverse Problem Formulation

A.1 Justification for Frame-Based Distance Approach

Suppose we have a protein's three-dimensional C-α coordinates $\mathbf{x}$. It can be converted into a set of frames $\{(\mathbf{R}_i, \mathbf{T}_i)\}_{i=1}^{L}$ via Algorithm A.1. Note that the positions for the N and C atoms are fixed given the rigid body assumption.

Algorithm A.1: Frame Construction from Atom Coordinates (adapted from Supplementary Material Algorithm 21 of Jumper *et al.* [16])

```
input  : coordinates of $i$th N, C-$\alpha$, and C atoms $\mathbf{x}_{i,N}, \mathbf{x}_{i,CA}, \mathbf{x}_{i,C}$
output : frame representation of $i$th residue $(\mathbf{R}_i, \mathbf{T}_i)$
# Get vectors pointing from C-$\alpha$ to N and C
Set $\mathbf{v}_{i,1} \leftarrow \mathbf{x}_{i,C} - \mathbf{x}_{i,CA}$
Set $\mathbf{v}_{i,2} \leftarrow \mathbf{x}_{i,N} - \mathbf{x}_{i,CA}$

# Do Gram-Schmidt process
Set $\mathbf{e}_{i,1} \leftarrow \mathbf{v}_{i,1} / ||\mathbf{v}_{i,1}||$
Set $\mathbf{u}_{i,2} \leftarrow \mathbf{v}_{i,2} - \mathbf{e}_{i,1}\left(\mathbf{e}_{i,1}^{\top}\mathbf{v}_{i,2}\right)$
Set $\mathbf{e}_{i,2} \leftarrow \mathbf{u}_{i,2} / ||\mathbf{u}_{i,2}||$
Set $\mathbf{e}_{i,3} \leftarrow \mathbf{e}_{i,1} \times \mathbf{e}_{i,2}$

# Construct frame components
Set $\mathbf{R}_i \leftarrow \left(\mathbf{e}_{i,1} \,|\, \mathbf{e}_{i,2} \,|\, \mathbf{e}_{i,3}\right)$
Set $\mathbf{T}_i \leftarrow \mathbf{x}_{i,CA}$
```

We consider the case of a reflected protein $\mathbf{x}_{\text{ref}} \coloneqq -\mathbf{x}$ and a rotated protein $\mathbf{x}_{\text{rot}} \coloneqq \mathbf{R}_\theta \mathbf{x}$. Passing these to the algorithm we find that

$$\mathbf{x}_{\text{ref},i} \mapsto \left(\mathbf{R}_i \left(\begin{smallmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right),\ -\mathbf{T}_i\right), \quad \mathbf{x}_{\text{rot},i} \mapsto \left(\mathbf{R}_\theta \mathbf{R}_i,\ \mathbf{R}_\theta \mathbf{T}_i\right).$$

For a pair of residues (i, j), we then have

$$
\begin{aligned}
\mathbf{R}(\mathbf{x}_{\mathrm{ref},i})^\top \mathbf{R}(\mathbf{x}_{\mathrm{ref},j}) &= \left(\mathbf{R}_i \left(\begin{smallmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)\right)^\top \mathbf{R}_j \left(\begin{smallmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right) \mathbf{R}_i^\top \mathbf{R}_j \left(\begin{smallmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right), \\
\mathbf{R}(\mathbf{x}_{\mathrm{ref},i})^\top \mathbf{R}(\mathbf{x}_{\mathrm{ref},j}) &= (\mathbf{R}_\theta \mathbf{R}_i)^\top \mathbf{R}_\theta \mathbf{R}_j = \mathbf{R}_i^\top \mathbf{R}_\theta^\top \mathbf{R}_\theta \mathbf{R}_j = \mathbf{R}_i^\top \mathbf{R}_j,
\end{aligned}
$$

which shows $\mathbf{R}_i^\top \mathbf{R}_j$ is rotation invariant but not generally reflection invariant. We remark that this expression is different from $\mathbf{R}_i \mathbf{R}_j^\top$, the rotation matrix that transforms the jth frame's orientation to that of the ith frame, that we use in finding the cosine of the angle in between the two matrices. Moreover, we keep the angle in its cosine form and do not compute its arccosine due to its instability with gradients.

A.2 Other Motif Placement Parameterisations

Here, we consider other motif placement parameterisations. Instead of the likelihood, we can also directly parameterise over the posterior

$$
p(\mathbf{x} \mid \mathbf{y}) = \sum_{\mathscr{M} \in \mathbb{M}} p(\mathbf{x} \mid \mathbf{y}, \mathscr{M}) p(\mathscr{M} \mid \mathbf{y}) \propto \sum_{\mathscr{M} \in \mathbb{M}} p(\mathbf{x} \mid \mathbf{y}, \mathscr{M}) p(\mathbf{y}, \mathscr{M}),
$$

and, for each motif placement, sample from $p(\mathbf{x} \mid \mathbf{y}, \mathscr{M})$ while estimating the normalising constant $p(\mathbf{y}, \mathscr{M})$. This side-steps the uniform prior assumption but introduces significant computations, as we need to sample from $|\mathbb{M}|$ different posteriors. A possibility is to choose a sufficiently small subset of $\mathbb{M}$ created by randomly selecting motif placements.

Another reason for changing the original likelihood parameterisation is that it favours proteins containing several copies of the motif. Clearly, each motif copy contributes to the sum of the likelihood value. As such, naively optimising for this quantity can lead to undesired proteins. To solve this, we propose to generalise the posterior as

$$
p\left(\mathbf{x} \mid \bigcup_{\mathscr{M} \in \mathbb{M}} \{\mathbf{y} = \mathscr{A}_{\mathscr{M}}(\mathbf{x})\}\right).
$$

In doing so, we avoid explicit assumptions on the distribution of $\mathscr{M}$. Note that we precisely retrieve the fixed case when $|\mathbb{M}| = 1$. The likelihood can then be expressed as

$$
\begin{aligned}
p\left(\bigcup_{\mathscr{M} \in \mathbb{M}} \{\mathbf{y}\mathscr{A}_{\mathscr{M}}(\mathbf{x})\} \mid \mathbf{x}\right) := \sum_{\mathscr{M} \in \mathbb{M}} p\left(\mathbf{y} = \mathscr{A}_{\mathscr{M}}(\mathbf{x}) \mid \mathbf{x}\right) - \sum_{\mathscr{M}_1 < \mathscr{M}_2 \in \mathbb{M}} p\left(\bigcap_{i=1}^{2} \{\mathbf{y} = \mathscr{A}_{\mathscr{M}_i}(\mathbf{x})\} \mid \mathbf{x}\right) \\
+ \ldots + (-1)^{|\mathbb{M}|-1} \sum_{\mathscr{M}_1 < \ldots < \mathscr{M}_{|\mathbb{M}|} \in \mathbb{M}} p\left(\bigcap_{i=1}^{|\mathbb{M}|} \{\mathbf{y} = \mathscr{A}_{\mathscr{M}_i}(\mathbf{x})\} \mid \mathbf{x}\right).
\end{aligned}
$$

Here, joint densities involving intersecting motif placements are set to zero. Those which are not are expressed as a product of the marginal densities. In practice, however, this computation is rather expensive. Apart from there being a large number of terms, the densities are extremely small and require being handled in their log-form. Even if an approximation is made to truncate the series after the second summation, `logsumexp` calculations in the complex domain are necessary to deal with subtractions.

A.3 Gradient of Log Likelihood Computation

Throughout, we compute $\nabla_{x_t} \log g(y_0 \mid \hat{x}_0(x_t, t))$ entirely via automatic differentiation. However, in cases where we want to minimise the storage of gradients in memory after each operation, we may also derive an analytical expression involving the gradient of $\mathscr{A}$, assuming it is known, and the gradient of the de-noising network ϵ_θ. We have, by the chain rule,

$$\begin{aligned}
&\nabla_{x_t} \log g(y_0 \mid \hat{x}_0(x_t, t)) \\
&= \nabla_x \log g(y_0 \mid x)|_{x=\hat{x}_0(x_t,t)} \cdot \nabla_{x_t} \hat{x}_0(x_t, t) \\
&= \nabla_x \left(-\frac{1}{2\sigma^2}(y_0 - \mathscr{A}(x))^2 \right)|_{x=\hat{x}_0(x_t,t)} \cdot \nabla_{x_t} \left(\frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t) \right) \right) \\
&= -\frac{1}{\sigma^2} \left(y_0 - \nabla_x \mathscr{A}(x)|_{x=\hat{x}_0(x_t,t)} \right) \cdot \frac{1}{\sqrt{\bar{\alpha}_t}} \left(1 - \sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \epsilon_\theta(x_t, t) \right),
\end{aligned}$$

where $\nabla_{x_t} \epsilon_\theta(x_t, t)$ is computed via backpropagation.

Appendix B

Experimental Setup

B.1 Benchmark Problems

B.1.1 Motif Scaffolding Benchmark

Problems in the RFDiffusion motif scaffolding benchmark are listed in Table B.1. Excluding 6VW1, there are 24 problems involving different motif types, lengths, and contiguity. Where the length and configuration define a range, we can design any length scaffold that fits those specifications but choose to fix the configuration in our experiments.

B.1.2 Multi-Motif Scaffolding Benchmark

Problems in the Genie2 multi-motif scaffolding benchmark are listed in Table B.2. In the Genie2 pre-print, problem 3NTN had a configuration with ranges in reverse. However, their actual specification was different in their GitHub repository. We chose to work with the ranges specified in their repository and made the correction in Table B.2.

B.1.3 Symmetric Motif Scaffolding

In addition to producing various symmetric monomers, we attempted to replicate RFDiffusion's design of a C_3-symmetric multivalent binder to the SARS-CoV-2 spike protein, containing three copies of ACE2 mimic AHB2 [28] as the motif. However, as we worked with a diffusion model capable of modelling a single chain only, we attempted to scaffold the three motifs with a single monomer. The multivalent binder designs from RFDiffusion were C_3-symmetric trimers with 615 residues in total length. The motif present in each monomer is the first 55 residues of the binder against the covid spike protein. We matched RFDiffusion's total protein length of 615 residues by using Genie-Scope-256 despite this being an out-of-distribution task.

Name	Description	Configuration	Length
1PRW	Double EF-hand motif	5-20, **A16-35**, 10-25, **A52-71**, 5-20	60-105
1BCF	Di-iron binding motif	8-15, **A92-99**, 16-30, **A123-130**, 16-30, **A47-54**, 16-30, **A18-25**, 8-15	96-152
5TPN	RSV F-protein Site V	10-40, **A163-181**, 10-40	50-75
5IUS	PD-L1 binding interface on PD-1	0-30, **A119-140**, 15-40, **A63-82**, 0-30	57-142
3IXT	RSV F-protein Site II	10-40, **P254-277**, 10-40	50-75
5YUI	Carbonic anhydrase active site	5-30, A**93-97**, 5-20, **A118-120**, 10-35, **A198-200**, 10-30	50-100
1QJG	Delta5-3-ketosteroid isomerase active site	10-20, **A38**, 15-30, **A14**, 15-30, **A99**, 10-20	53-103
1YCR	P53 helix that binds to Mdm2	10-40, **B19-27**, 10-40	40-100
2KL8	De novo designed protein	**A1-7**, 20, **A28-79**	79
7MRX_60	Barnase ribonuclease inhibitor	0-38, **B25-46**, 0-38	60
7MRX_85	Barnase ribonuclease inhibitor	0-68, **B25-46**, 0-63	85
7MRX_128	Barnase ribonuclease inhibitor	0-122, **B25-46**, 0-122	128
4JHW	RSV F-protein Site 0	10-25, **F196-212**, 15-30, **F63-69**, 10-25	60-90
4ZYP	RSV F-protein Site 4	10-40, **A422-436**, 10-40	30-50
5WN9	RSV G-protein 2D10 site	10-40, **A170-189**, 10-40	35-50
5TRV_short	De novo designed protein	0-35, **A45-65**, 0-35	56
5TRV_med	De novo designed protein	0-65, **A45-65**, 0-65	86
5TRV_long	De novo designed protein	0-95, **A45-65**, 0-95	116
6E6R_short	Ferridoxin Protein	0-35, **A23-35**, 0-35	48
6E6R_med	Ferridoxin Protein	0-65, **A23-35**, 0-65	78
6E6R_long	Ferridoxin Protein	0-95, **A23-35**, 0-95	108
6EXZ_short	RNA export factor	0-35, **A28-42**, 0-35	50
6EXZ_med	RNA export factor	0-65, **A28-42**, 0-65	80
6EXZ_long	RNA export factor	0-95, **A28-42**, 0-95	110

Table B.1: RFDiffusion motif scaffolding benchmark details. The specification for each scaffolding problem is under "Configuration", with the motif structures in bold. For example, in motif 2KL8, the problem requires a protein that contains residues from chain A of the motif at the motif's residues 1-7 and 28-79, joined together by a scaffold of 20 residues. Furthermore, the total length of the generated protein must fall in the range specified in the "Length" column.

Name	Description	Configuration	Length
4JHW+5WN9	Two epitopes	10-40, **4JHW/F254-278{1}**, 20-50, **5WN9/A170-189{2}**, 10-40	85-175
1PRW_two	Two 4-helix bundles	5-20, **1PRW/A16-35{1}**, 10-25, **1PRW/A52-71{1}**, 10-30, **1PRW/A89-108{2}**, 10-25, **1PRW/A125-144{2}**, 5-20	120-200
1PRW_four	Four EF-hands	5-20, **1PRW/A21-32{1}**, 10-25, **1PRW/A57-68{2}**, 10-25, **1PRW/A94-105{3}**, 10-25, **1PRW/A125-144{4}**, 5-20	88-163
3BIK+3BP5	Two PD-1 binding motifs	5-15, **3BIK/A121-125{1}**, 10-20, **3BP5/B110-114{2}**, 5-15	30-60
3NTN	Two 3-helix bundles	**3NTN/A342-348{1}**, 10, **3NTN/A367-372{2}**, 10-20, **3NTN/B342-348{2}**, 10, **3NTN/B367-372{1}**, 10-20, **3NTN/C367-372{1}**, 10, **3NTN/C342-348{2}**	89-109
2B5I	Two binding sites	5-15, **2B5I/A11-23{2}**, 10-20, **2B5I/A35-45{1}**, 10-20, **2B5I/A61-72{1}**, 5-15, **2B5I/A81-95{2}**, 20-30, **2B5I/A119-133{2}**	116-166

Table B.2: Genie2 multi-motif scaffolding benchmark details. The specification for each scaffolding problem is under "Configuration", with the motif structures in bold. Here, the motif structures are formatted as `<MOTIF_NAME>/<CHAIN_SEGMENT>{<MOTIF_GROUP>}`, where structures belonging to the same motif group are fixed in their orientations relative to each other. Furthermore, the total length of the generated protein must fall in the range specified in the "Length" column.

B.2 Sampler Optimisations

In our implementation of samplers, we make several optimisations. By partitioning particles into separate groups, we could run several trials simultaneously, making the most of throughput gains with increased model batch sizes. This is achieved by computing weights of particles relative to those in their group and similarly resampling on a per-group basis. We further perform multiprocessing across several GPUs.

Beyond parallelism, we minimise the number of calls made to the diffusion model. We cache the predicted noise (or score) between computing weights and sampling from the proposal. Additionally, as resampling often yields duplicate particles, we only compute the predicted noise of unique particles. Then, we apply different batches of Gaussian noise to differentiate duplicate particles from each other. We remark, however, that the latter does not apply to latent projection methods, as the weights involve the predicted noise and need to be computed for all particles.

Appendix C

Additional Results

Here, we document additional results to the main and supporting experiments.

C.1 Motif Scaffolding

C.1.1 Hyperparameter Search and Ablation Study

The hyperparameters of methods were chosen to maximise performance on two selected motifs: 1PRW and 3IXT, in a discretised grid-search fashion with 16 samples for each parameter combination. We considered values $\zeta = 0.4, 0.7, 1.0$ for the temperature value. We found a value of $\zeta = 0.4$ to have produced non-zero successes in the observation-projection methods. We also accounted for the guidance scale γ in TDS as shown in Figure C.1. TDS-MASK achieved the best unique success rate at two combinations but had a lower average scRMSD at $\gamma = 1.0$ and $\zeta = 1.0$.

On the other hand, higher guidance scales had a negative impact on TDS-DIST. It began to optimise the likelihood without considering the handedness of the generated structures, producing a reflected motif with left-handed helices up to 50% of the time. By reducing the guidance scale, the unconditional model makes a bigger contribution to the de-noising process and is less likely to make such mistakes. TDS-DIST is thus configured at $\gamma = 0.25$ and $\zeta = 0.4$.

With TDS-FRAME-DIST, an additional parameter η is available to scale the contribution of the rotation deviations to the likelihood. When $\eta = 0$, we retrieve back TDS-DIST. Due to the large parameter space, we limit our search to a fixed temperature value of $\zeta = 0.4$ to match TDS-DIST. As shown in Figure C.2, a non-zero rotation scale indeed corrects for reflections from as little as $\eta = 1.0$. However, as with 1PRW, there is a range of values between 1.0 and 8.0 where the rotation contribution is too weak, hurting the motif RMSD as it attempts to steer the trajectory away from solutions that meet the distance

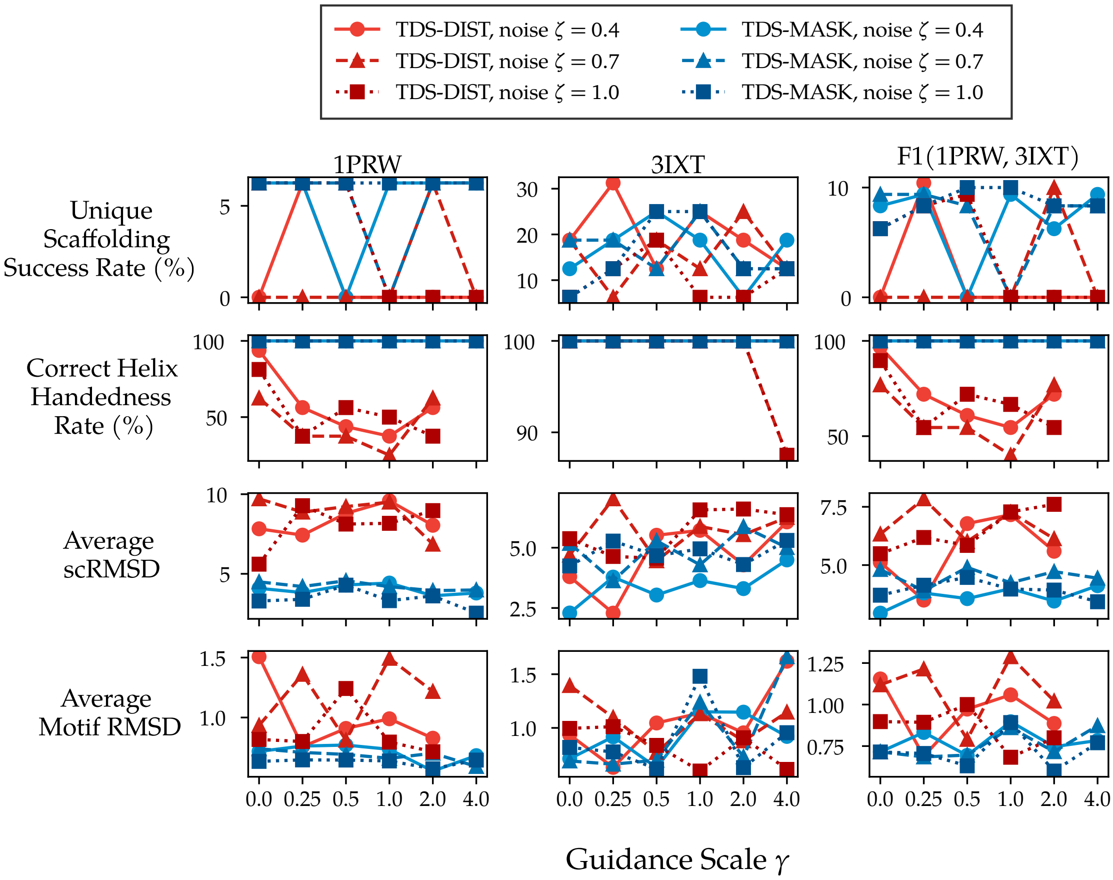

Figure C.1: Grid search for TDS-DIST and TDS-MASK. The rates at which unique success and right-handed helices were achieved are reported. Average values for two of the four designability criteria are also shown.

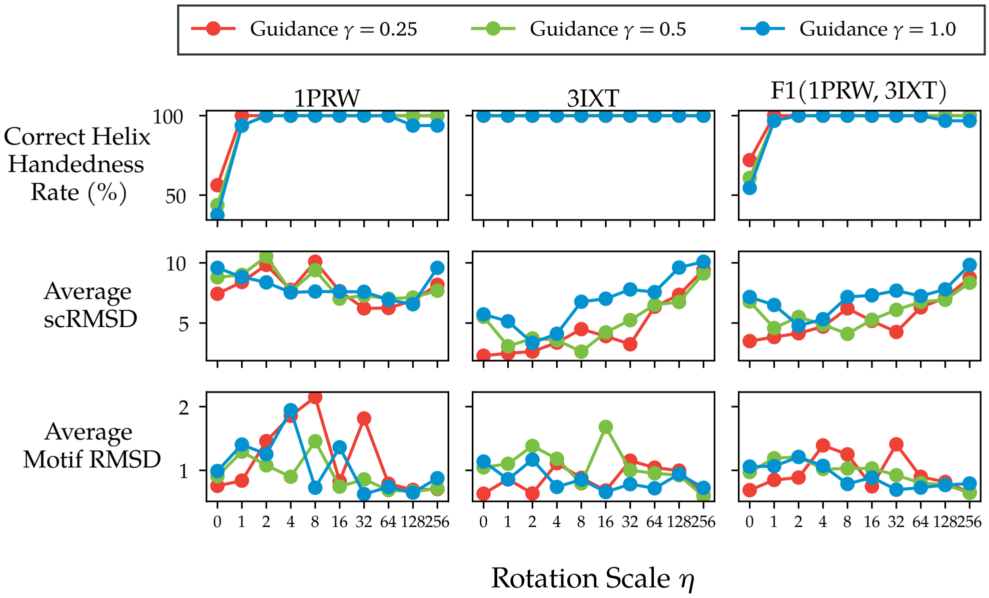

Figure C.2: Effect of TDS-FRAME-DIST's rotation scale η on the handedness and designability of generated structures. The right column depicts the F1 score between the values in the left and middle columns. A noise scale value of $\zeta = 0.4$ is used throughout.

constraints to those matching the right orientation. When the scale is too large, solutions have the correct orientations but incorrect distances, leading to malformed backbones. Here, the scRMSD is at its highest. We find that a value of $\eta = 64.0$ balances this trade-off but remark that the optimal value differs in the case of both motifs.

The varied optimal parameter values across motif problems show a potential weakness of these methods. For example, we believe the optimal scale value varies with the motif's size in space. To overcome these, more work is needed to explore non-static parameter values.

C.1.2 Further Breakdown of Results

Figure C.3 provides a finer breakdown of the performance of various samplers according to their average values in all four success criteria. Here, we can roughly measure the difficulty of motif problems according to their scaffolds' average scRMSD and motif RMSD. Problems such as 1BCF, 4JHW, 5IUS, and 5YUI have the highest average deviation from the success thresholds. In fact, problem 4JHW has yet to be solved through any method, conditional training or not, and very few successes have been reported for the others. Another observation is that the 6E6R and 6EXZ problems appear to be about the same difficulty. As successes were found in the 6EXZ problems, 6E6R problems can likely be solved by the current setup with more samples.

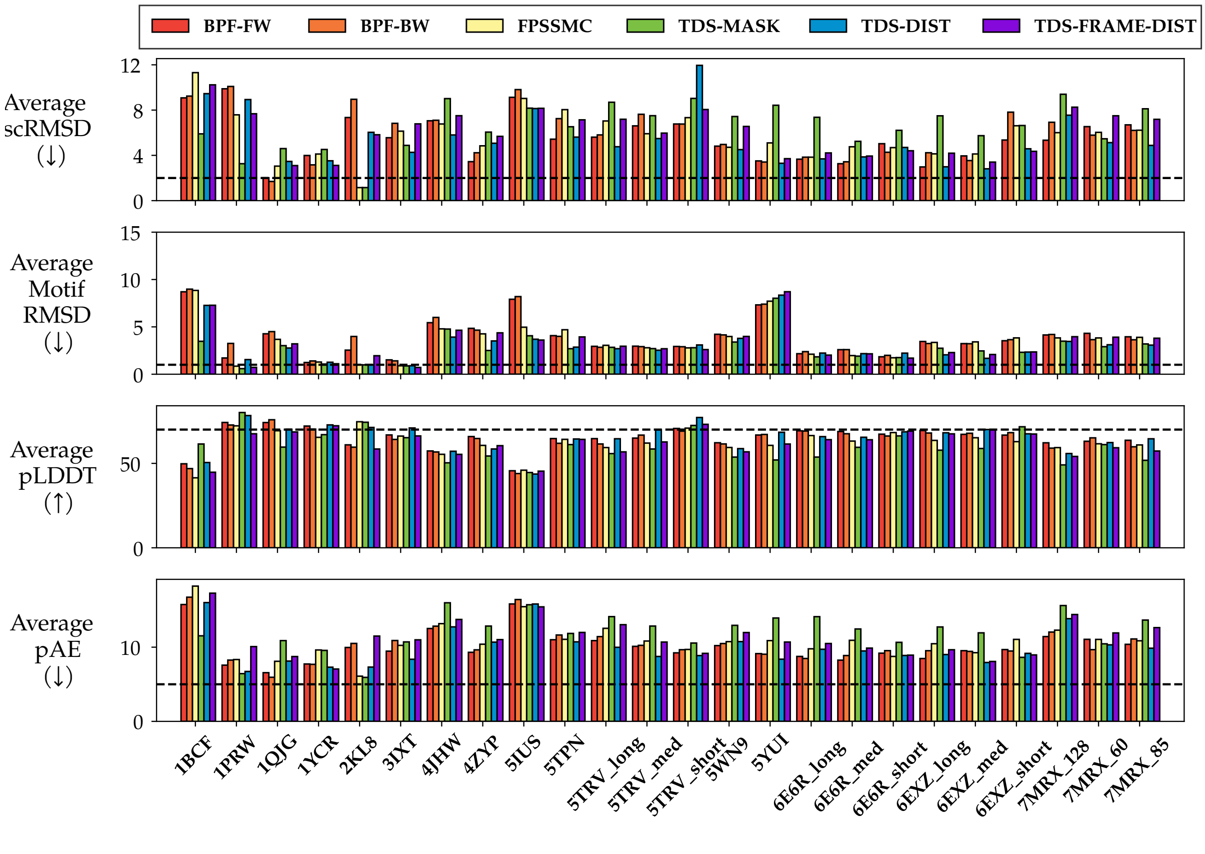

Figure C.3: Motif scaffolding benchmark results according to average values of success criteria. Values for a pass in each criteria are denoted by the dashed line.

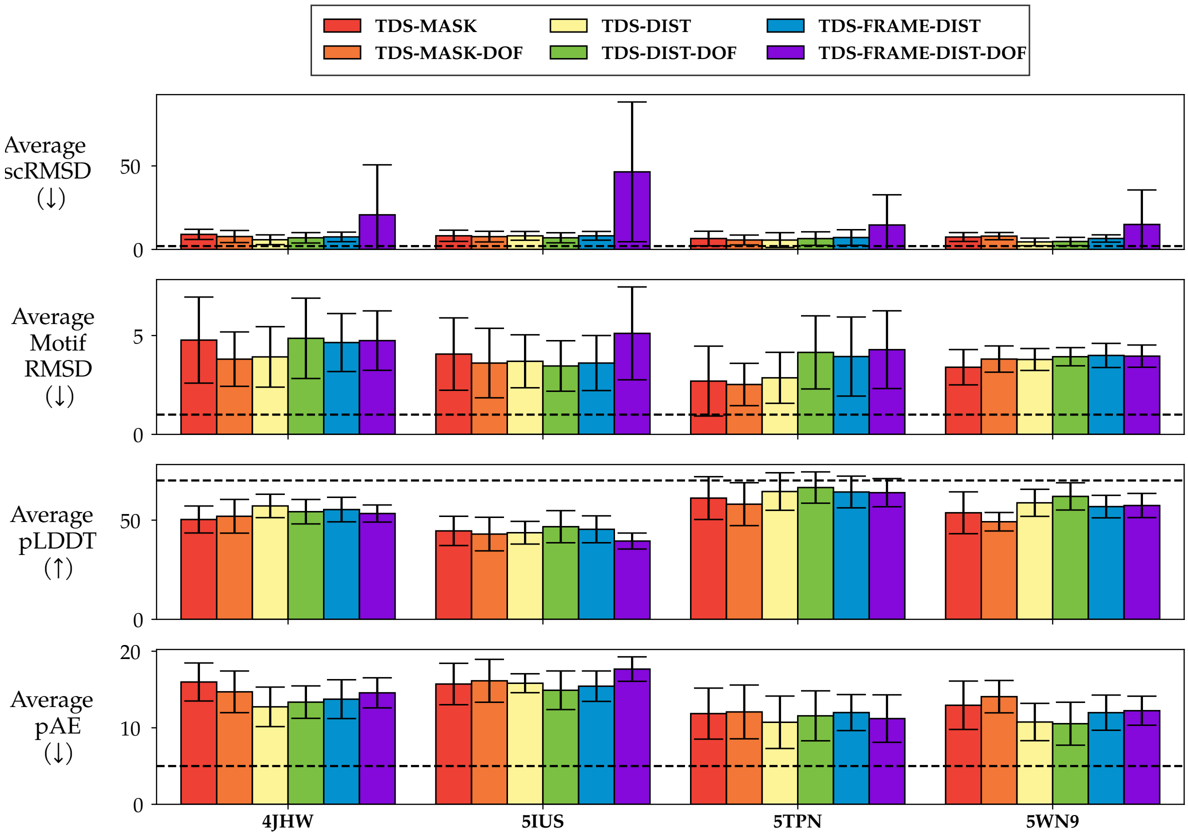

Figure C.4: Average values for success metrics when compounding the latent projection methods with the mixture likelihood to perform scaffolding with degrees of freedom. Values for a pass in each criterion are denoted by the dashed line. Error bars indicate one standard deviation from the mean.

C.1.3 Variable Motif Placement for Unsolved Problems

We parameterised the likelihood as a mixture of several likelihoods corresponding to all the possible motif placements as in Equation 4.2. The samplers optimised for the motif to be in at least one of these placements. Since we generally do not know where this may be, we chose the placement corresponding to the maximised likelihood component.

As in Figure C.4, we found varying results for a select four unsolved motif problems. An immediate observation is that the average scRMSD increases significantly for TDS-FRAME-DIST. Upon inspection, these were cases where the motif was present but was located considerably far away from the rest of the scaffold. One possible explanation is its progress for meeting the distance constraints was already slow, but further compounding several possible placements made it unable to commit to one until much later in the de-noising process when the rest of the protein's shape had already formed, effectively

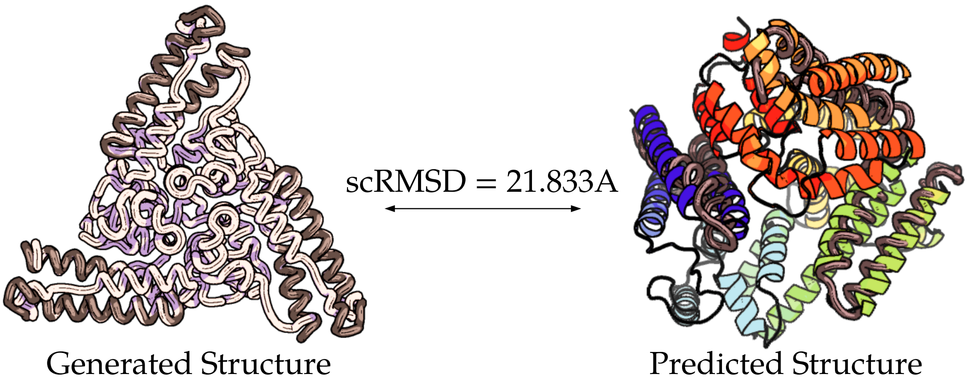

Figure C.5: An example of a C_3 symmetric scaffold containing three binder motifs and its self-consistency predicted structure. The motif, coloured grey, is aligned with the structures. Large scRMSDs are observed throughout all the samples.

forcing the motif at some location. And, as the rotation matrix's contribution to the gradient is unclear, it is possible that it fixes one of the residues and naively orients the motif with respect to it. While speculative, the absence of this spike in the TDS-DIST case points towards the rotation matrix conditioning as the reason.

The remaining methods' added conditioning improves performance on some problems but not others. Notably, TDS-MASK-DOF achieved one successful scaffold for problem 5TPN.

C.2 Symmetric Motif Scaffolding

Tasked with generating C_3 symmetric monomers scaffolding three binder motifs, FPSSMC and TDS-MASK could not produce designable scaffolds. All scaffolds had scRMSD $> 16A$. An example is given in Figure C.5. This is likely due to modelling the binder as a monomer instead of a trimer and our usage of Genie-Scope-256 in an out-of-distribution task of generating proteins as large as 600 residues. RFDiffusion, capable of modelling several chains and producing long proteins, succeeds at this task by explicitly symmetrising the protein as it is being de-noised.