Protein structure generation via folding diffusion

Written By: Wu et. al
Collaboration of: Stanford University and MSFT Research
Presented By: Bernard Moussad
November 16, 2023

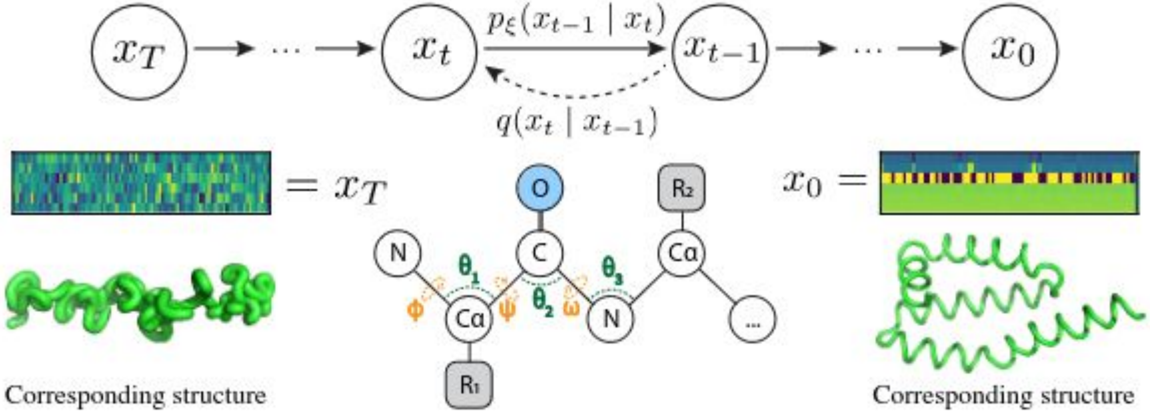
The Problem

- In the grand scheme of life, humans are plagued with various types of “incurable” disease
  - Huntington’s
  - Parkinson’s
  - Alzheimer’s
  - Cystic fibrosis
- Proteins by their nature are capable of performing complex tasks with “high specificity”
  - Not likely to stumble across the right protein *in-vitro* or *in-vivo* thus designing proteins becomes the more tenable approach

Prior Attempts to addressing Protein Structure Generation

- Pairwise/Orientation restraints generation (GANs)
    - Must be post-processed via some methods (ex. pyRosetta)
- Protein Assembly Heuristics
    - Time consuming and restricted to only “known proteins”
- Equivariant Diffusion on 3D Point clouds/Coordinates
    - Can exhibit issues with chirality (handedness) of the protein
- VAE with equivariant loss to generate backbones in 3D space (IgVAE)
    - Required refinement via Rosetta and only worked on small immunoglobulin proteins

Can we diffuse on the angular space instead of the coordinate space?

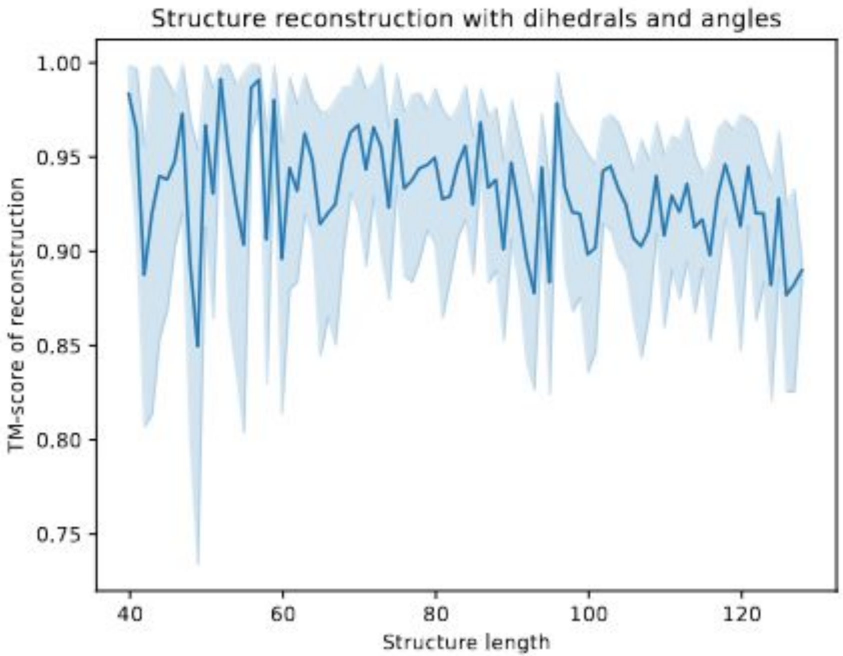


Why should we do so?

- Can prevent the issue of incorrect “handedness” due to removal of dependence on equivariance
- Diffusion on the angular space in a sense echoes the “folding” of proteins in nature
- Simple to reconstruct protein geometry via trigonometry and idealized bond lengths

Disadvantages to this approach

- Reliance on idealized bond lengths
    - Yet does not appear to accumulate errors



What angles should we focus on?

Angle	Description
Ψ (Psi)	Dihedral torsion about $N_i - C\alpha_i - C_i - N_{i+1}$
Ω (Omega)	Dihedral torsion about $C\alpha_i - C_i - N_{i+1} - C\alpha_{i+1}$
Φ (Phi)	Dihedral torsion about $C_i - N_{i+1} - C\alpha_{i+1} - C_{i+1}$
θ_1 (Theta 1)	Bond angle about $N_i - C\alpha_i - C_i$
θ_2 (Theta 2)	Bond angle about $C\alpha_i - C_i - N_{i+1}$
θ_3 (Theta 3)	Bond angle about $C_i - N_{i+1} - C\alpha_{i+1}$

Method Formulation

- Swap from standard normal to wrapped normal distribution when sampling for the Markov forward noising

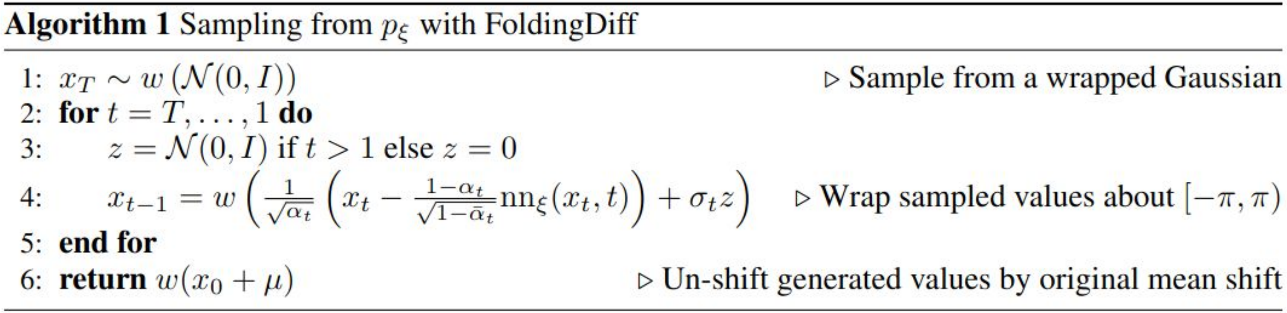
$$q(x_t \mid x_{t-1}) = \mathcal{N}_{\text{wrapped}}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I) \propto \sum_{k=-\infty}^{\infty} \exp\left(\frac{-\|x_t - \sqrt{1-\beta_t} x_{t-1} + 2\pi k\|^2}{2\beta_t^2}\right)$$

- $\beta_t \in (0, 1)^T_{t=1}$ set by cosine variance schedule with T = 1000 timesteps
  - Add s = 8e-3 for numerical stability

$$\beta_t = \text{clip}\left(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999\right) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)$$

Method Formulation (cont.)

- Network is trained using a model that predicts the noise given the timestep: nnξ (xt, t)
    - As opposed to the denoised mean
- Adopted a vanilla bidirectional transformer architecture with relative positional embeddings for the reverse (denoising) model: pξ (x_{t-1}|x_t)



Algorithm 1 Sampling from p_ξ with FoldingDiff

```
$x_T \sim w(\mathcal{N}(0, I))$                                   ▷ Sample from a wrapped Gaussian
for $t = T, \ldots, 1$ do
    $z = \mathcal{N}(0, I)$ if $t > 1$ else $z = 0$
    $x_{t-1} = w\left(\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathrm{nn}_\xi(x_t, t)\right) + \sigma_t z\right)$   ▷ Wrap sampled values about $[-\pi, \pi)$
end for
return $w(x_0 + \mu)$                 ▷ Un-shift generated values by original mean shift
```

Loss Calculations

- Introduced a function to “wrap” values within the range [−π, π): w(x) = ((x + π) mod 2π) − π
  - Handles periodic nature of angular values
- Set βL = 0.1π for L_{ω} (loss formulation which behaves similar to Huber Loss)

$$[-\pi, \pi):\ w(x) = ((x + \pi) \bmod 2\pi)$$

$$\beta_t = \text{clip}\left(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999\right) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad f(t) = \cos\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right)$$

$$d_w = w\left(\epsilon - \text{nn}_{\xi}\left(w\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon\right), t\right)\right) \qquad \epsilon \sim \mathcal{N}(0, I)$$

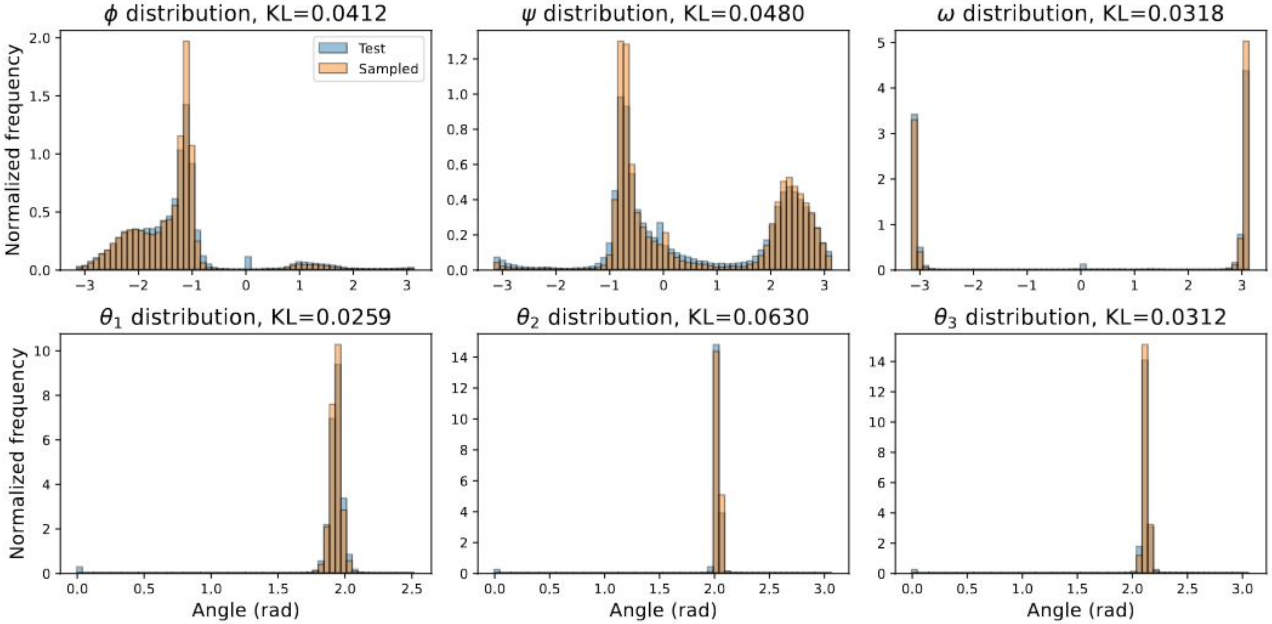
$$L_w = \begin{cases} 0.5\frac{d_w^2}{\beta_L} & \text{if } |d_w| < \beta_L \\ |d_w| - 0.5\beta_L & \text{otherwise} \end{cases}$$

Training

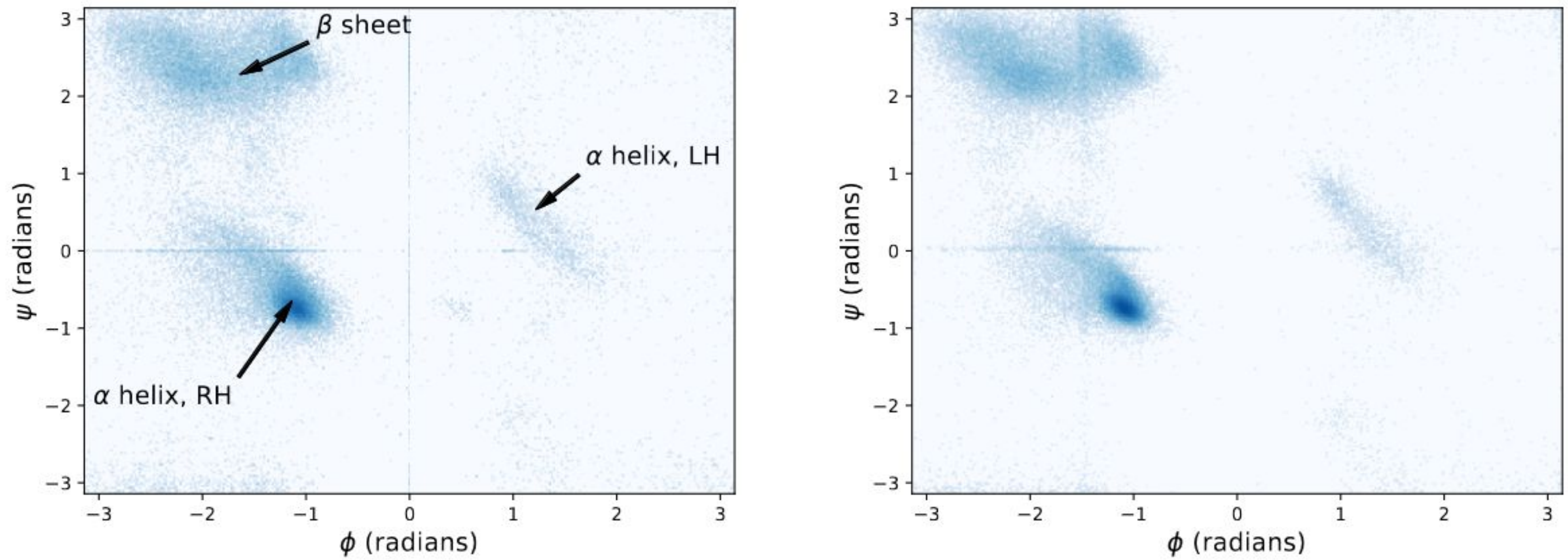
- CATH dataset
    - No two chains share more than 40% seq. Identity over 60% overlap
- Exclude chains < 40 residues
- Crops chains > 128 randomly to a 128-residue window
- Train/Val/Test size
    - 24316
    - 3039
    - 3040

Testing Results

- Reconstructed 10 backbones for every length L ∈ [50, 128)



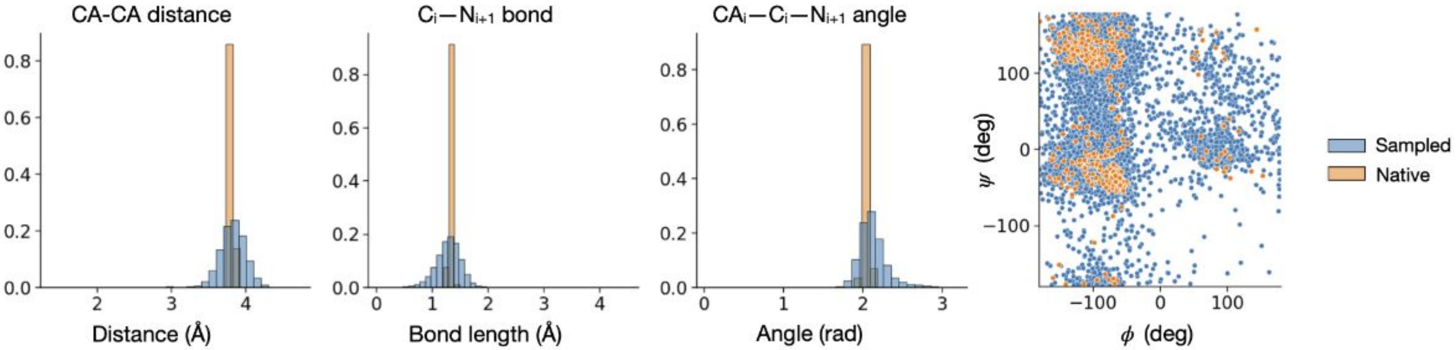
Ramachandran Plot Test



(a) Ramachandran plot, test set

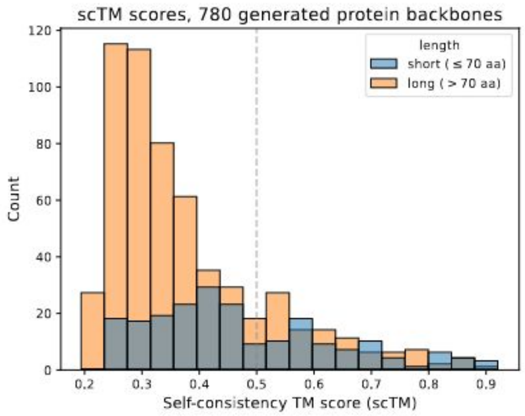
(b) Ramachandran plot, generated backbones

Comparison to Equivariant Diffusion Model (on coordinates)

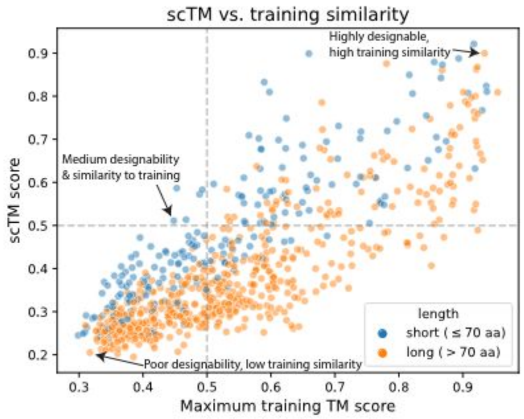


Designability of Predicted Backbones

- Generated 8 different AA sequences via ProteinMPNN
- Generated structures from these AA sequences with OmegaFold
  - Found 177/780 to be "designable" (163/780 when tested with AF2 (no-MSA))

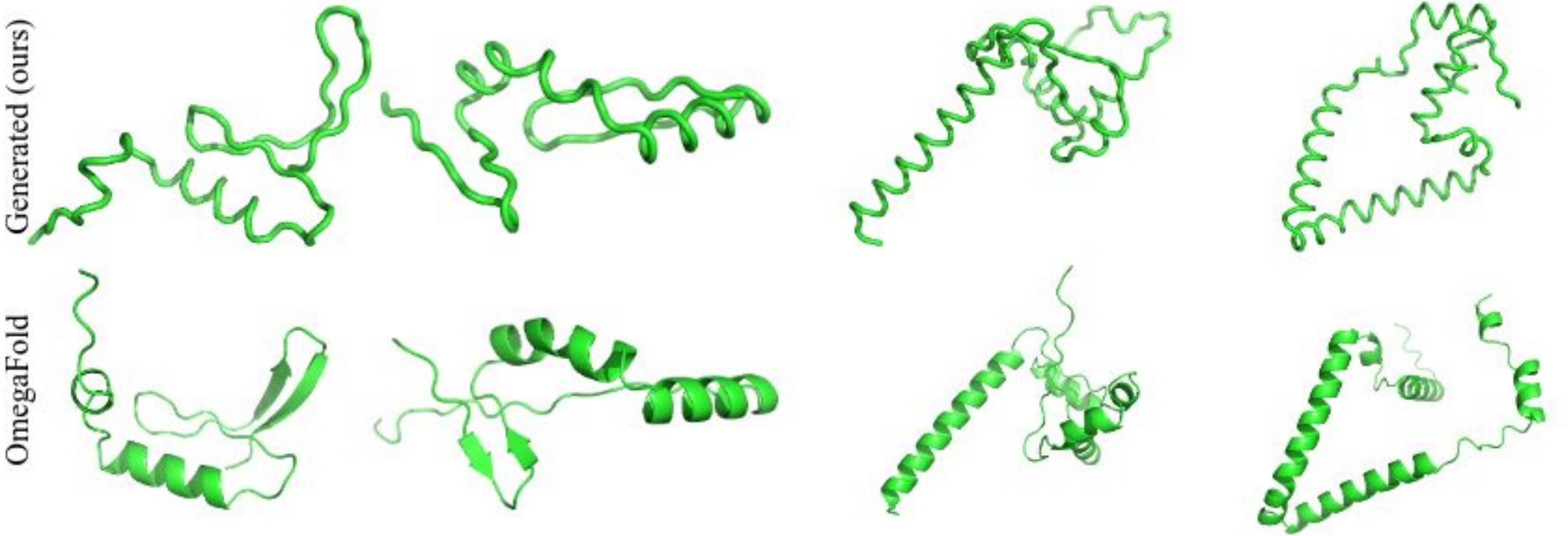


(a) Backbone designability by length



(b) Designability compared to training set similarity

Case Studies of Designability



59 residues, scTM = 0.59

61 residues, scTM = 0.51

107 residues, scTM = 0.53

111 residues, scTM = 0.55