Part I

Appendix

Table of Contents

A ADDITIONAL RELATED WORK

We next cover additional related work on generative models of proteins sequence and structure, beyond the discussion in Section 1.1. Following the success of deep language models, Ferruz et al. (2022) developed protein sequence models to generate new proteins, but these models do not allow specification of structural motifs. Another class of methods, referred to as fixed backbone sequence design (Fleishman et al., 2011; Ingraham et al., 2019; Xiong et al., 2020; McPartlon et al., 2022; Hsu et al., 2022), attempts to solve the problem of identifying a sequence that folds into any given designable backbone structure. In the present work, we utilize a particular sequence design method, `ProteinMPNN` (Dauparas et al., 2022), but in principle any other fixed-backbone sequence design method could be used in its place. Anand & Huang (2018); Lin et al. (2021); Wu et al. (2021) propose generative adversarial networks, variational autoencoders, and energy-based models, respectively, on distance matrices, but these approaches (1) do not generate backbones compatible with a specified motif and (2) rely on an unwieldy optimization step to translate the distance matrix into backbone coordinates. Other authors use neural net (Tischer et al., 2020; Anishchenko et al., 2021; Wang et al., 2022; Huang et al., 2022; Wu et al., 2021), but require a computationally challenging conformational landscape exploration.

B PROBLEM ASSUMPTIONS AND MODELING HEURISTICS

The formulation of the motif-scaffolding problem presented in Section 2.1 makes several simplifying assumptions, and our modeling approach relies on several heuristics. We describe these assumptions and heuristics in what follows, and comment on how they might be addressed by further methodological developments. But we first describe an illustrative example of an instance of motif scaffolding.

Protein sequence-structure relationship. Generally speaking, a protein's sequence encodes an ensemble of conformations, populated to different degrees at biological temperatures. Anfinsen's hypothesis states that the ground state conformation is thermodynamically accessible (Anfinsen, 1973), providing a mapping from sequence to a unique (ground state) structure. In practice, the ground state structures make up the vast majority of experimentally determined protein conformations, as over 95% of structures in the Protein Data Bank (PDB) are collected at cryogenic temperatures (Fraser et al., 2011). Thus we simplify our problem by saying that a sequence uniquely maps to a static structure (i.e. the ground state structure). However, violations of this assumption arise in some PDB structures as a result of (1) of context specific determinants of structure such as post-translational modifications and environmental factors including pH, binding partners, and salts, as well as (2) thermodynamic inaccessibility of the ground state.

Motif sequence and side-chains. As stated in Section 2.1, we assume we may represent a functional motif by the coordinates of its C-α atoms. However, the biochemical functions of proteins depend not only on backbone structure, but also on side-chains. For example, the activity of many enzymes is imparted by triplets of residues, known as *catalytic triads*, whose ability to catalyze reactions depends on the spatial organization of side-chain atoms. Our problem statement and subsequent evaluation scheme are agnostic to the amino acid identity of motif residues, let alone side-chain positioning. A more complete representation of a motif would include the side-chain identities (i.e. the amino acid *sequence*) and side-chain atom coordinates.

Scaffold length and motif placement. We have additionally assumed that the size of scaffolds and the indices of motif residues within the backbone chain, $\mathcal{M}$, are known a priori. However, in practice satisfactory scaffolds could have different lengths and different motif placements, and typically it is not known a priori what lengths and placements will be best. Previous works have addressed this challenge through brute force by sampling multiple lengths and placements, and relied on post-hoc filtering to identify the most promising scaffolds (Wang et al., 2022; Yang et al., 2021). Subsequent work on ML methods could potentially generalize beyond this assumption to efficiently sample appropriate scaffold lengths and motif placements.

Sequence and side-chain modeling. `ProtDiff` models only the backbone coordinates and leaves sequence design to a subsequent stage, for which we have used `ProteinMPNN`. A more complete representation of a proteins could include both sequence and structure (where structure can be divided into the backbone and side-chain atom coordinates). To model sequence, we rely on a separately trained neural network, `ProteinMPNN`, but this is not ideal. Unless `ProtDiff` produces perfect backbones, one would expect the backbone samples of `ProtDiff` to present a substantial domain shift when used as input for `ProteinMPNN`.

3D backbone representation. In this work, we represent a protein structure using the C-α coordinates of every residue along the backbone. However, this representation is coarse-grained and ignores additional backbone atomic coordinates, namely the backbone carbon and nitrogen atoms. Dauparas et al. (2022) observed additionally modeling the heavy atoms of the backbone nitrogen and carbon atoms along with the C-β of every residue (to capture side-chain information) improved performance (by sequence recovery) for fixed-backbone sequence design. We hypothesize modeling additional coordinates of every residue would also improve designability performance of `ProtDiff`. Constraining `ProtDiff` to place the remaining atoms in the correct orientation could help enforce correct chirality and mitigate chain breaks.

C ADDITIONAL `ProtDiff` DETAILS

As a reminder from Section 3, each node in the graph is indexed by $n = 1, \ldots, N$ and corresponds to a residue with coordinates $\mathbf{x}_n \in \mathbb{R}^3$ and node features $h_n \in \mathbb{R}^D$. For each pair of nodes n, n' we define an edge and associate it with edge features $a_{nn'} \in \mathbb{R}^D$. Our neural network to predict ϵ_θ is an instance of EGNN composed of multiple EGCL layers . We recount details of EGCL and then discuss construction of edge and node features, $a_{nn'}$ and h_n.

Equivariant graph convolution layers (EGCL). Each layer $l = 1, \dots, L$ defines an update as $(\mathbf{x}^l, h^l) = \text{EGCL}[\mathbf{x}^{l-1}, h^{l-1}]$ where for each node n

$$\mathbf{x}_n^l = \mathbf{x}_n^{l-1} + \sum_{n' \neq n} \vec{\omega}_{nn'} \cdot \phi_{\mathbf{x}}(h_n^{l-1}, h_{n'}^{l-1}, d_{nn'}, a_{nn'}) \quad \text{and} \quad h_n^l = \phi_h(h_n^{l-1}, m_n), \quad \text{for}$$

$$\vec{\omega}_{nn'} = \frac{\mathbf{x}_n^{l-1} - \mathbf{x}_{n'}^{l-1}}{\sqrt{d_{nn'}} + \gamma}, \quad m_n = \sum_{n' \neq n} \phi_e(h_n^{l-1}, h_{n'}^{l-1}, d_{nn'}, a_{nn'}), \text{ and } d_{nn'} = \|\mathbf{x}_n^{l-1} - \mathbf{x}_{n'}^{l-1}\|_2^2.$$

ϕ_e, ϕ_h, and $\phi_{\mathbf{x}}$ are fully connected neural networks, and γ is a small positive constant included for numerical stability. The first EGCL layer takes in initial node embeddings, h^0 while edge embeddings, $a_{nn'}$, are kept fixed throughout.

Initial node and edge embeddings. Each edge between two residues indexed in the sequence by (n, n') is featurized with D features obtained through a sinusoidal encoding of its relative offset:

$$a_{nn'} = \begin{bmatrix} \varphi(n - n', 1) \\ \vdots \\ \varphi(n - n', D) \end{bmatrix}, \text{ where } \varphi(x, k) = \begin{cases} \sin\left(x \cdot \pi / N^{2 \cdot k / D}\right), & k \mod 2 = 0 \\ \cos\left(x \cdot \pi / N^{2 \cdot (k-1)/D}\right), & k \mod 2 = 1. \end{cases}$$

For node features, we similarly use a sinusoidal encoding of sequence position as well as of the diffusion time step t as

$$h_n(t) = \begin{bmatrix} \varphi(n, 1) \\ \vdots \\ \varphi(n, D) \end{bmatrix} + R \begin{bmatrix} \varphi(t, 1) \\ \vdots \\ \varphi(t, D) \end{bmatrix},$$

where R is a $D \times D$ orthogonal matrix chosen uniformly at random. Intuitively, applying R transforms the time encoding to be orthogonal to the positional encoding.

Coordinate scaling While protein structures are typically parameterized in Angstroms, we transform the input protein coordinates to be in nanometers rather by dividing by 10. This scaling brings the backbones to a spatial scale similar to the reference distribution at which the forward noising process is stationary, a unit variance isotropic Gaussian. Importantly, the distribution of the final step T is indistinguishable from an isotropic Gaussian (Supplementary Fig. 5.)

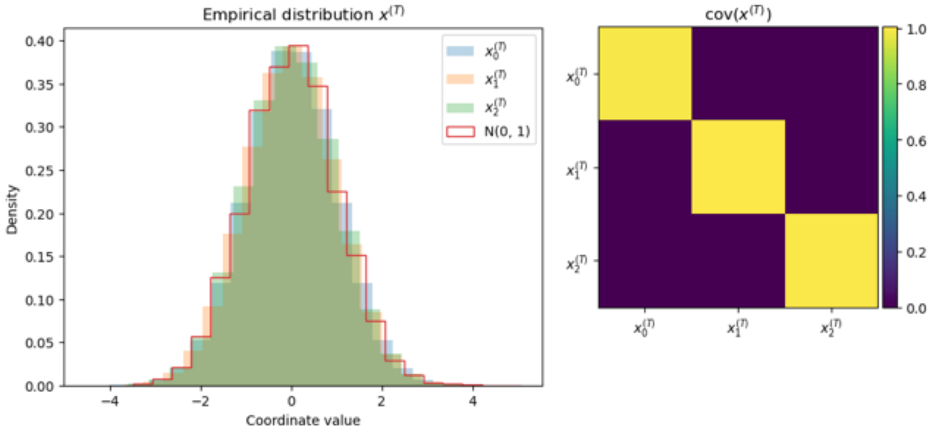


Figure 5: Distribution of $x^{(T)}$ after centering and scaling $x^{(0)}$ to nanometers.

D CONDITIONAL SAMPLING: `SMCDiff` DETAILS AND SUPPLEMENTARY PROOFS

We here provide additional details related to `SMCDiff` and the replacement method described in Section 4. Details of the replacement method (Song et al., 2021) and our analysis of its error are in

Algorithm 2 Replacement method for approximate conditional sampling

```
**Input:** $\mathbf{x}_{\mathcal{M}}^{(0)}$ (motif)
// Forward diffuse motif
$\breve{\mathbf{x}}_{\mathcal{M}}^{(1:T)} \sim q(\mathbf{x}_{\mathcal{M}}^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)})$

// Reverse diffuse scaffold
$\mathbf{x}^{(T)} \sim p_\theta(\mathbf{x}^{(T)})$
**for** $t = T, \dots, 1$ **do**
    // *Replace* with forward diffused motif
    $\mathbf{x}^{(t)} \leftarrow [\breve{\mathbf{x}}_{\mathcal{M}}^{(t)}, \mathbf{x}_{\mathcal{S}}^{(t)}]$

    // Propose next step
    $\mathbf{x}^{(t-1)} \sim p_\theta(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)})$
**end for**
Return $\mathbf{x}_{\mathcal{S}}^{(0)}$, $\mathbf{x}^{(1:T)}$
```

Appendix D.1. Appendix D.2 provides details of our sampling method, `SMCDiff`, including (1) a proof of Proposition 4.1 and (2) details of the residual resampling step. We leave technical proofs and lemmas to Appendix D.3.

Notation. In the following, we require notation that is more precise than in previous sections. For each $t = 0, \dots, T$, we let $q_t(\cdot)$ and $p_t(\cdot)$ denote the density functions of $\mathbf{x}^{(t)}$ according to the forward process and to our neural network approximation of the reverse process, respectively. We denote densities restricted to the motif and scaffold with subscripts $\mathcal{M}$ and $\mathcal{S}$. For example, we here write $p_{\mathcal{M},t}(\mathbf{x}_{\mathcal{M}}^{(t)})$, whereas we wrote $p_\theta(\mathbf{x}_{\mathcal{M}}^{(t)})$ in the main text. We write (random) conditional densities as $q_{\mathcal{M},t}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(t-1)})$ and write the (deterministic) conditional density for an observation $\mathbf{x}_{\mathcal{M}}^{(t-1)} = x_{\mathcal{M}}$ as $q_{\mathcal{M},t}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(t-1)} = x_{\mathcal{M}})$.

An object of interest will be the Kullback-Leibler (KL) divergence. We write $\mathrm{KL}\left[q_t(\cdot) \| p_t(\cdot)\right] := \int q_t(x) \log \frac{q_t(x)}{p_t(x)} dx$, where $\log(\cdot)$ is the natural (base e) logarithm. We will also encounter the expected KL between conditional densities, which we will write as $\mathrm{EKL}\left[q_t(\cdot \mid \mathbf{x}^{(t-1)}) \| p_t(\cdot \mid \mathbf{x}^{(t-1)})\right] := \int q_{t-1}(x) \mathrm{KL}\left[q_t(\cdot \mid \mathbf{x}^{(t-1)} = x) \| p_t(\cdot \mid \mathbf{x}^{(t-1)} = x)\right] dx$, where the outer expectation is taken with respect to the unconditional density associated with first argument of $\mathrm{EKL}\left[\cdot \| \cdot\right]$.

D.1 THE REPLACEMENT METHOD AND ITS ERROR

The replacement method was proposed by Song et al. (2021) for the task of inpainting in the context of score-based generative models. Work (Ho et al., 2022) concurrent with the present paper applied the replacement method to DPMs. Although Song et al. (2021) notes that this approach can be understood as *approximate* conditional sampling, they provide no discussion of approximation error. We here show that the replacement method introduces irreducible error that is inherent to the forward process. Algorithm 2 provides an explicit description of the replacement method.

The first return of Algorithm 2, $\mathbf{x}_{\mathcal{S}}^{(0)}$, is used as a putative inpainting solution or approximate conditional sample. But Algorithm 2 additionally returns subsequent time steps, $\mathbf{x}^{(1:T)}$. We denote the approximation over all steps implied by the generative procedure in Algorithm 2 by $p_{1:T}^{\mathrm{Repl}}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})$ and compare it to the exact conditional, $q_{1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})$. We here consider error in KL divergence because it permits an analytically tractable and transparent analysis. We additionally consider the idealized scenario where $p_{0:T}(\cdot)$ perfectly captures the reverse process. Under this condition, the forward KL takes a surprisingly simple form.

Proposition D.1. *Suppose that $p_{0:T}(\cdot)$ exactly matches the forward diffusion process such that for every x, $p_t(\cdot \mid \mathbf{x}^{(t+1)} = x) = q_t(\cdot \mid \mathbf{x}^{(t+1)} = x)$. Then for any motif $x_{\mathcal{M}}$,*

$$\begin{aligned} &\mathrm{KL}\left[q_{1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \| p_{1:T}^{\mathrm{Repl}}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})\right] \\ &= \sum_{t=1}^{T-1} \mathrm{EKL}\left[q_{\mathcal{S},t}(\cdot \mid \mathbf{x}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \| q_{\mathcal{S},t}(\cdot \mid \mathbf{x}^{(t+1)})\right]. \end{aligned} \tag{2}$$

Proposition D.1 reveals that the replacement method introduces approximation error that is intrinsic to the forward process and cannot be eliminated by making $p_{0:T}(\cdot)$ more expressive. Although the individual terms in the right hand side of Equation (2) are not analytically tractable in general, in the following corollary we show that this approximation error can be non-trivial by considering a special case. For this following example, we depart from the earlier assumption that $\mathbf{x}$ is in 3D, and consider scalar valued $\mathbf{x}_M$ and $\mathbf{x}_{\mathcal{S}}$.

Corollary D.2. *Suppose $[\mathbf{x}_{\mathcal{M}}^{(0)}, \mathbf{x}_{\mathcal{S}}^{(0)}]$ is bivariate normal distributed with mean zero, unit variance, and covariance ρ. Further suppose that $q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(0)}) = \mathcal{N}(\cdot; \sqrt{\bar{\alpha}^{(t)}}\mathbf{x}_{\mathcal{S}}^{(0)}, 1-\bar{\alpha}^{(t)})$ and $q_{\mathcal{S},t+1}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t)}) = \mathcal{N}(\cdot; \sqrt{1-\beta^{(t+1)}}\mathbf{x}_{\mathcal{S}}^{(t)}, \beta^{(t+1)})$ as in Section 2, where $\beta^{(t+1)}$ and $\bar{\alpha}^{(t)}$ are between 0 and 1. Then*

$$\mathrm{EKL}\left[q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}) \| q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)})\right] \geq -\frac{1}{2}\left(\log(1-\beta^{(t+1)}\bar{\alpha}^{(t)}\rho^2) + \beta^{(t+1)}\bar{\alpha}^{(t)}\rho^2\right).$$

We note two takeaways of Corollary D.2. First, as we might intuitively expect, this error can be large when significant correlation in the target distribution is present. Second, we see that the approximation error can be larger at earlier time steps, when $\bar{\alpha}^{(t)}$ is closer to 1.

D.2 `SMCDiff` DETAILS AND VERIFICATION PROOF OF PROPOSITION 4.1

The idea behind the `SMCDiff` procedure in Algorithm 1 is to break sampling of $\mathbf{x}_{\mathcal{S}}^{(0)} \sim q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)})$ into three stages:

1. Draw $\mathbf{x}_{\mathcal{M}}^{(1:T)} \sim q_{\mathcal{M},1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)})$.
2. Draw $\mathbf{x}_{\mathcal{S}}^{(1:T)} \sim q_{\mathcal{S},1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$.
3. Draw $\mathbf{x}_{\mathcal{S}}^{(0)} \sim q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)}, \mathbf{x}_{\mathcal{S}}^{(1:T)})$

If all three steps were performed exactly, by the law of total probability $\mathbf{x}_{\mathcal{S}}^{(0)}$ in step (3) would (marginally) be an exact sample from $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)})$. As such, `SMCDiff` aims to perform step (1) and approximate steps (2) and (3). Step (1) corresponds to forward diffusing the motif in lines 2–3 and is exact because we diffuse according to q.

Step (3) corresponds to line 17 in the last iteration (when $t = 1$). Specifically, to sample from $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)}, \mathbf{x}_{\mathcal{S}}^{(1:T)})$ we make three observations. (i) The Markov structure of the forward process implies that $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)}, \mathbf{x}_{\mathcal{S}}^{(1:T)}) = q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:1)}, \mathbf{x}_{\mathcal{S}}^{(1)})$. (ii) By the assumption that the forward and approximated reverse process agree, we have $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:1)}, \mathbf{x}_{\mathcal{S}}^{(1)}) = p_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:1)}, \mathbf{x}_{\mathcal{S}}^{(1)})$. (iii) Finally, because $p_t(\cdot \mid \mathbf{x}^{(t+1)})$ factorizes across $\mathcal{M}$ and $\mathcal{S}$ for each t, $p_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:1)}, \mathbf{x}_{\mathcal{S}}^{(1)}) = p_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(1)}, \mathbf{x}_{\mathcal{S}}^{(1)})$. As a result, under the assumptions of the proposition, we may sample from $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)}, \mathbf{x}_{\mathcal{S}}^{(1:T)})$, and perform step (3) exactly as well.

Step (2) is the only non-trivial step, and cannot be performed exactly. The challenge is that although the reverse process approximation, $p_{\mathcal{S},1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$, is well-defined, computing it explicitly involves an intractable, high-dimensional integral.

The sequential Monte Carlo approach of `SMCDiff`, then, is to circumvent this intractability by constructing a sequence of approximations. For each $t = T, T-1, \ldots, 1$, we approximate $p_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(t-1:T)})$ (and thereby $q_{\mathcal{S},t}(\cdot \mid \mathbf{x}^{(t-1:T)})$) with K weighted atoms (the *particles*). We denote these

Algorithm 3 Residual Resample

```
**Input:** $w_{1:K}$ (weights), $\mathbf{x}_{1:K}$ (particles)
$\forall k,\ (c_k, r_k) \leftarrow (\lfloor K w_k \rfloor, K w_k - \lfloor K w_k \rfloor)$
$\tilde{\mathbf{x}}_C = [\underbrace{\mathbf{x}_1, \ldots, \mathbf{x}_1}_{c_1}, \ldots, \underbrace{\mathbf{x}_K, \ldots, \mathbf{x}_K}_{c_K}]$
$R \leftarrow K - \sum_{k=1}^{K} c_k$
$[i_1, \ldots, i_R] \sim \text{Multinomial}(r_{1:K}, R)$
$\tilde{\mathbf{x}}_R \leftarrow [\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_R}]$
$\tilde{\mathbf{x}} = \text{concat}(\mathbf{x}_R, \mathbf{x}_C)$
Return $\tilde{\mathbf{x}}$
```

approximations (which are implicit in Algorithm 1) by $\mathbb{P}_K^{(t)}(\cdot) := \sum_{k=1}^{K} \tilde{w}_k^{(t)} \delta(\cdot; \mathbf{x}_{\mathcal{S},k}^{(t)})$, where each $\tilde{w}_k^{(t)}$ and $\mathbf{x}_{\mathcal{S},k}^{(t)}$ are as in Algorithm 1, and $\delta(\cdot; \mathbf{x})$ denotes a Dirac mass at $\mathbf{x}$. In particular, $\mathbb{P}_K^{(1)}(\cdot)$ is an approximation to $p_{\mathcal{S},1}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$. Proving the proposition amounts to showing that in the limit as K goes to infinity, each $\mathbb{P}_K^{(1)}(\cdot)$ converges weakly to $p_{\mathcal{S},1}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$, which by assumption is equal to $q_{\mathcal{S},1}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$. This weak convergence follows from standard asymptotics for particle filters (Chopin & Papaspiliopoulos, 2020, Proposition 11.4), which we make explicit in Lemma D.1. As a result, if we perform step (3) with $\mathbf{x}_{\mathcal{S}}^{(1)} \sim \mathbb{P}_K^{(1)}(\cdot)$, then this lemma implies that $\mathbf{x}_{\mathcal{S}}^{(0)}$ converges in distribution to $q_{\mathcal{S},0}(\mathbf{x}_{\mathcal{S}}^{(0)} \mid \mathbf{x}_{\mathcal{M}}^{(0)})$, since (i) $q_{\mathcal{S},0}(\mathbf{x}_{\mathcal{S}}^{(0)} \mid \mathbf{x}_{\mathcal{M}}^{(1)}, \mathbf{x}_{\mathcal{S}}^{(1)})$ is continuous in $\mathbf{x}_{\mathcal{S}}^{(1)}$ and (ii) $\mathbf{x}_{\mathcal{S}}^{(0)}$ is independent of $\mathbf{x}_{\mathcal{M}}^{(0)}$ conditional on $\mathbf{x}^{(1)}$.

Recall that to show the proposition, it was to sufficient to show that $\mathbb{P}_K^{(1)}$ converged weakly to $q_{\mathcal{S},1}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$; this implied that the K particle returned by Algorithm 1 would then converge in distribution to $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$ which, by the law of total probability, implied that they marginally converge to $q_{\mathcal{S},0}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)})$. However, while the particles return by Algorithm 1 may be treated as exchangeable, they are not independent, because they depend on shared randomness in $\mathbf{x}_{\mathcal{M}}^{(1:T)}$. To obtain approximate samples that are independent, it is necessary to run Algorithm 1 multiple times.

Residual resampling. Line 14 of Algorithm 1 indicates a `Resample` step. In particle filtering, resampling steps (or *branching mechanisms* (Doucet et al., 2001, Chapter 2)) filter out particles with very small weights, and replace them with additional copies of particles with large weights. Notably, the resampling step is the only point of departure of Algorithm 1 from the replacement method; without resampling, the algorithms behave identically. While a variety of possible branching mechanisms exist, we use *residual resampling* (Algorithm 3) in our implementation for its simplicity.

D.3 PROOFS AND LEMMAS

Particle filtering lemma with technical conditions

Lemma D.1. *Consider* $\mathbb{P}_K^{(1)} := \sum_{k=1}^{K} \tilde{w}_k \delta(\cdot; \mathbf{x}_{\mathcal{S},k}^{(1)})$*, where* $\tilde{w}_k$ *and* $\mathbf{x}_{\mathcal{S},1:K}^{(1)}$ *are as constructed in Algorithm 1. Assume the conditions of Proposition 4.1. Then* $\mathbb{P}_K^{(1)}$ *converges weakly to* $p_{\mathcal{S},1}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$ *as* K *goes to infinity. That is, for any Borel measurable* A, $\lim_{K\to\infty} \mathbb{P}_K^{(1)}(A) = \int_A p_{\mathcal{S},1}(x \mid \mathbf{x}_{\mathcal{M}}^{(0:T)}) dx$.

Proof. The proof of the lemma follows from an application of standard asymptotics for particle filtering (Chopin & Papaspiliopoulos, 2020, Proposition 11.4). In particular, to apply Proposition 11.4 we use the formalism of Feynman–Kac (FK) models, following the notation of (Chopin & Papaspiliopoulos, 2020, Chapter 5). Though typically (and in (Chopin & Papaspiliopoulos, 2020)) FK models are defined via a sequence of approximations at increasing time steps, we consider decreasing time steps because we are approximating the reverse time process. We take the initial distribution as $\mathbb{M}_T(\mathbf{x}_{\mathcal{S}}^{(T)}) = p_{\mathcal{S},T}(\mathbf{x}_{\mathcal{S}}^{(T)})$, the transition kernel as $M_t(\mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{S}}^{(t)}) = p_{\mathcal{S},t}(\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}^{(t+1)})$, and the potential functions as $G_t(\mathbf{x}_{\mathcal{S}}^{(t)}) = p_{\mathcal{M},t-1}(\mathbf{x}_{\mathcal{M}}^{(t-1)} \mid \mathbf{x}^{(t)})$. The sequence of FK models, $\mathbb{Q}_t$, then correspond to

$$\mathbb{Q}_t(\mathbf{x}_{\mathcal{S}}^{(t:T)}) = L_t^{-1} \mathbb{M}_T(\mathbf{x}_{\mathcal{S}}^{(T)}) G_T(\mathbf{x}_{\mathcal{S}}^{(T)}) \prod_{i=T-1}^{t} M_i(\mathbf{x}_{\mathcal{S}}^{(i+1)}, \mathbf{x}_{\mathcal{S}}^{(i)}) G_i(\mathbf{x}_{\mathcal{S}}^{(i)})$$

for each t, where L_t is a normalizing constant.

By substituting in our choices of M_t and G_t, we can rewrite and simplify $\mathbb{Q}_t$ as

$$\begin{aligned}
\mathbb{Q}_t(\mathbf{x}_{\mathcal{S}}^{(t:T)}) &= L_t^{-1} p_{\mathcal{S},T}(\mathbf{x}_{\mathcal{S}}^{(T)}) p_{\mathcal{M},T-1}(\mathbf{x}_{\mathcal{M}}^{(T-1)} \mid \mathbf{x}^{(T)}) \prod_{i=T-1}^{t} p_{\mathcal{S},i}(\mathbf{x}_{\mathcal{S}}^{(i)} \mid \mathbf{x}^{(i+1)}) p_{\mathcal{M},i-1}(\mathbf{x}_{\mathcal{M}}^{(i-1)} \mid \mathbf{x}^{(i)}) \\
&= L_t^{-1} p_{\mathcal{S},T}(\mathbf{x}_{\mathcal{S}}^{(T)}) p_{t:T-1}(\mathbf{x}^{(t:T-1)} \mid \mathbf{x}^{(T)}) p_{\mathcal{M},t-1}(\mathbf{x}_{\mathcal{M}}^{(t-1)} \mid \mathbf{x}^{(t)}) \\
&\propto p_{t:T}(\mathbf{x}^{(t:T)} \mid \mathbf{x}_{\mathcal{M}}^{(t-1)}) \\
&\propto p_{\mathcal{S},t:T}(\mathbf{x}_{\mathcal{S}}^{(t:T)} \mid \mathbf{x}_{\mathcal{M}}^{(t-1:T)}),
\end{aligned}$$

where lines 3 and 4 drop multiplicative constants that do not depend on $\mathbf{x}_{\mathcal{S}}^{(t:T)}$. From the above derivation, we see that each $\mathbb{Q}_t(\mathbf{x}_{\mathcal{S}}^{(t)}) = p_{\mathcal{S},t}(\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{M}}^{(t-1:T)})$, and in particular that $\mathbb{Q}_1(\mathbf{x}_{\mathcal{S}}^{(1)}) = p_{\mathcal{S},1}(\mathbf{x}_{\mathcal{S}}^{(1)} \mid \mathbf{x}_{\mathcal{M}}^{(0:T)})$. As such, the desired convergence in the statement of the lemma is equivalent to that $\mathbb{P}_K^{(1)}$ converges to $\mathbb{Q}_1$.

Chopin & Papaspiliopoulos (2020, Proposition 11.4) provide this result for the generic particle filtering algorithm (see Chopin & Papaspiliopoulos (2020, Algorithm 10.1), which is written in the FK model form described above). More specifically, Proposition 11.4 proves almost sure convergence of all Borel measurable functions of $\mathbb{P}_K^{(t)}$, which implies the desired weak convergence.

Although the proof provided in Chopin & Papaspiliopoulos (2020) is restricted to the simpler, but higher variance, case where the resampling step uses multinomial resampling, the authors note that Chopin (2004) proves it holds in the case of residual resampling (which we use in our experiments) as well. □

Replacement method error — lemmas and proofs

We here provide proofs of Proposition D.1 and Corollary D.2.

Proof of Proposition D.1:

Proof. The result obtains from recognizing where the replacement method approximation agrees with the forward process, using conditional independences in both processes, and applying the chain rule for KL divergences. We make this explicit in the derivation below, with comments explaining the transition to the following line.

$$\mathrm{KL}\left[q_{1:T}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \| p_{1:T}^{\mathrm{Repl}}(\cdot \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})\right]$$

$$= \int q_{1:T}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \log \frac{q_{1:T}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})}{p_{1:T}^{\mathrm{Repl}}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})} dx^{(1:T)}$$

// By the chain rule of probability.

$$= \int q_{1:T}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \Big[\log \frac{q_{\mathcal{M},1:T}(x_{\mathcal{M}}^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})}{p_{1:T}^{\mathrm{Repl}}(x_{\mathcal{M}}^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}})} +$$

$$\log \frac{q_{\mathcal{S},1:T}(x_{\mathcal{S}}^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})}{p_{\mathcal{S},1:T}^{\mathrm{Repl}}(x_{\mathcal{S}}^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})} \Big]$$

// By the agreement of q and p_{Repl} on the motif, and the chain rule of probability.

$$= \int q_{1:T}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \Big[\log \frac{q_{\mathcal{S},T}(x_{\mathcal{S}}^{(T)} \mid \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})}{p_{T}^{\mathrm{Repl}}(x_{\mathcal{S}}^{(T)} \mid \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})} +$$

$$\sum_{t=1}^{T-1} \log \frac{q_{\mathcal{S},t}(x_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})}{p_{\mathcal{S},t}^{\mathrm{Repl}}(x_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0:T)} = x_{\mathcal{M}}^{(0:T)})} \Big] dx^{(1:T)}$$

// Because $q_{\mathcal{S},T}(\cdot) = p_{\mathcal{S},T}^{\mathrm{Repl}}(\cdot) = \mathcal{N}(\cdot; 0, I)$ and the assumption that p_θ matches q.

$$= \int q_{1:T}(x^{(1:T)} \mid \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}) \Big[\sum_{t=1}^{T-1} \log \frac{q_{\mathcal{S},t}(x_{\mathcal{S}}^{(t)} \mid \mathbf{x}^{(t+1)} = x^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}^{(0)})}{q_{\mathcal{S},t}(x_{\mathcal{S}}^{(t)} \mid \mathbf{x}^{(t+1)} = x^{(t+1)})} \Big] dx^{(1:T)}$$

$$= \sum_{t=1}^{T-1} \mathrm{EKL}\left[q_{\mathcal{S},t}(\cdot \mid \mathbf{x}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}^{(0)}) \| q_{\mathcal{S},t}(\cdot \mid \mathbf{x}^{(t+1)})\right].$$

□

Proof of Corollary D.2:

The proof of the corollary relies of on a lemma on the variances of the two relevant conditional distributions. We state this lemma, whose proof is at the end of the section, before continuing. For notational simplicity, we drop the scripts and annotations on $\bar{\alpha}^{(t)}$ and $\beta^{(t+1)}$, and instead write α and β, respectively.

Lemma D.2. *Suppose* $\mathbf{x}_{\mathcal{M}}^{(0)}, \mathbf{x}_{\mathcal{S}}^{(t)}$, *and* $\mathbf{x}_{\mathcal{S}}^{(t+1)}$ *are distributed as in Corollary D.2. Then* $\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}] = \beta$ *and* $\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}] \leq \beta(1 - \beta\rho^2\alpha)$.

Now we provide a proof of Corollary D.2.

Proof. First recall that

$$\mathrm{KL}\left[\mathcal{N}(\mu_1, \sigma_1^2) \| \mathcal{N}(\mu_2, \sigma_2^2)\right] = \frac{1}{2}\left(\log \frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{\sigma_2^2} - 1\right)$$

$$\geq \frac{1}{2}\left(\log \frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2}{\sigma_2^2} - 1\right)$$

and observe that this lower bound is monotonically decreasing in σ_1^2 for $\sigma_1^2 \leq \sigma_2^2$. Therefore

$$\begin{aligned}
&\mathrm{EKL}\left[q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}) \| q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)})\right] \\
&= \int q_{\mathcal{M},0}(x_{\mathcal{M}}^{(0)}) q_{\mathcal{S},t+1}(x_{\mathcal{S}}^{(t+1)} \mid x_{\mathcal{M}}^{(0)}) \Big[\\
&\quad \mathrm{KL}\left[q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}^{(0)}]) \| q_{\mathcal{S},t}(\cdot \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}])\right] \\
&\Big] dx_{\mathcal{M}}^{(0)} x_{\mathcal{S}}^{(t+1)} \\
&\geq \int q_{\mathcal{M},0}(x_{\mathcal{M}}^{(0)}) q_{\mathcal{S},t+1}(x_{\mathcal{S}}^{(t+1)} \mid x_{\mathcal{M}}^{(0)}) \Big[\\
&\quad \mathrm{KL}\left[\mathcal{N}(0, \mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)} = x_{\mathcal{M}}^{(0)}]) \| \mathcal{N}(0, \mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)} = x_{\mathcal{S}}^{(t+1)}])\right] \\
&\Big] dx_{\mathcal{M}}^{(0)} x_{\mathcal{S}}^{(t+1)} \\
&\geq \mathrm{KL}\left[\mathcal{N}(0, \beta(1-\beta\rho^2\alpha)) \| \mathcal{N}(0,\beta)\right] \\
&\geq \frac{1}{2}\left(\log\frac{\beta}{\beta(1-\beta\rho^2\alpha)} + \frac{\beta(1-\beta\rho^2\alpha)}{\beta} - 1\right) \\
&= -\frac{1}{2}\left(\log(1-\beta\rho^2\alpha) + \beta\rho^2\alpha\right)
\end{aligned}$$

where the second inequality follows from Lemma D.2, and the monotonicity of the KL in σ_1^2. ☐

Proof of Lemma D.2:

Proof. That $\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}] = \beta$ follows immediately from that $[\mathbf{x}_{\mathcal{S}}^{(t)}, \mathbf{x}_{\mathcal{S}}^{(t+1)}]$ is marginally bivariate normal distributed with covariance $\sqrt{1-\beta}$.

The upper bound on $\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}]$ is trickier. Observer that $[\mathbf{x}_{\mathcal{S}}^{(t)}, \mathbf{x}_{\mathcal{S}}^{(t+1)}] \mid \mathbf{x}_{\mathcal{M}}^{(0)}$ is bivariate Gaussian and that

$$\mathrm{Var}\left[\begin{bmatrix}\mathbf{x}_{\mathcal{S}}^{(t)} \\ \mathbf{x}_{\mathcal{S}}^{(t+1)}\end{bmatrix} \mid \mathbf{x}_{\mathcal{M}}^{(0)}\right] = \begin{bmatrix} 1-\rho^2\alpha & \sqrt{1-\beta}(1-\rho^2\alpha) \\ \sqrt{1-\beta}(1-\rho^2\alpha) & 1+\beta\rho^2\alpha-\rho^2\alpha \end{bmatrix}.$$

As such, the conditional variance may be computed in closed form as $\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}] = \beta(1-\rho^2\alpha)+(1-\beta)(1-\rho^2\alpha)\left(1-(1-\rho^2\alpha)/(1-\rho^2\alpha+\beta\rho^2\alpha)\right)$. But since $(1-\rho^2\alpha)/(1-\rho^2\alpha+\beta\rho^2\alpha) \geq 1-(\beta\rho^2\alpha)/(1-\rho^2\alpha)$ and therefore $1-(1-\rho^2\alpha)/(1-\rho^2\alpha+\beta\rho^2\alpha) \leq (\beta\rho^2\alpha)/(1-\rho^2\alpha)$ we can write

$$\begin{aligned}
\mathrm{Var}[\mathbf{x}_{\mathcal{S}}^{(t)} \mid \mathbf{x}_{\mathcal{S}}^{(t+1)}, \mathbf{x}_{\mathcal{M}}^{(0)}] &= \beta(1-\rho^2\alpha) + (1-\beta)(1-\rho^2\alpha)\left(1 - \frac{1-\rho^2\alpha}{1-\rho^2\alpha+\beta\rho^2\alpha}\right) \\
&\leq \beta(1-\rho^2\alpha) + (1-\beta)(1-\rho^2\alpha)\frac{\beta\rho^2\alpha}{1-\rho^2\alpha}) \\
&= \beta(1-\rho^2\alpha) + (1-\beta)\beta\rho^2\alpha \\
&= \beta(1-\beta\rho^2\alpha).
\end{aligned}$$

☐

E DETECTING CHIRALITY

Section 6 noted the limitation of `ProtDiff` that it can generate left-handed helices (which do not stably occur in natural proteins). Figure 6 presents two such examples. We additionally note that, as in Figure 6 Left, model samples can include multiple helices with differing chirality.

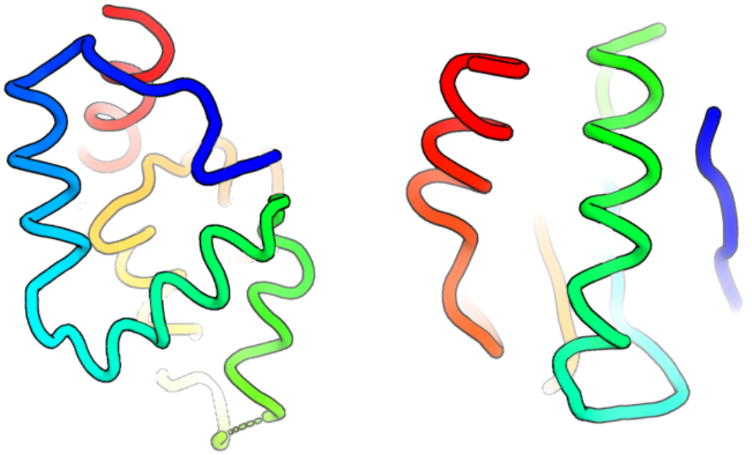


Figure 6: Two examples of protein backbone samples with incorrect left handed helices.

F TRAINING DETAILS

`ProtDiff` uses 4 equivariant graph convolutional layers (EGCL) with 256 dimensions for node and edge embeddings. The training data was restricted to single chain proteins (monomers) found in PDB and lengths in the range [40, 128]. We additionally filtered out PDB with >5Å atomic resolution. This amounted to 4269 training examples. Training was performed using the Adam optimizer with hyperparameters `learning_rate`=1e-4, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We trained for 1,000,000 steps using batch size 16. We used a single Nvidia A100 GPU for approximately 24 hours. We implemented all models in PyTorch. We used the same linear noise schedule as Ho et al. (2020) where $\beta_0 = 0.0001$, $\beta_T = 0.02$, and $T = 1024$. We did not perform hyperparameter tuning.

G ADDITIONAL METRIC DETAILS

Self-consistency algorithm. Section 5.1 described our self-consistency metrics for evaluating the designability of backbones generated with `ProtDiff`. Algorithm 4 makes explicit the procedure we use for computing these metrics.

Algorithm 4 Self-consistency calculation

```
Input: $\mathbf{x} \in \mathbb{R}^{N,3}$
for $i \in 1, \dots, 8$ do
    $s_i \leftarrow \texttt{ProteinMPNN}(\mathbf{x})$
    $\hat{\mathbf{x}}_i \leftarrow \texttt{AF2}(s_i)$
end for
$\texttt{sc\_tm} \leftarrow \max_{i \in 1,\dots,8} \texttt{TMscore}(\hat{\mathbf{x}}_i, \mathbf{x})$
Output: , $\texttt{sc\_tm}$
```

Using dihedral angles to calculate helix chirality. Natural proteins are chiral molecules that contain only right-handed alpha helices. However, because the underlying EGNN in our model is equivariant to reflection, it can produce samples with left-handed helices. While examining model samples, we additionally observed samples with both left and right-handed helices (Figure 6), even though in theory the EGNN should be able to detect and avoid the chiral mismatch. Left-handed helices are fundamentally invalid geometries in proteins and represent a trivial failure mode when calculating the self-consistency and other metrics. Samples with a mixture of left and right-handed helices are especially problematic because they cannot be corrected simply by reflecting the coordinates. As such, it is important to identify and separate samples with mixed chirality.

To detect chiralty, we compute the dihedral angle between four consecutive C-α atoms as a chiral metric to distinguish between the two helix chiralities. Algorithmically, for every C-α `i`, we calculate the dihedral between C-α `i`, `i+1`, `i+2`, and `i+3`. C-α `i` with dihedral angles between 0.6 and 1.2 radians are classified as right-handed helices, and angles between -1.2 and -0.6 are classified as left-handed helices, with everything else classified as non-helical. Because C-α atoms in native helices tend to form contiguous stretches longer than one residue in the primary sequence, helical stretches less than one amino acid were removed. This filtering is meant to help avoid accidentally counting the occasional isolated backbone geometry that falls into a helical bin as a true helix. Finally, for all C-α atoms `i` that are still categorized as part of a helix, the associated `i+1`, `i+2` and `i+3` C-α atoms are also counted as part of that helix.

H ADDITIONAL EXPERIMENTAL RESULTS

In this section, we describe additional results to complement the main text. We provide a description of the motif targets in Section 4, along with results of a scaffolding failure case in Appendix H.1. To understand the qualitative outcomes of `scTM`, we present additional results of backbone designs, their AF2 prediction, and most closely related PDB parent chain for different thresholds of `scTM` in Appendix H.2. We provide additional examples of latent interpolations in Appendix H.3. Finally, Appendix H.4 presents a structural clustering of unconditional backbone samples; this result provides further evidence of `ProtDiff`'s ability to generate diverse backbone structures.

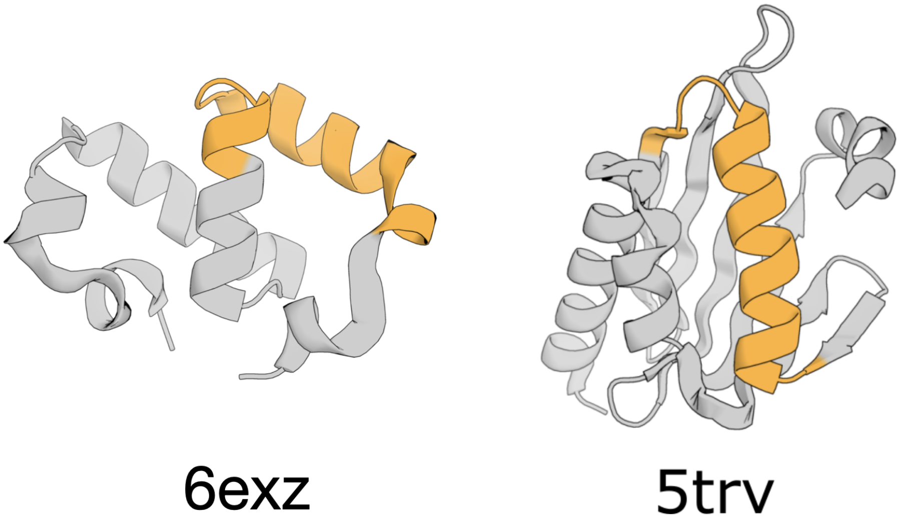


Figure 7: Structures used for motif-scaffolding test cases. Native structures (grey) and their motifs (orange) that were used for the motif-scaffolding task are shown.

H.1 ADDITIONAL MOTIF-SCAFFOLDING RESULTS

We here provide additional details of the motif-scaffolding experiments described in Section 5. Table 1 specifies the total lengths, motif sizes, and motif indices of our test cases. In Figure 7 we depict the structures of the native proteins (6exz and 5trv) from which the motifs examined quantitatively in the main text were extracted. Figure 8 analyzes commonly observed failure modes of `ProtDiff` backbone samples involving chain breaks, steric clashes, and incorrect chirality.

Figure 9 presents quantitative results on a harder inpainting target. In this case, the motif is defined as residues 163–181 of chain A of respiratory syncytial virus (RSV) protein (PDB ID: 5tpn). We attempted to scaffold this motif into a 62 residue protein, with the motif as residues 42–62. We chose this placement because previous work (Wang et al., 2022) identified a promising candidate scaffold with this motif placement. In contrast to the cases described in the main text, for which a suitable scaffold exists in the training set, `SMCDiff` and the other inpainting methods failed to identify scaffolds that recapitulated this motif to within a motif RMSD of 1 Å.

Table 1: Motif-scaffolding test case additional details.

Origin/ Protein	Total length	Motif size (residue range)
6exz	72	15 (30–44)
5trv	118	21 (42–62)
`RSV` (PDB-ID: 5tpn)	62	19 (16–34)
`EF-hand` (PDB-ID: 1PRW)	53	5 (0–4), 13 (31–43)

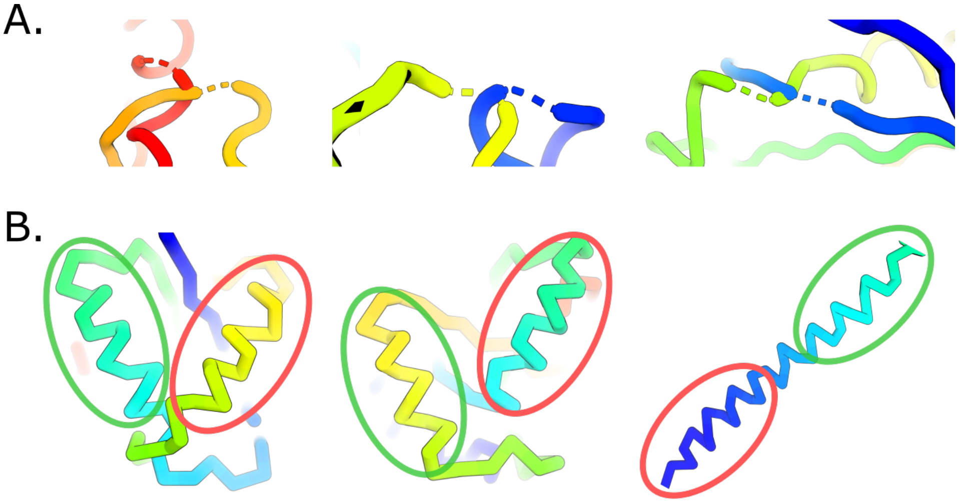


Figure 8: Failure modes in `ProtDiff` backbone samples. (A) Backbone clashes and chain breaks. The C-α atoms can be spaced further than the typical 3.8Å between neighbors, resulting in a chain break (dashed lines). Additionally, backbone segments can be too close to each other, resulting in obvious overlaps and clashes. (B) Backbones with a mixture of left (circled in red) and right (circled in green) handed helices. These chirality errors cannot be corrected simply by mirroring the sampled backbone.

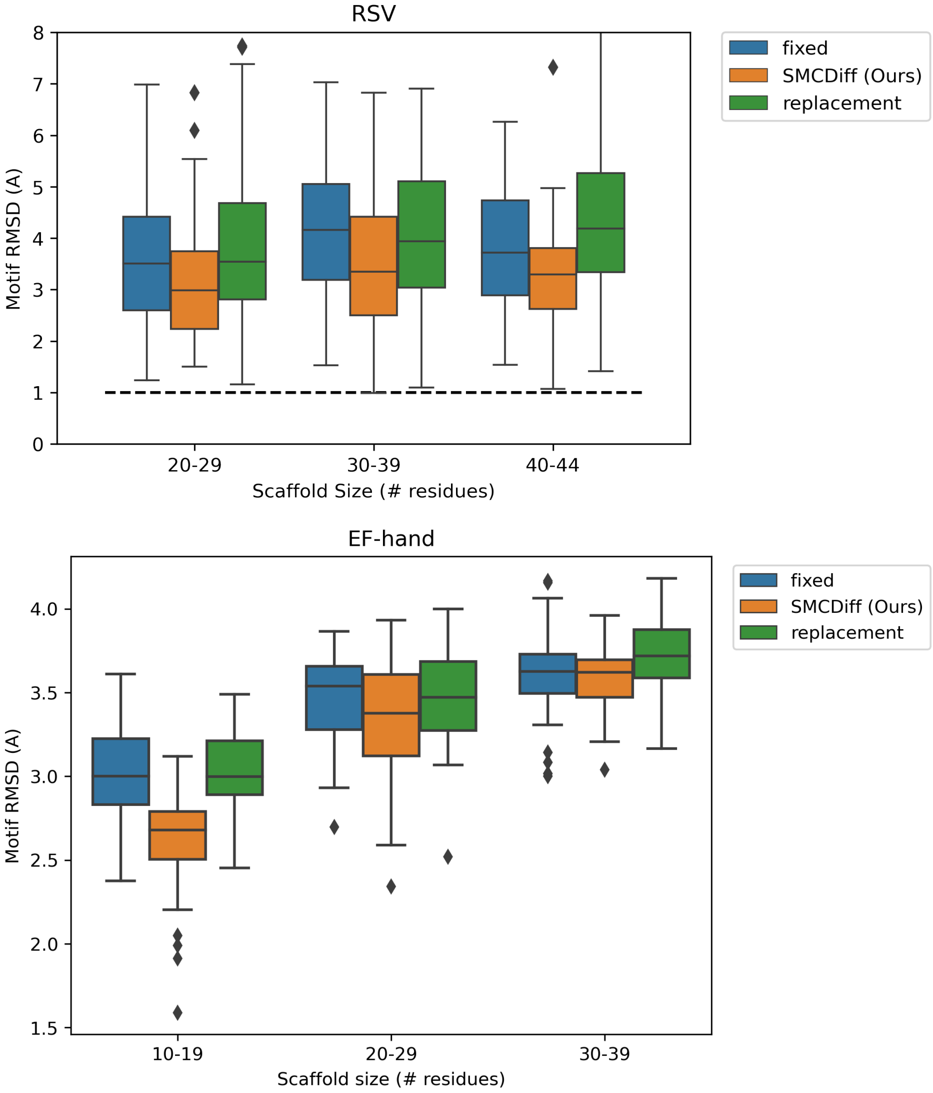


Figure 9: Additional inpainting results on a more challenging motif extracted from the respiratory syncytial virus (RSV) and EH-hand motif. The three inpainting methods are evaluated as described in Section 5.

H.2 QUALITATIVE ANALYSIS OF `scTM` IN DIFFERENT RANGES

In this section, we give intuition for backbone designs and AF2 predictions associated with different values of `scTM` to aid the interpretation of the `scTM` results provided in Section 5. Figure 10 examines a possible categorization of `scTM` in three ranges. The first two rows correspond to backbone designs that achieve `scTM` > 0.9. We see the backbone designs in the first column closely match the AF2 prediction in the second column. A closely related PDB example can be found when doing a similarity search of the highest PDB chain with the highest TM-score to the AF2 prediction. We showed in Figure 3B that `scTM` > 0.9 is indicative of a close structural match being found in PDB.

The middle two rows correspond to designs that achieve `scTM` ~ 0.5. These are examples of backbone designs on the edge of what we deemed as designable (`scTM` > 0.5). In these cases, the AF2 prediction shares the same coarse shape as the backbone design but possibly with different secondary-structure ordering and composition. In the length 69 example, we see the closest PDB chain has a TM-score of only 0.65 to the AF2 prediction but roughly the same secondary-structure ordering as the backbone design. The length 100 sample is a similar case of AF2 producing a roughly similar shape to the backbone design, but has no matching monomer in PDB.

The final category of `scTM` < 0.25 reflects failure cases when `scTM` is low. The AF2 predictions in this case have many disordered regions and bear little structural similarity with the original backbone design. Similar PDB chains are not found. We expect that improved generative models of protein backbones would not produce any samples in this category.

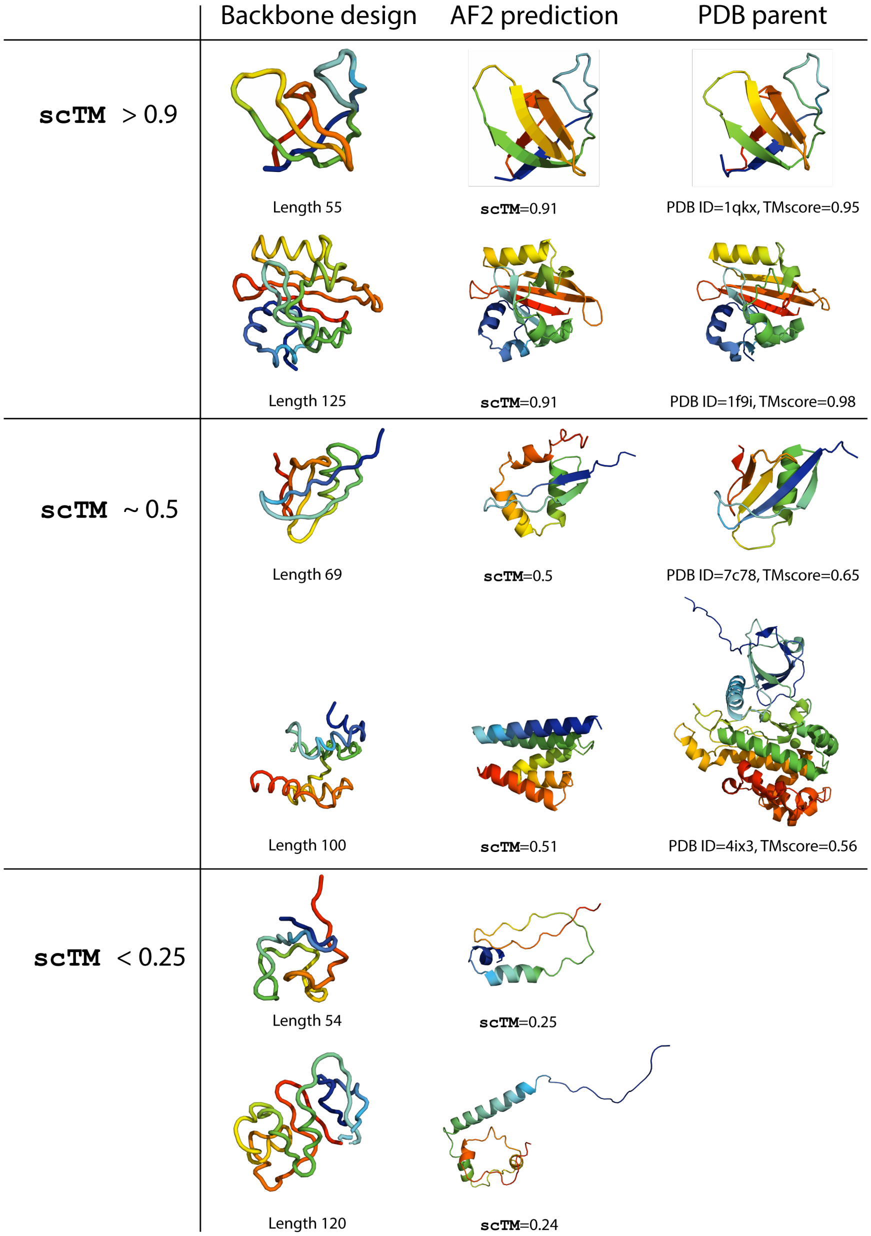
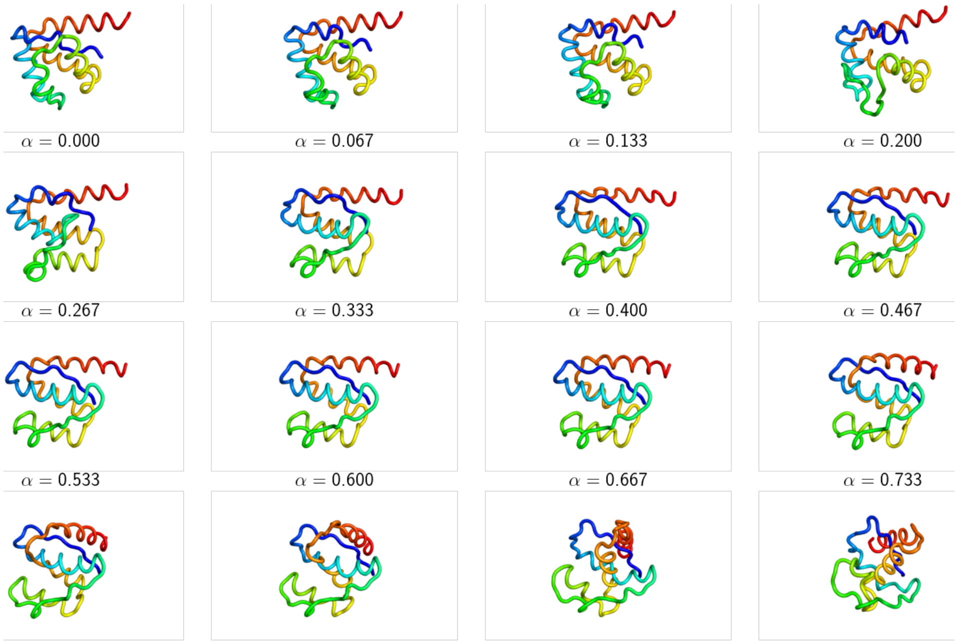


Figure 10: Qualitative analysis of unconditional backbone samples from `ProtDiff`. The first column displays backbone designs from `ProtDiff` and their sequence lengths. The second column displays the highest `scTM` scoring AF2 predictions from the `ProteinMPNN` sequences of the corresponding backbone design in the first column. The third column displays the closest PDB chain to the AF2 prediction in the second column with the PDB ID and TM-score written below. The third column is blank for the last two rows since no PDB match could be found. See Appendix H.2 for discussion.

H.3 ADDITIONAL LATENT INTERPOLATION RESULTS

We here provide additional latent interpolations. Figures 11 and 12 depict interpolations for between model samples for lengths 89 and 63, respectively.



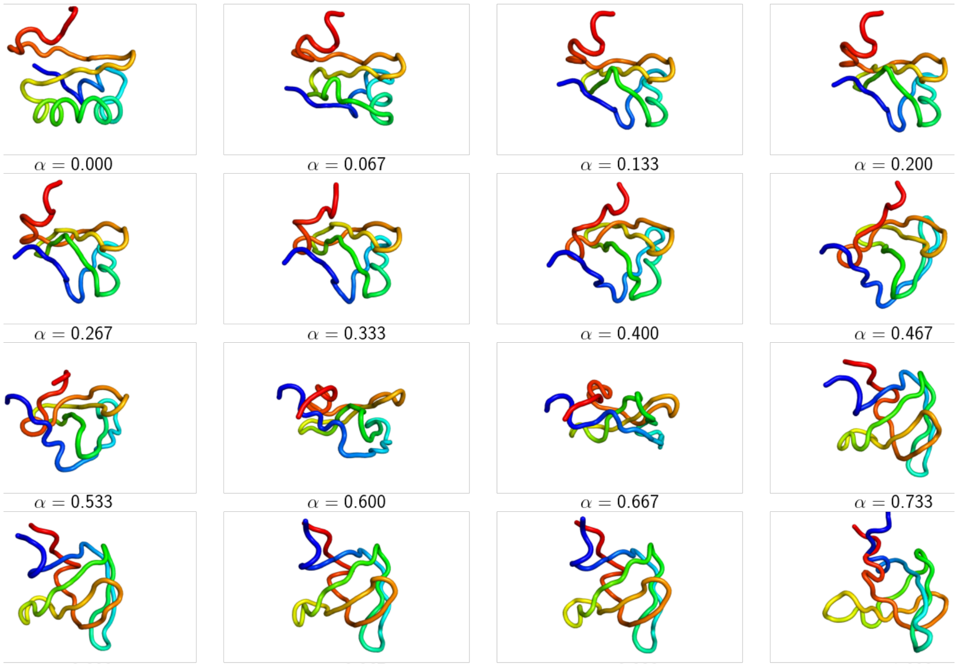
$\alpha = 0.000$ $\alpha = 0.067$ $\alpha = 0.133$ $\alpha = 0.200$

$\alpha = 0.267$ $\alpha = 0.333$ $\alpha = 0.400$ $\alpha = 0.467$

$\alpha = 0.533$ $\alpha = 0.600$ $\alpha = 0.667$ $\alpha = 0.733$

$\alpha = 0.800$ $\alpha = 0.867$ $\alpha = 0.933$ $\alpha = 1.000$

Figure 11: Latent interpolation of length 89 backbone sample from $\alpha = 0$ to 1.



$\alpha = 0.000$ $\alpha = 0.067$ $\alpha = 0.133$ $\alpha = 0.200$

$\alpha = 0.267$ $\alpha = 0.333$ $\alpha = 0.400$ $\alpha = 0.467$

$\alpha = 0.533$ $\alpha = 0.600$ $\alpha = 0.667$ $\alpha = 0.733$

$\alpha = 0.800$ $\alpha = 0.867$ $\alpha = 0.933$ $\alpha = 1.000$

Figure 12: Latent interpolation of length 63 backbone sample from $\alpha = 0$ to 1.

H.4 STRUCTURAL CLUSTERING

All 92 samples with `scTM` > 0.5 were compared and clustered using MaxCluster Herbert & Sternberg (2008). Structures were compared in a sequence independent manner, using the TM-score of the maximal subset of paired residues. They were subsequently clustered using hierarchical clustering with average linkage, 1 - TM-score as the distance metric and a TM-score threshold of 0.5 (Figure 13 A).

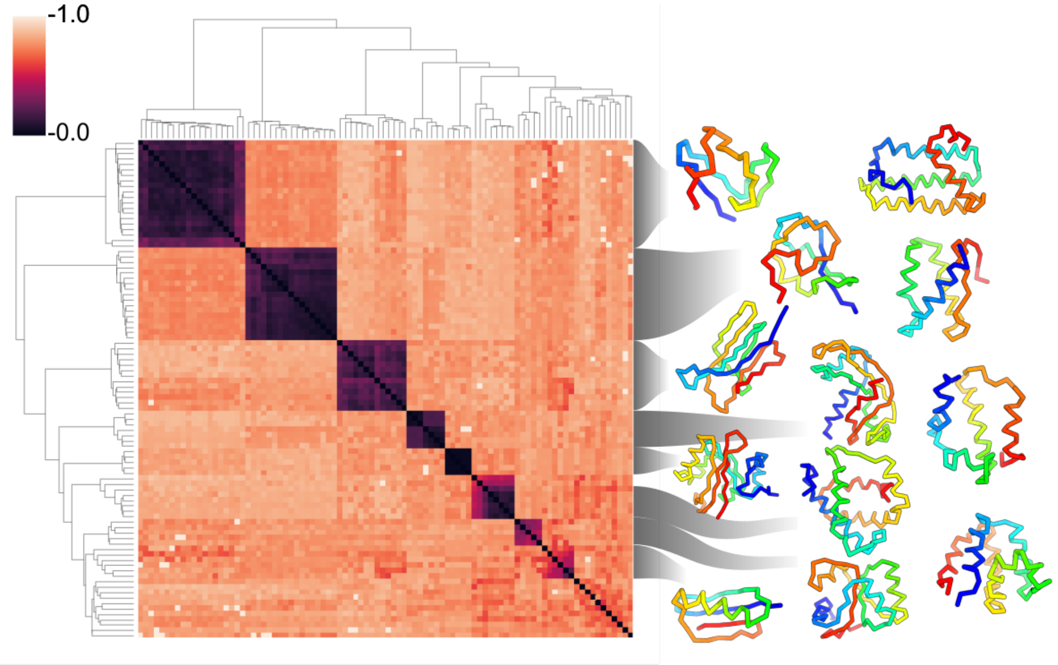


Figure 13: Clustering of self-consistent `ProtDiff` samples. The distance matrix is 1 - TM-score between pairs of samples, and ranges from 0 (exact matach) to 1 (no match). Dendrograms are from hierarchical clustering using the average distance metric. Designs on the right are cluster centroids. Gray lines connect larger clusters with more than one member to its centroid, while the remaining designs are from a random selection of the remaining single-sample clusters. Protein backbones are colored from blue at the N-terminus to red at the C-terminus.

I APPLICABILITY OF SMCDIFF BEYOND PROTEINS: MNIST INPAINTING

Our goal in this section is to study the applicability of `SMCDiff` beyond motif-scaffolding, by applying it to inpainting on the MNIST digits dataset. We compare `SMCDiff` with the replacement method on the task of sampling the remaining half of MNIST digits. We first train DDPM with $\beta_1 = 10^{-4}, \beta_T = 0.2, T = 1000$ using a small 8-layer CNN on MNIST with batch size 128 and ADAM optimizer for 100 epochs until it is able to generate reasonable MNIST samples (Figure 14). We then selected 3 random MNIST images and occluded the right half. The left half would then serve as the conditioning information to the diffusion model (Figure 15).

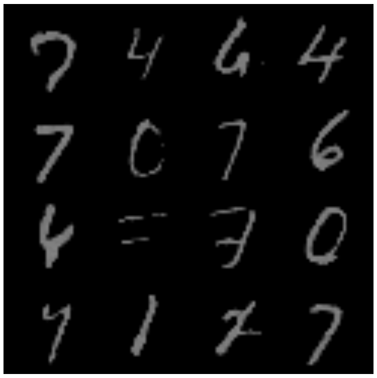


Figure 14: Unconditional MNIST samples.

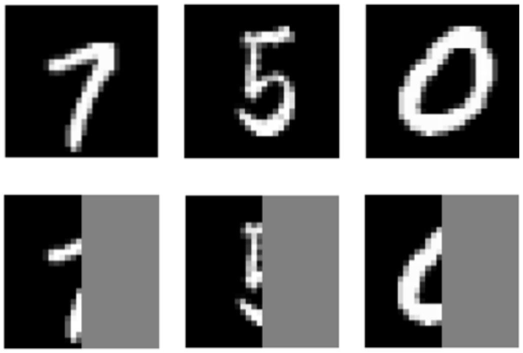


Figure 15: Full MNIST images and their occluded halves used for inpainting experiments.

For each occluded image, we fixed a single forward trajectory and sampled 16 images from each method: replacement method and `SMCDiff` with 16 or 64 particles (K). Results are shown in Fig. 16. We observe the replacement method can sometimes produce coherent samples as a continuation of the conditioning information, but more often it attempts to produce incoherent digits. `SMCDiff` on the other hand tends to produce digits that compliment the conditioning information. For more difficult occlusions, such as 5 and 0, `SMCDiff` can still fail although increasing the number of particles $(K = 64)$ tends to produce samples that are more visually coherent.

It is important to note SMCDiff has additional computation overhead based on the number of particles. It can be more expensive than replacement method but result in higher quality samples. Investigating `SMCDiff` in more difficult datasets with improved architectures is a direction of future research.

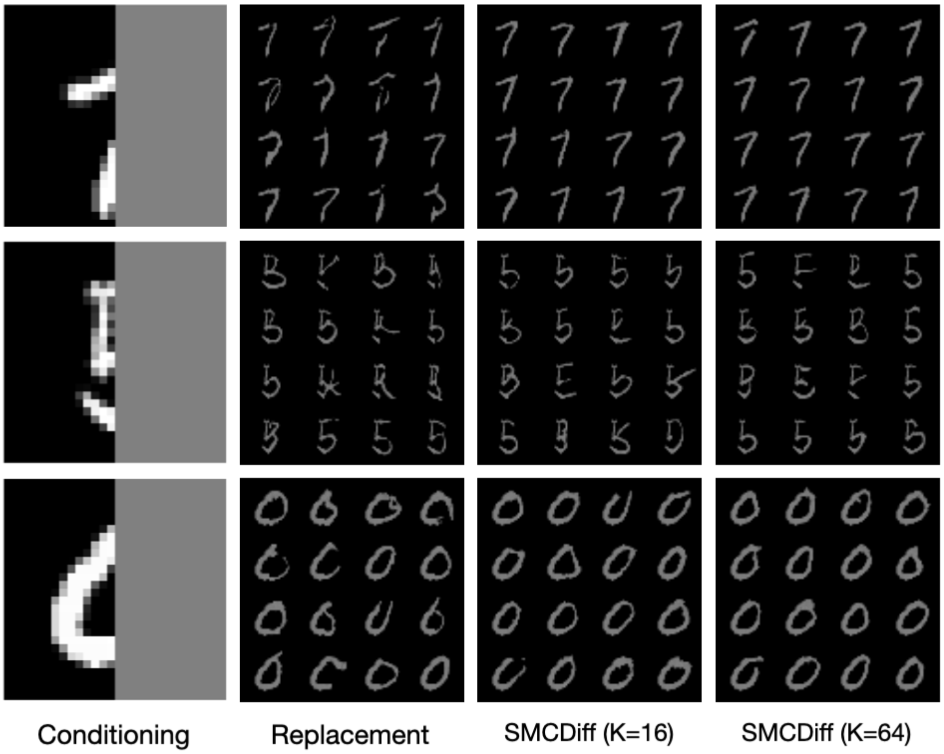


Figure 16: MNIST inpainting results for replacement and `SMCDiff`. See text for explanation.