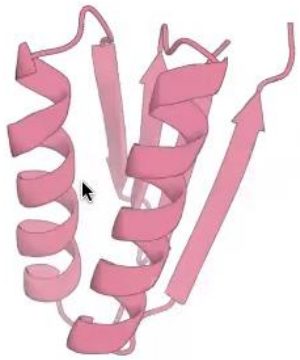
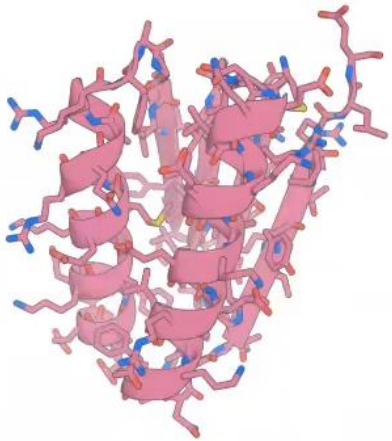
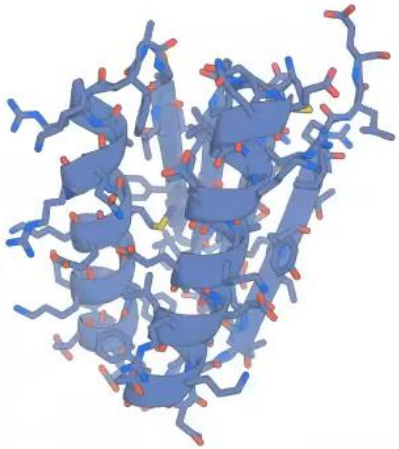
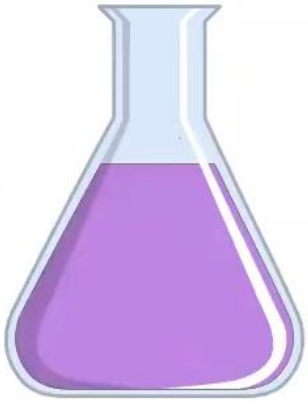
RF Diffusion

Rosetta Workshop
Yunchao (Lance) Liu
12.6.2023

RoseTTAFold Diffusion (RF Diffusion)

RF Diffusion is an AI-based tool for protein backbone design

Backbone Generation	Sequence Design	Computational Filtering	Experimental Characterisation
			
RF Diffusion	ProteinMPNN	AlphaFold2 RoseTTAFold ESMFold OmegaFold	~ 1 week

Outline of Talk

1. What is Diffusion
2. Intro to RF Diffusion
3. How to use RF Diffusion

Outline of Talk

1. What is Diffusion
2. Intro to RF Diffusion
3. How to use RF Diffusion

What is Diffusion Model

“Photo of a cat riding on a bicycle”

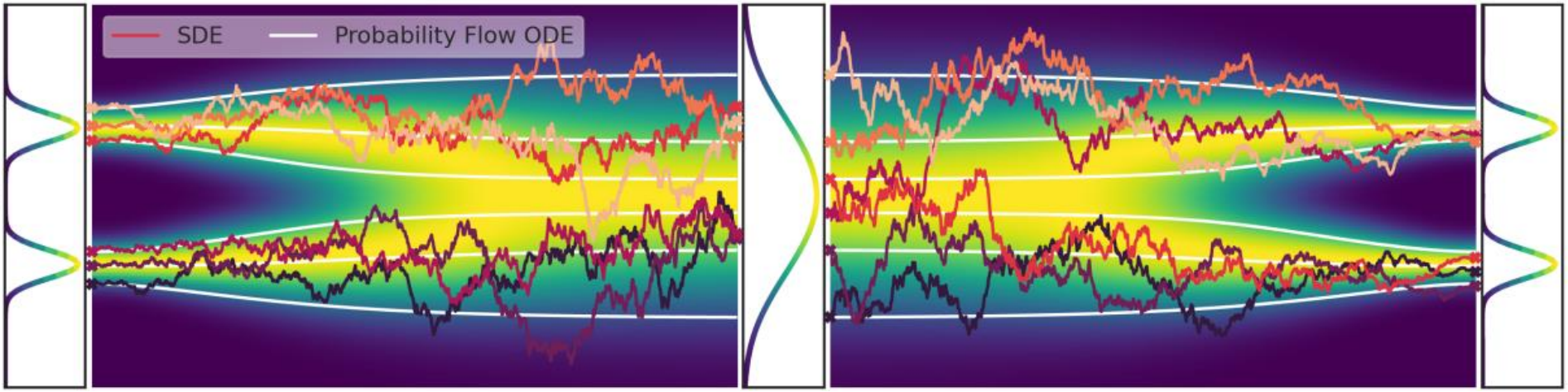
AI image generation tool example:

Stable Diffusion

DALL·E

How Does Diffusion Model Work

Diffusion Model learns to map random noise to data distribution



Forward Process

Reverse Process

Image example

Why Diffusion Model For Backbone Design

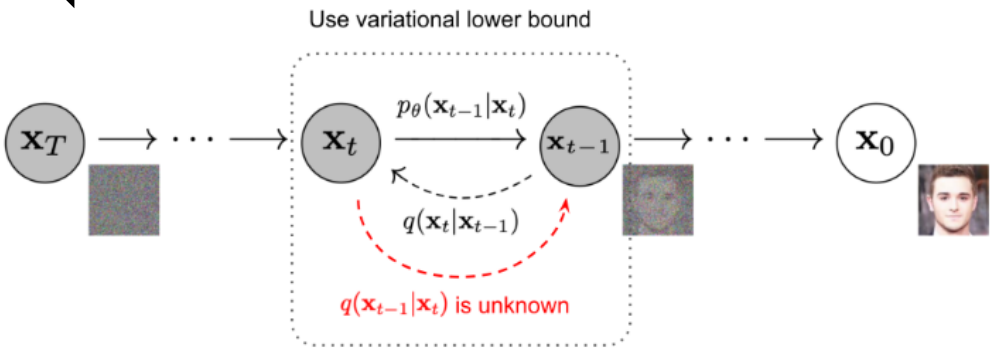
1. Generates highly diverse outputs that resemble training data
2. Can be guided toward design objective
3. Can operate directly on amino acid coordinates

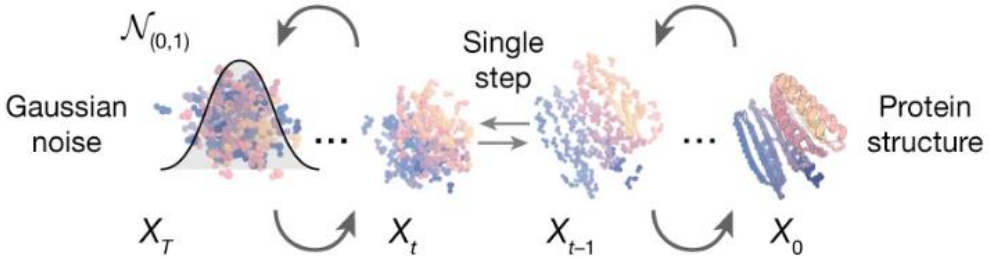
Outline of Talk

1. What is Diffusion
2. Intro to RF Diffusion
3. How to use RF Diffusion

How Does RF Diffusion Work

Forward Process





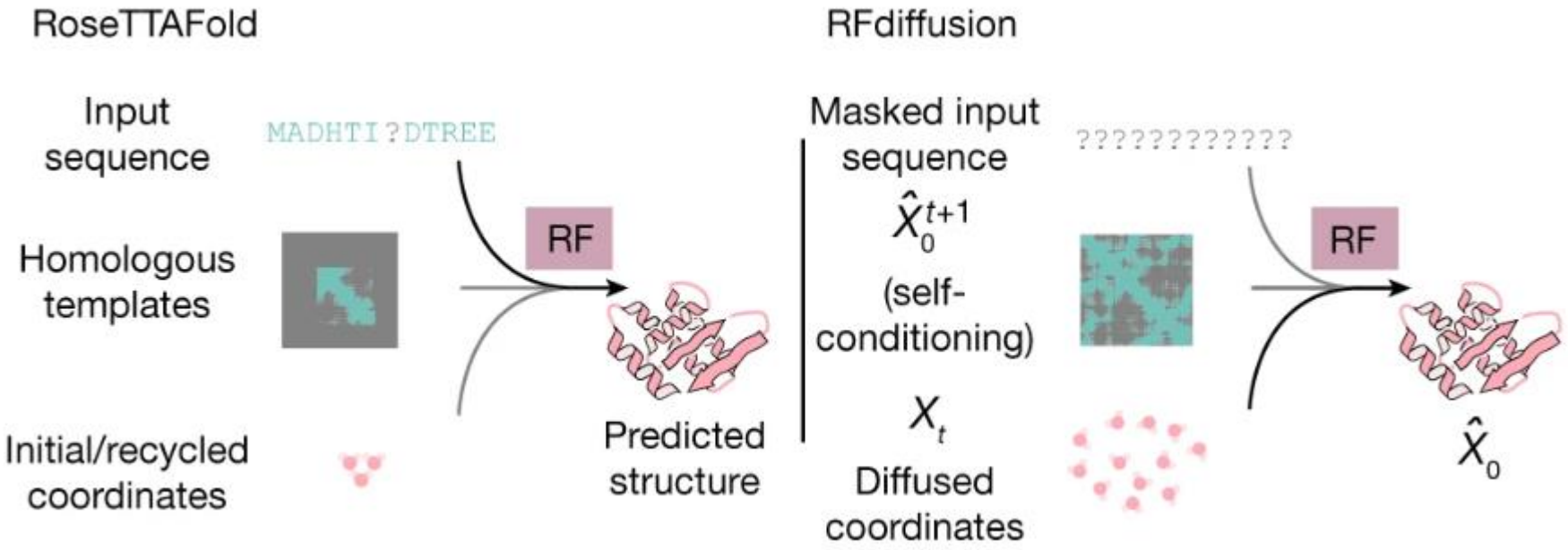
Reverse Process

Challenges of
Proteins vs Images

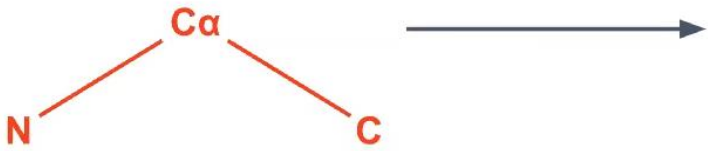
1. Strong geometric constraints
- 4 backbone heavy atoms
- 3 covalent bonds
- continuous chain

2. Must have a sequence that can encode it

How Does RF Diffusion Work

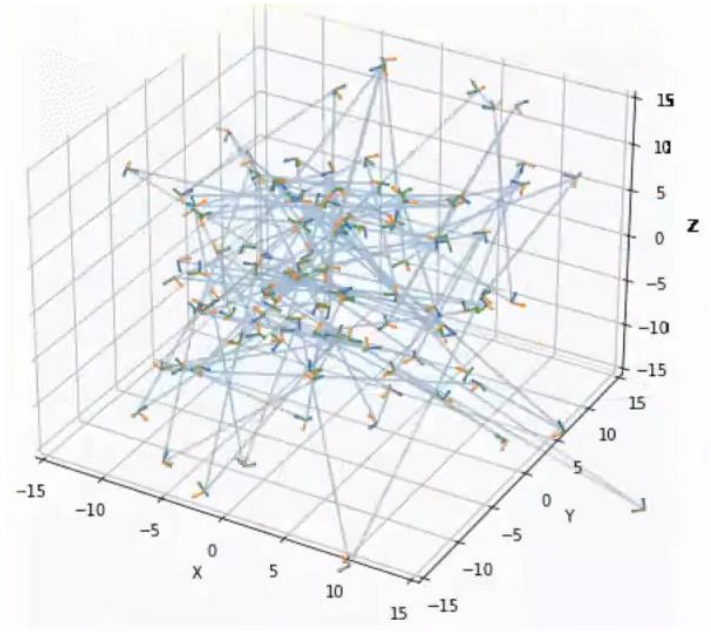


How Does RF Diffusion Work

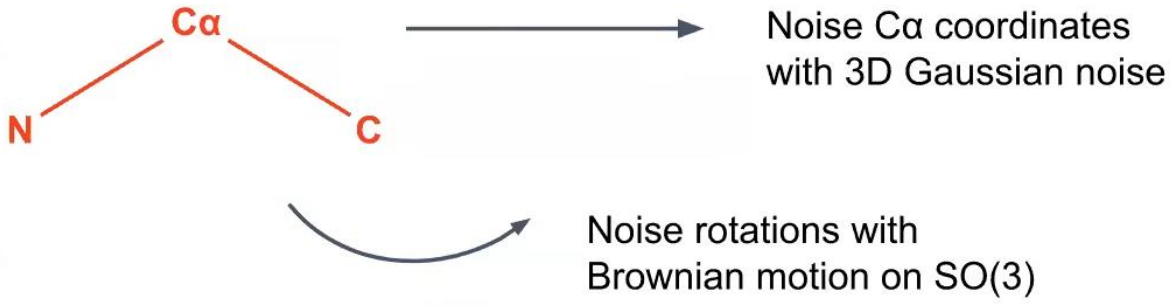


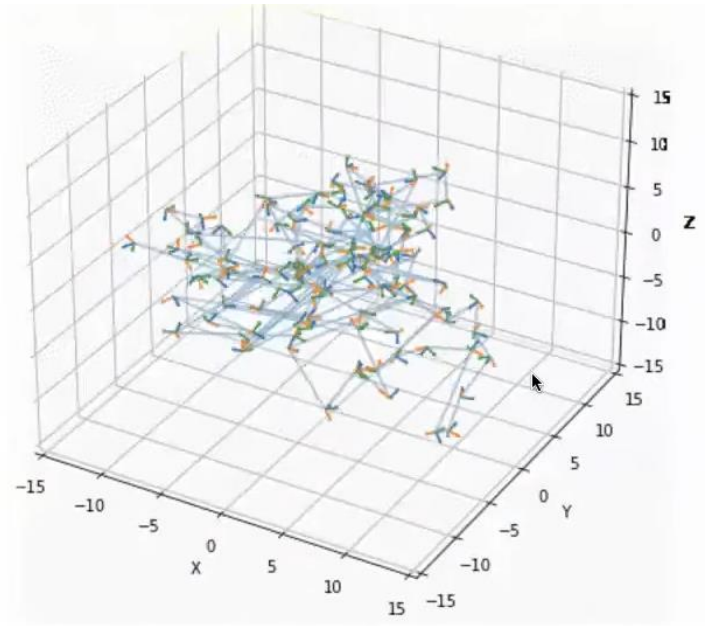
Noise Cα coordinates with 3D Gaussian noise

Noise rotations with Brownian motion on SO(3)

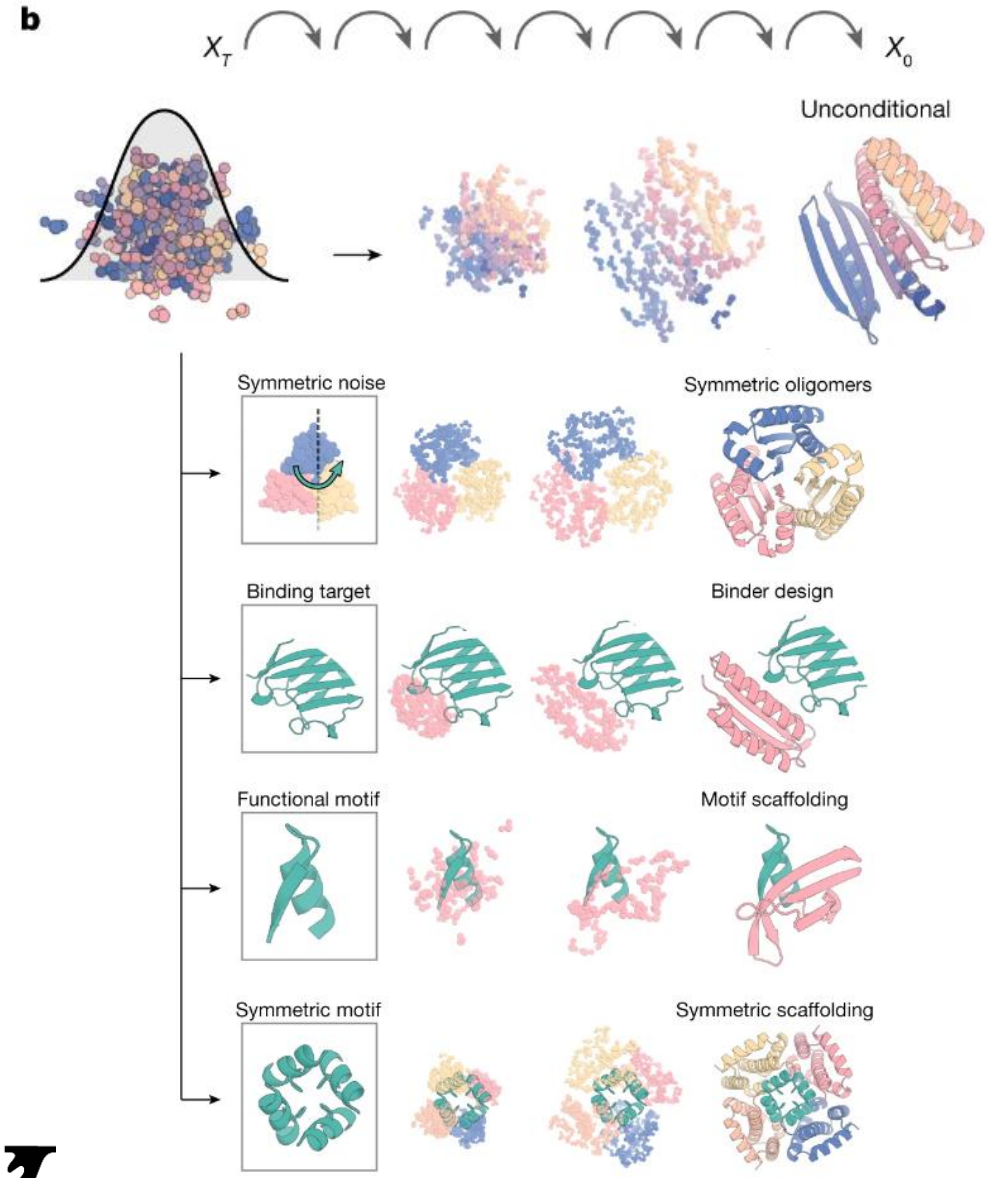


How Does RF Diffusion Work



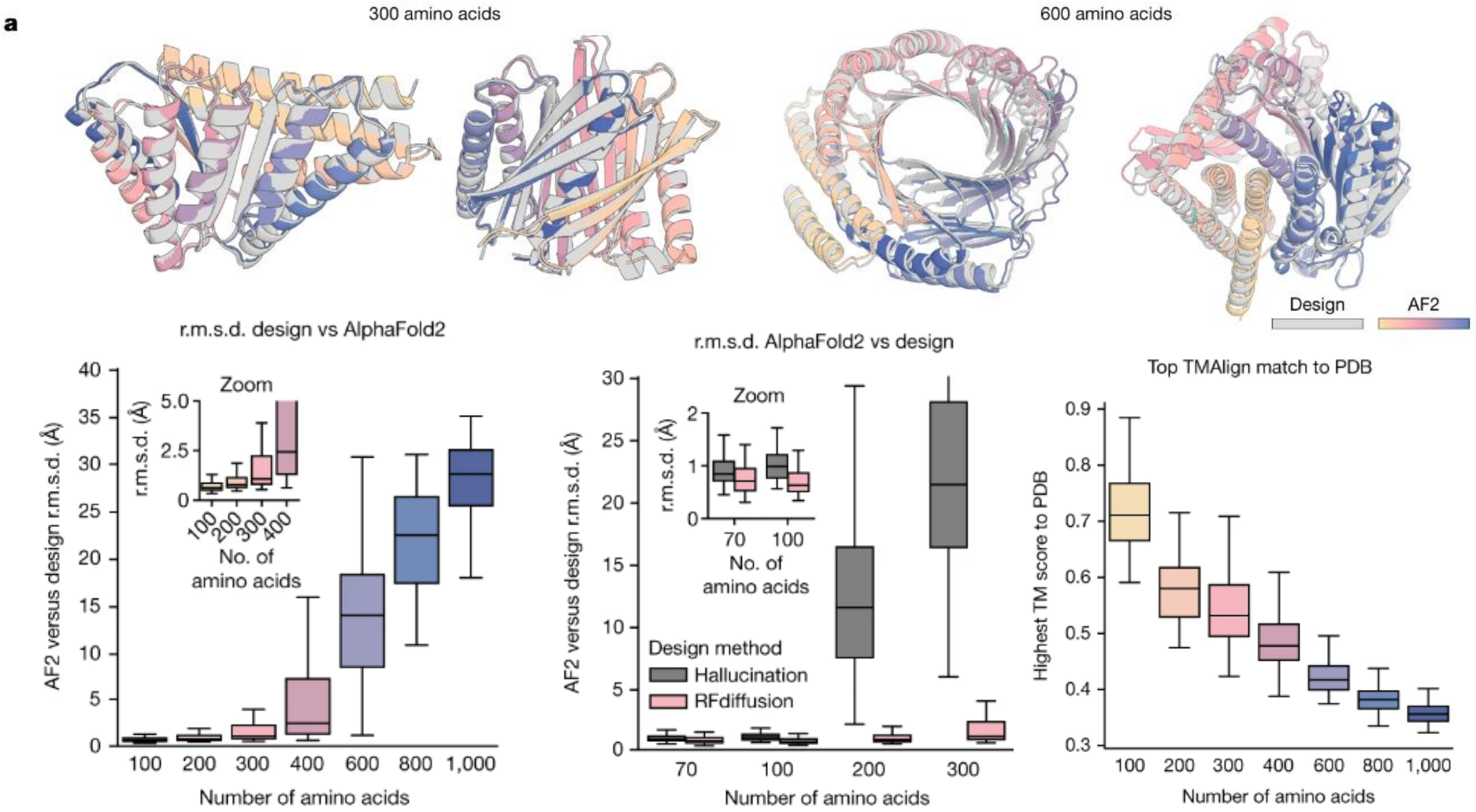


What Can RF Diffusion Do



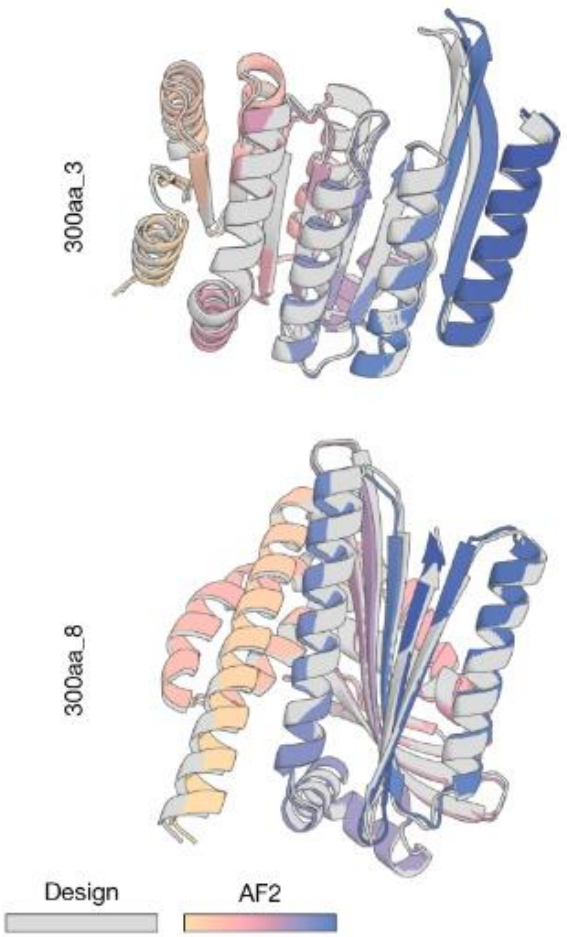
Monomer/Higher-order Oligomer
Unconditional/Conditional

Performance – Diverse and New-to-nature

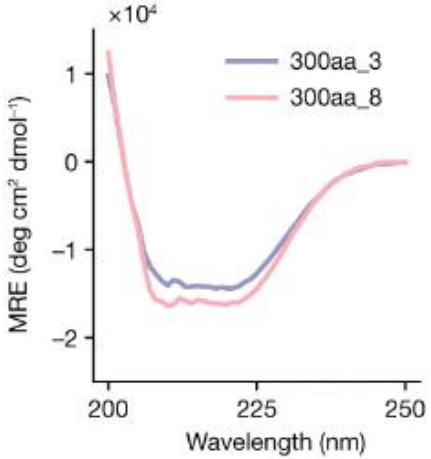


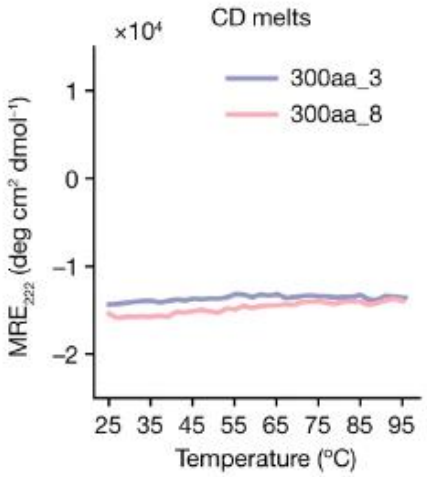
Experimental

Unconditional Design
Expressed in E.coli

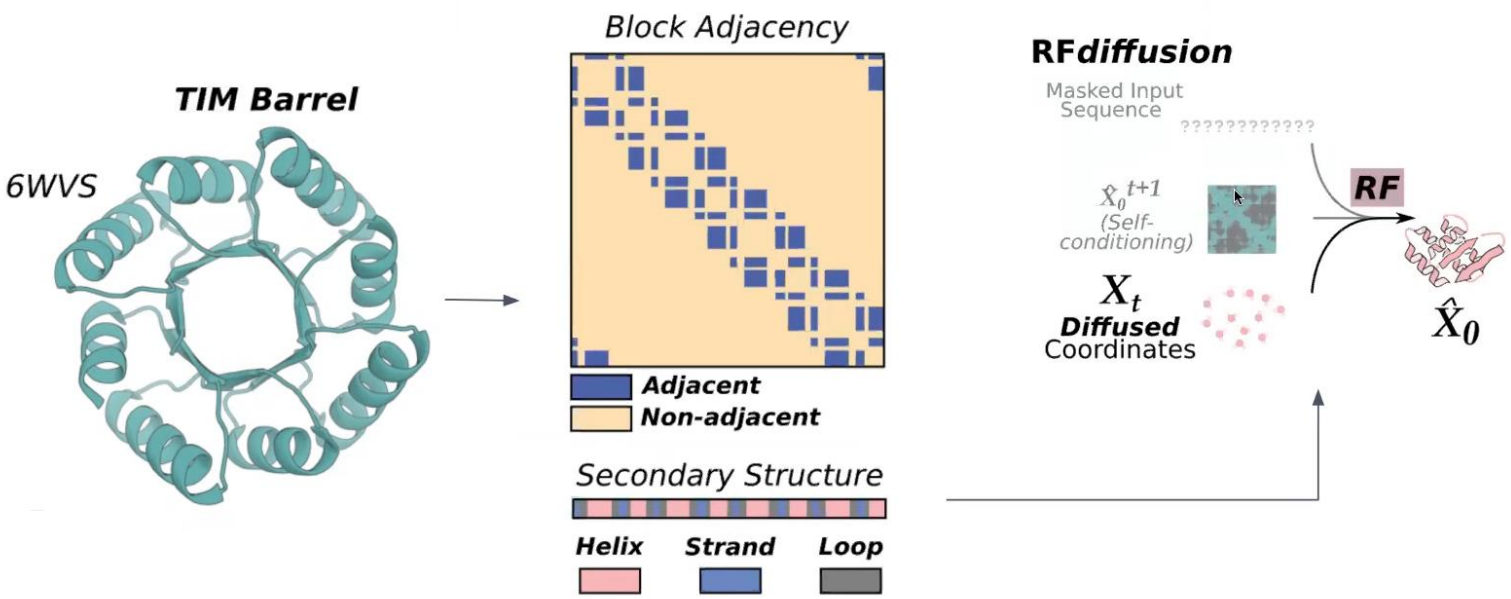


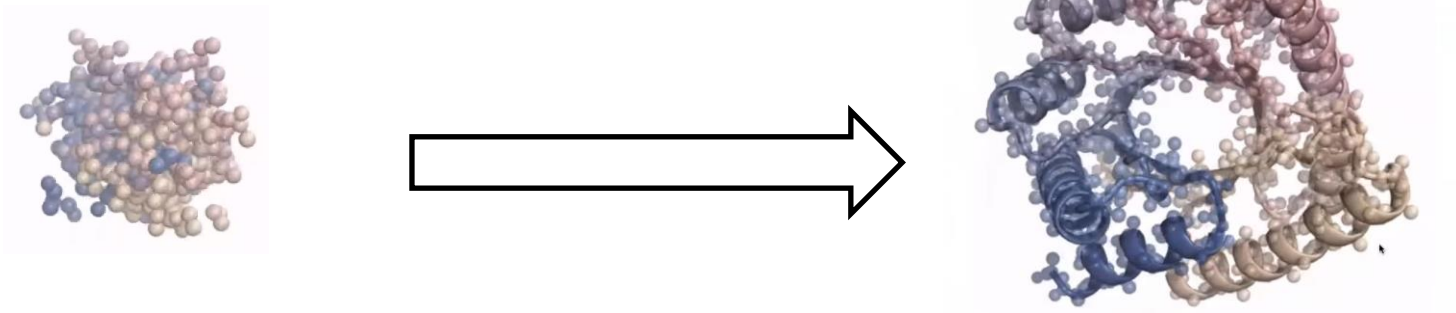
circular dichroism (CD) spectra



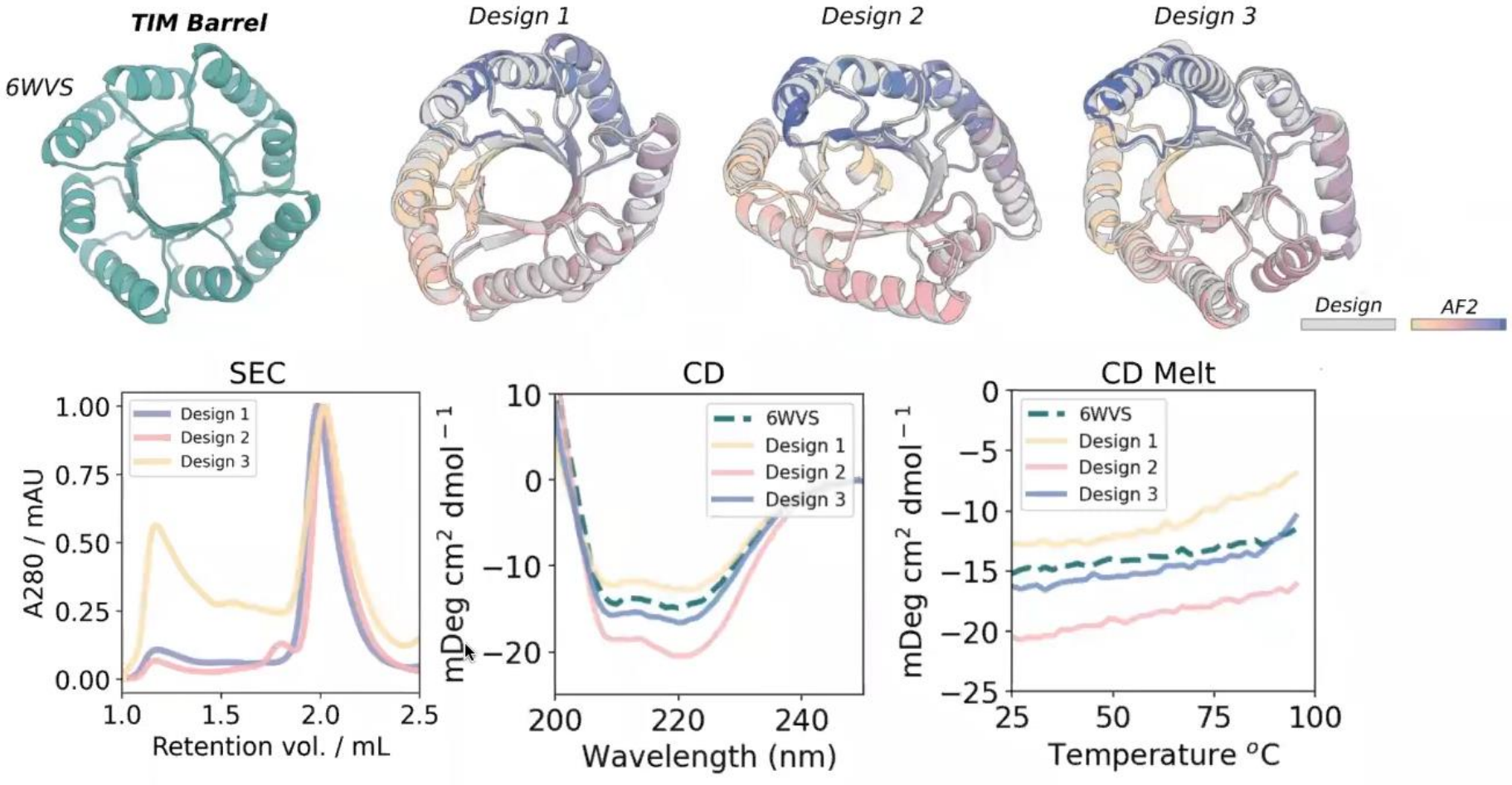


Conditional-Fold Family

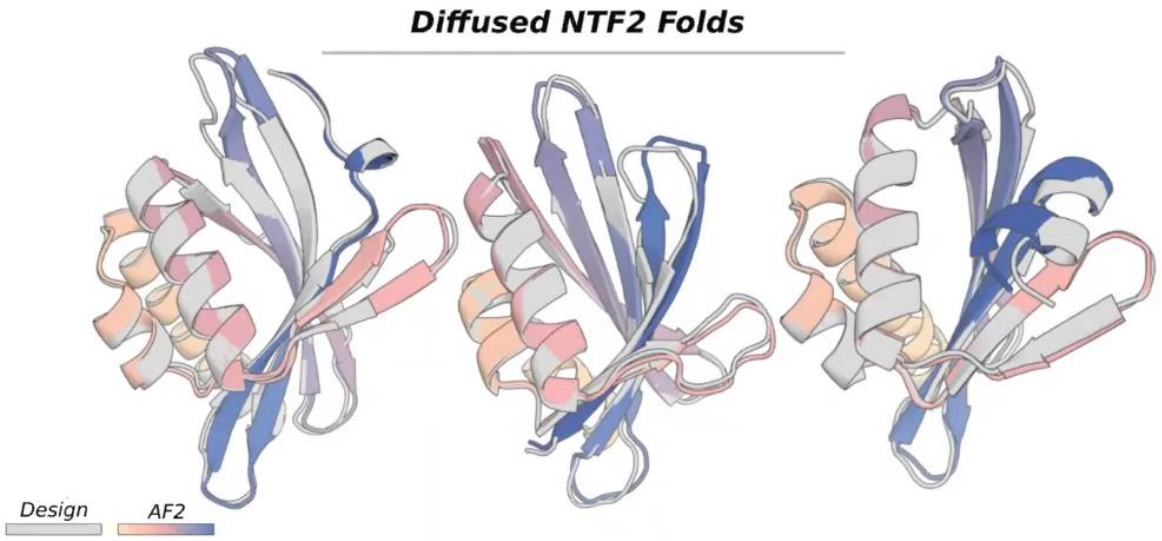


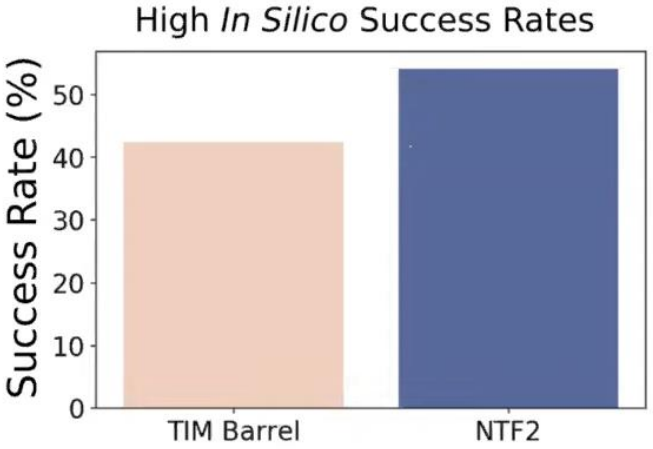


Conditional-Fold Family



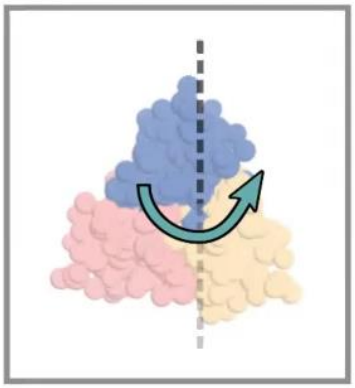
Conditional-Fold Family

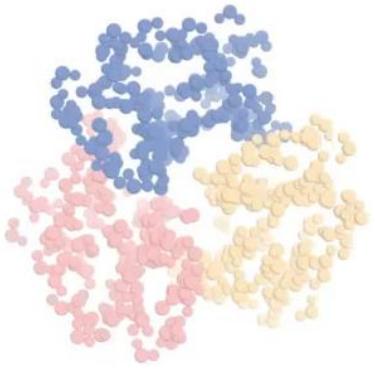




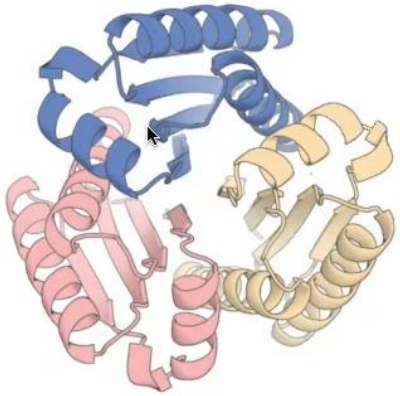
Symmetric Oligomers

Symmetric Noise

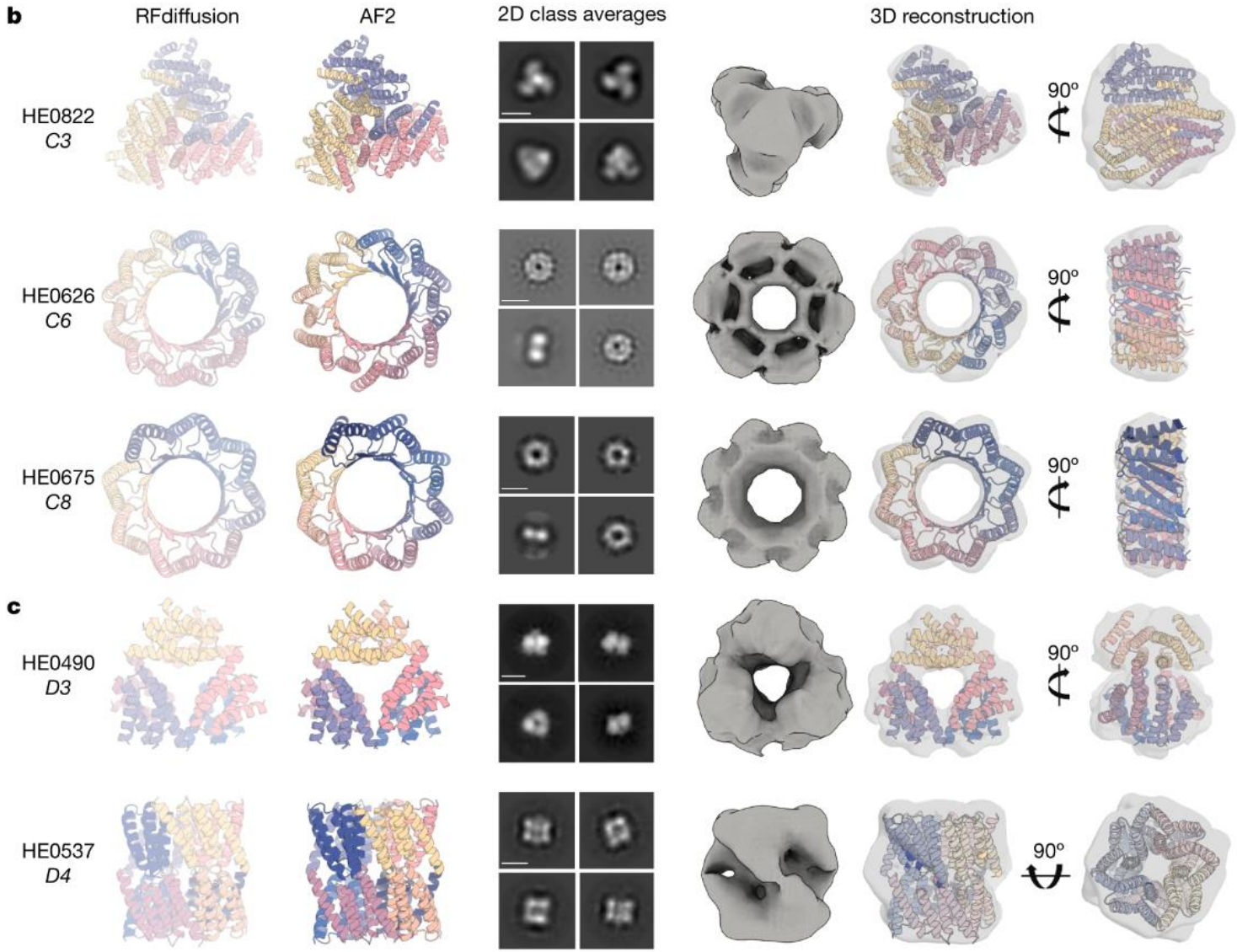




Symmetric Oligomers



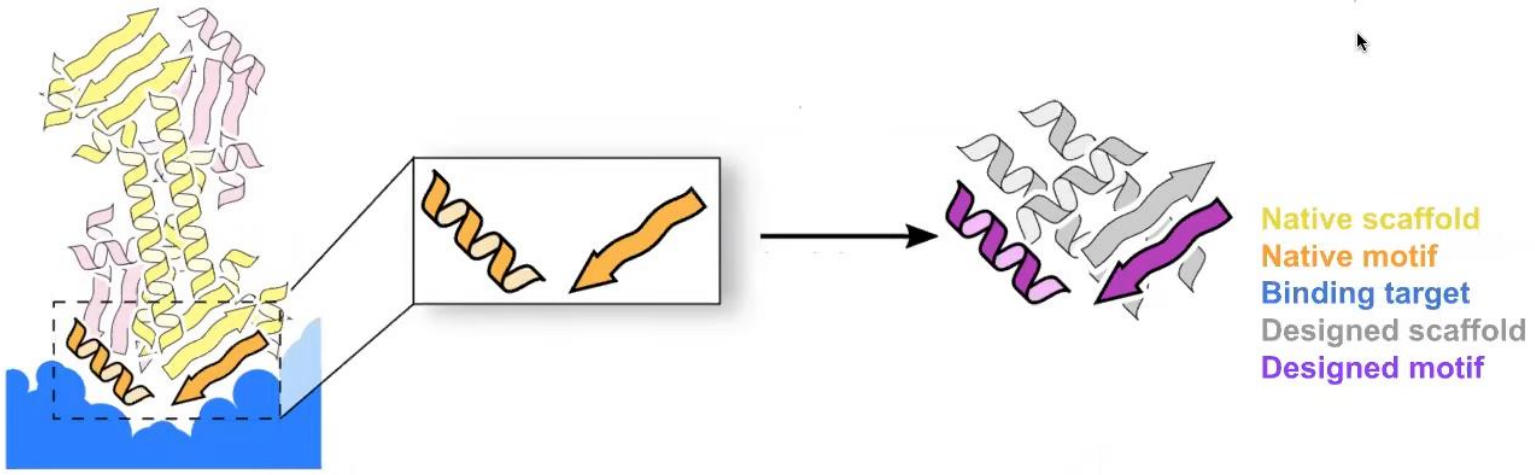
Experiment-Symmetric Oligomers



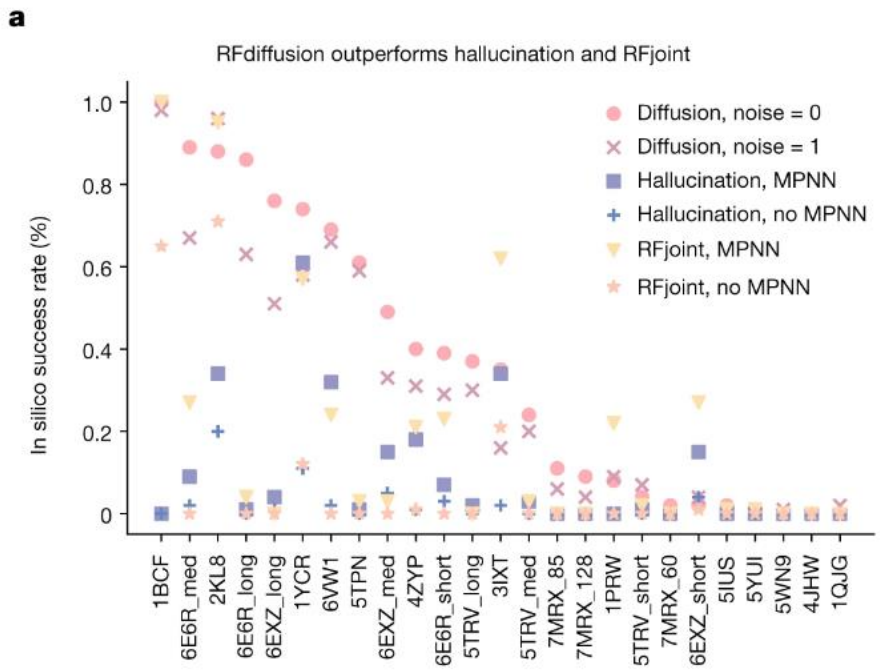
Conditional-Motif Scaffolding

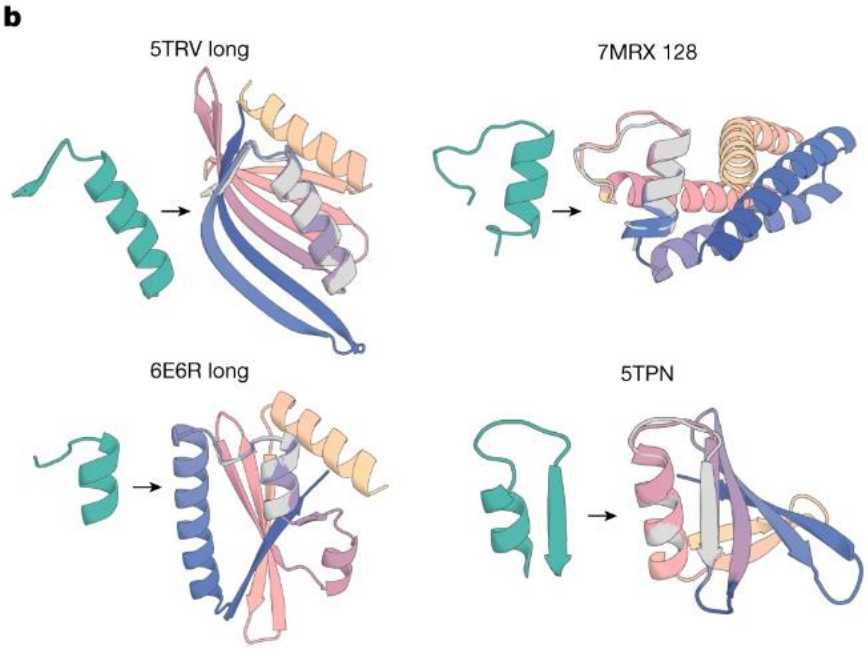
Given: functional “motif” - set of atoms/residues arranged in space, known or theorised to possess chemical functionality

Goal: Generate a protein sequence which, when folded, recapitulates the structure and dynamics of the original functional motif.

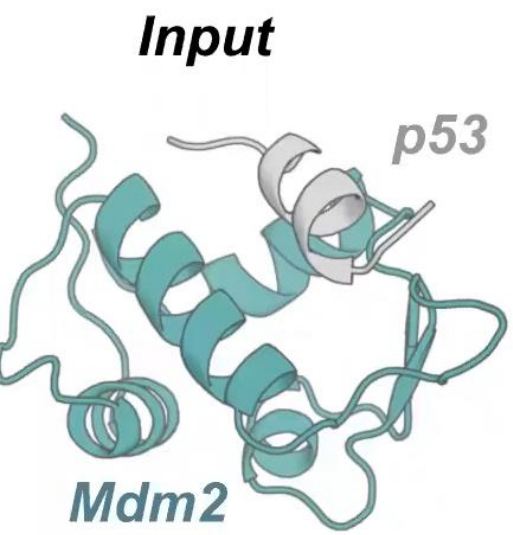


Conditional-Motif Scaffolding

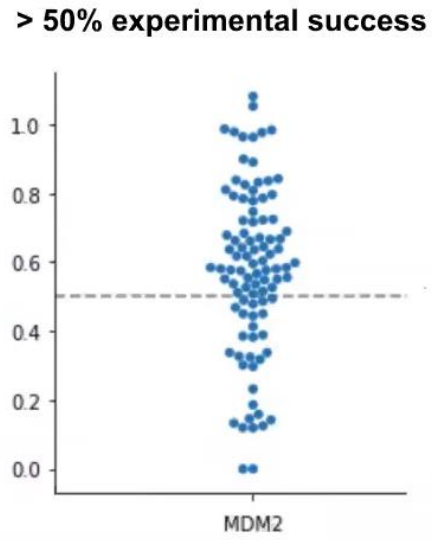


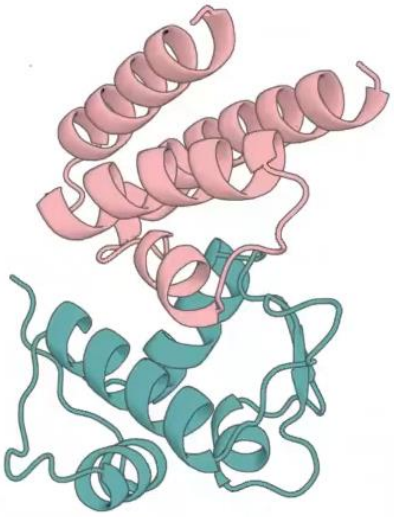


Conditional-Motif Scaffolding-Experiment

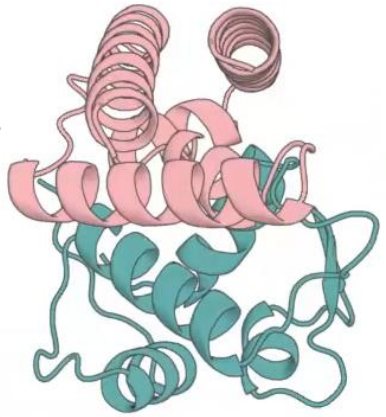


600nM affinity



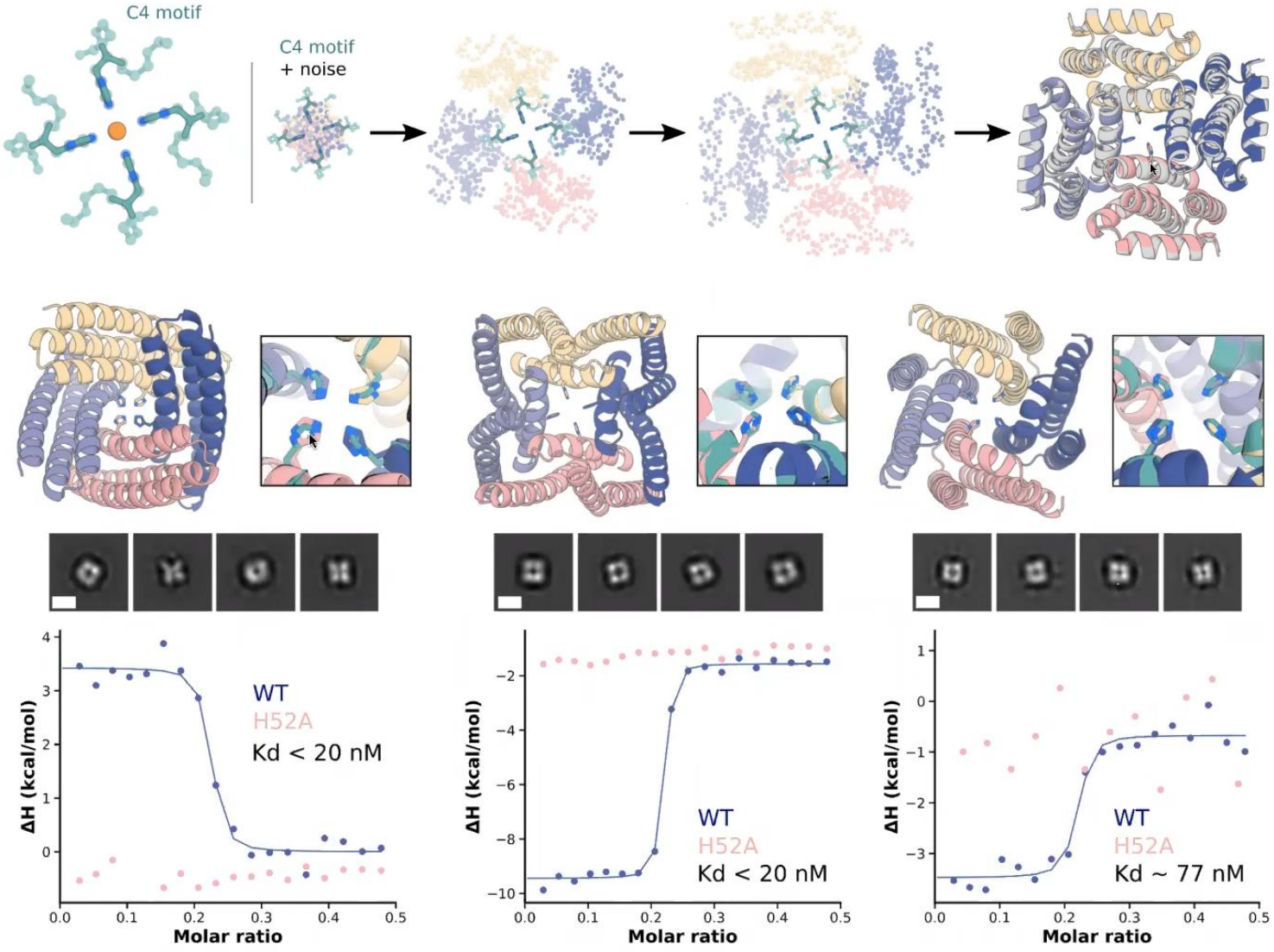


< 1nM



< 1nM

Symmetric Metal-Binding Oligomers



Outline of Talk

1. What is Diffusion
2. Intro to RF Diffusion
3. How to use RF Diffusion

How to Use RF Diffusion

Unconditional

- Specifiy length of the protein with **contigmap.contigs**

  Example: **‘contigmap.contigs = [150-150]’**

  Don’t forget the **‘’**

- Specify the output

  Example: **inference.output_prefix=path_to_output**

- Number of different backbones

  Example: **inference.num_designs = 5**

- Example command:

```
../RFdiffusion/scripts/run_inference.py \
'contigmap.contigs=[150-150]' \
inference.output_prefix=./output_monomer/monomer \
inference.num_designs=5
```

Examining the Output

Outputs are in **output_monomer** folder

All outputs are poly-Glycine sequence

- **PDB file**, containing generated backbone
- **.trb file**, contains all metadata with the run, including input options
- **traj folder**, contrains trajectory files (pdb)

Use pymol for visualization:

```
pymol monomer_0.pdb
```

Motif Scaffolding

- use **‘contigmap.contigs’** to specify the motifs.

  - Prefixed by a letter: is motif. e.g. A2-10, residue 2-10 from chain A
  - Not prefixed residue will be built

  Example:

  ‘contigmap.contigs=[10-20/E422-440/40-50]’

  Means to built 10-20 residues N-terminally of the motif, and then 422-440 from chain E of the input, followed by 40-50 residue

- use **‘inference.input_pdb=path’** to specify the input

  Example command:

```
../RFdiffusion/scripts/run_inference.py \
'contigmap.contigs=[20-30/E422-440/40-50]' \
inference.input_pdb=6UYG.pdb \
inference.output_prefix=./output_scaffold/scaffold \
inference.num_designs=1
```

Motif Scaffolding-No Clashing

- A more complicated case:

  Also input info of other chains avoid clashes：

  Use **/0** followed by a space

```
cd ../../
../RFdiffusion/scripts/run_inference.py \
'contigmap.contigs=[10-20/E422-440/40-50/0 H4-22/H29-71/H77-113/0
L2-29/L31-95/L97-109]' \
inference.input_pdb=6UYG.pdb \
inference.output_prefix=./output_scaffold/scaffold_Ab \
inference.num_designs=1
```

Heavy chain info

Light chain info

Acknowledgement

Dieter Hoffmann
Dr. Zhaoqian (Josh) Su
Dr. Rocco Moretti
Dr. Jens Meiler

References

Screenshots from

1. ML for protein engineering seminar series: https://www.youtube.com/watch?v=wIHwHDt2NoI
2. Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).
3. https://the-decoder.com/stable-diffusion-google-shows-new-method-for-more-control/
4. Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Questions?

Yunchao (Lance) Liu
刘运超

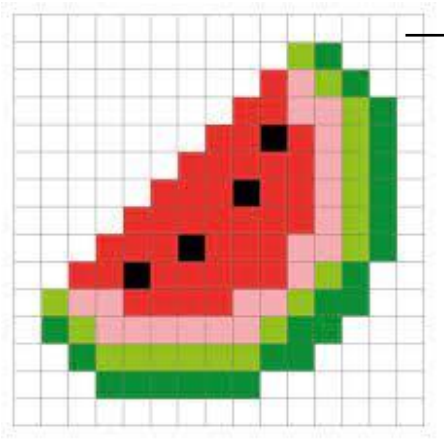
Research Interests

- AI Drug Discovery
- Geometric/Topological Deep Learning
- Generative Models
- Self-supervised Learning

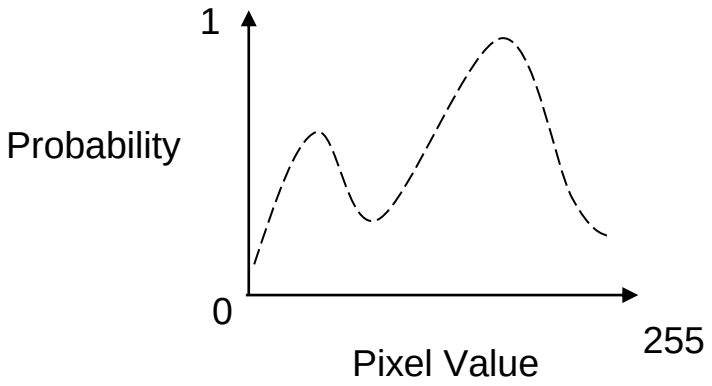
www.LiuYunchao.com

Supplement

High-dimensional Data Distribution

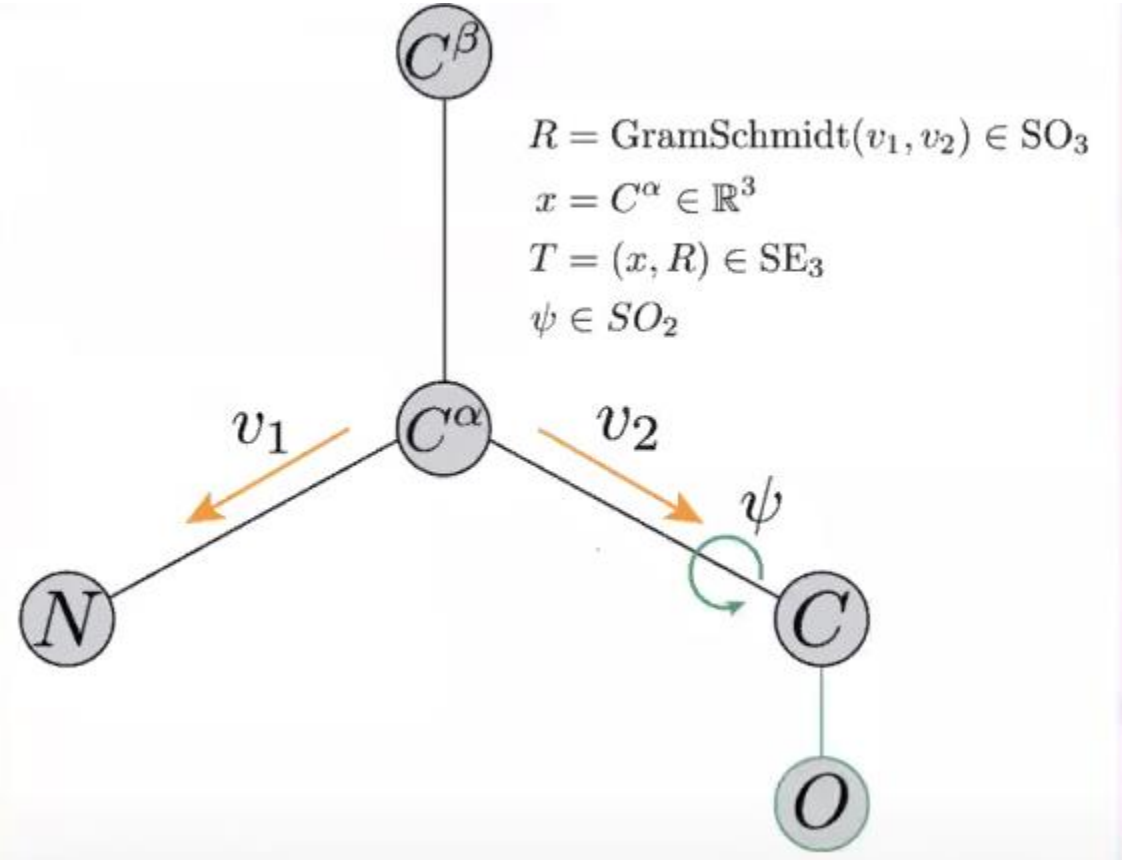


For a grayscale image,
each pixel has a value ranging from 0-255

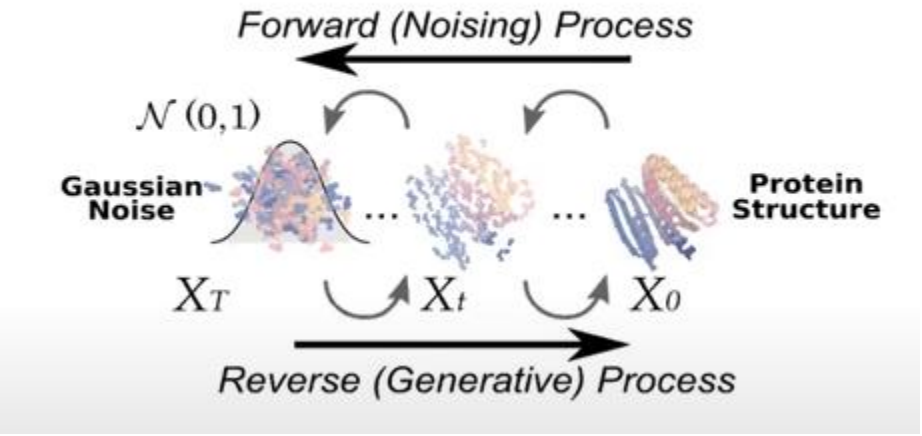


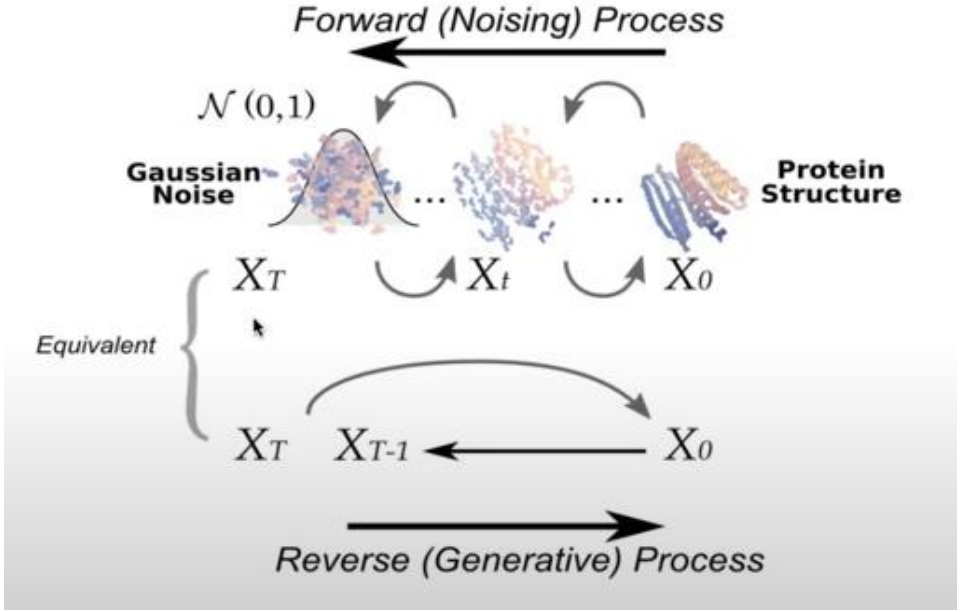
Essentially the image data distribution lives in a high-dimensional space

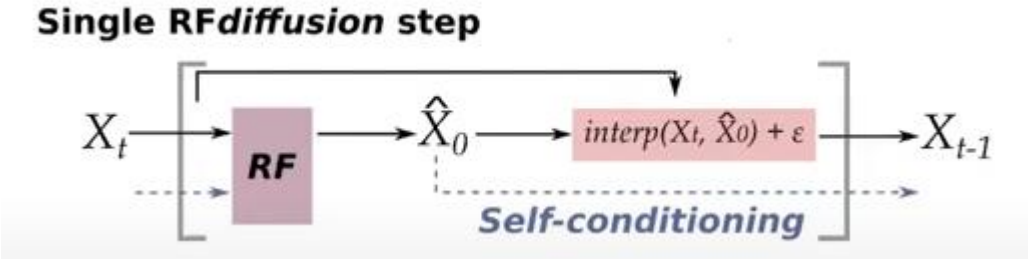
Each Residue Is Represented as A Translation (Cα coordinates) and Rotation



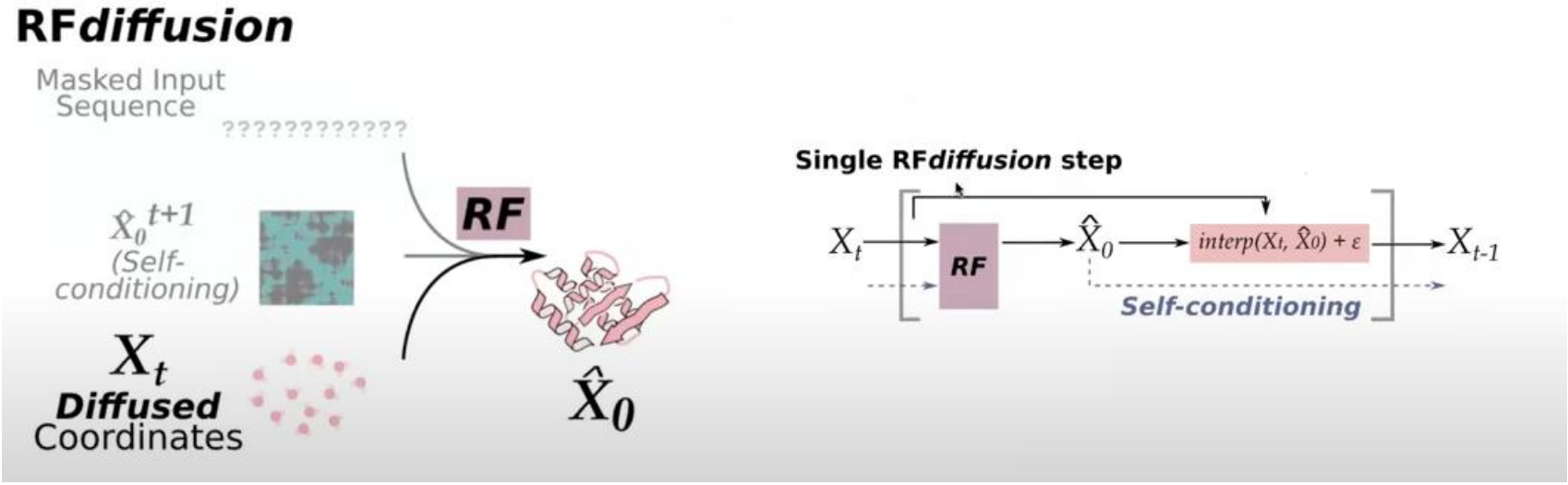
How to Denoise

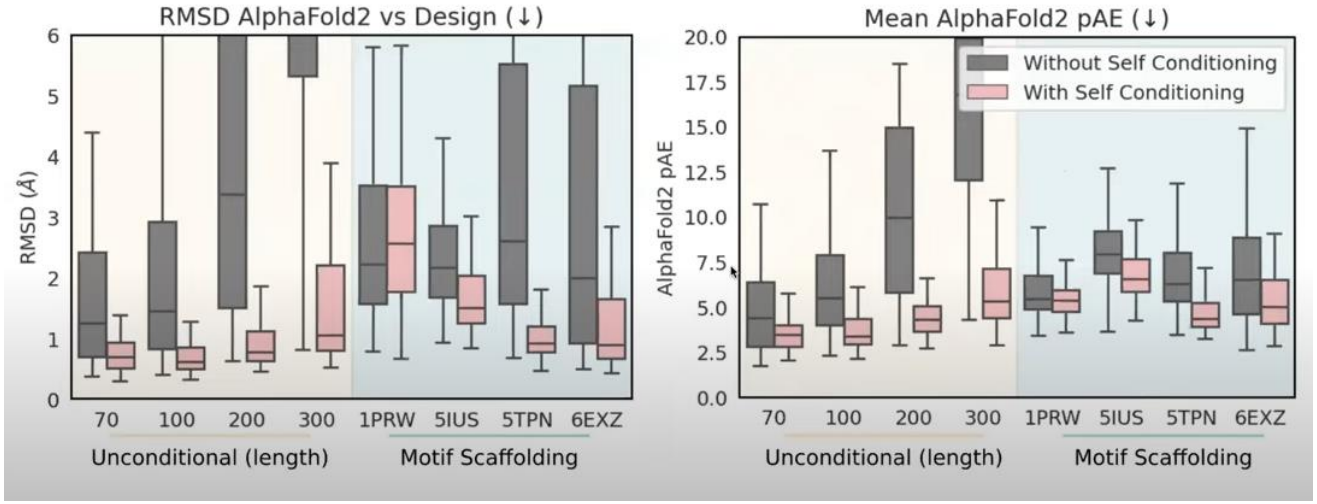




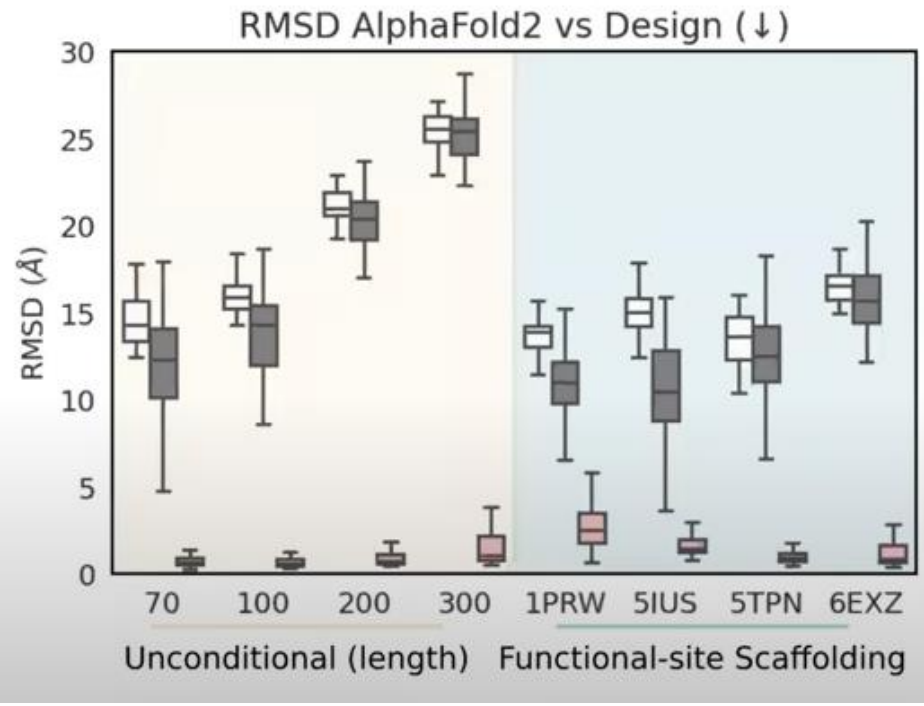


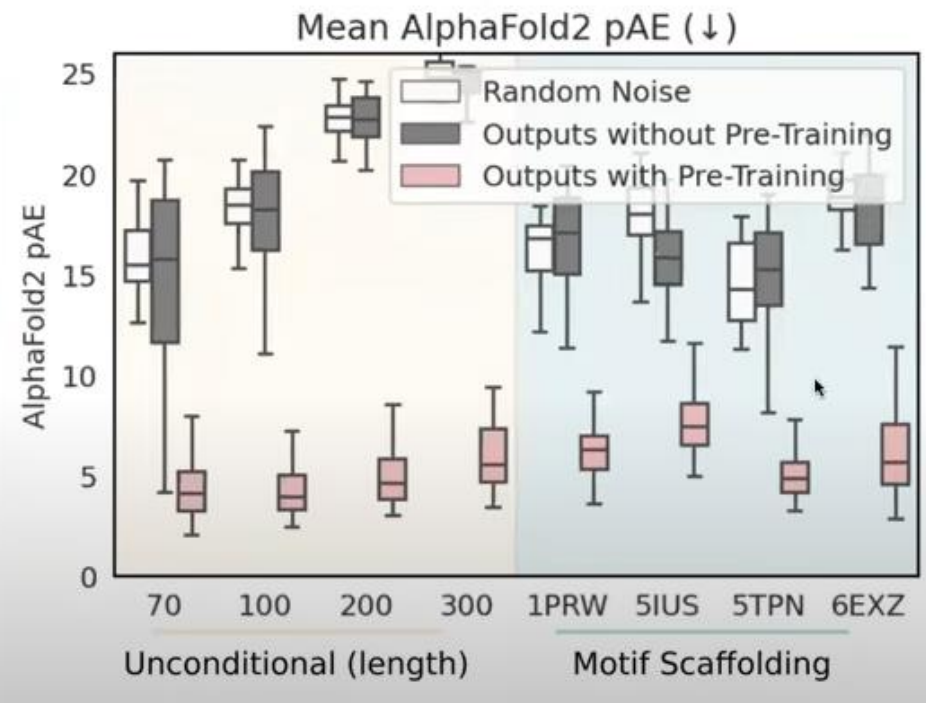
Self-Conditioning





Pretrained RoseTTAFold Weights Make Training Computationally Tractable





Pre-training and Self-conditioning