

# Diffusion models for protein structure and *de novo* design

Jason Yim



JAMEEL  
CLINIC



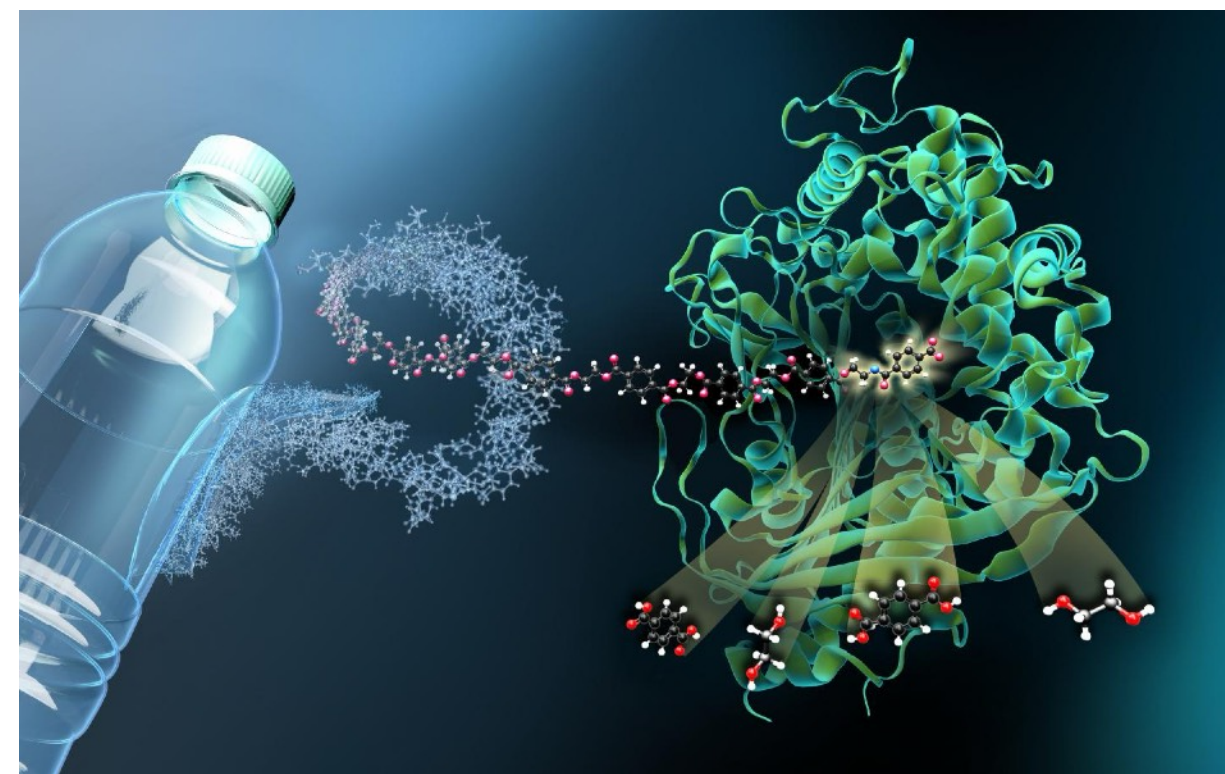
# Motivation

- **Nature** has produced many proteins to support life.
- **Evolution** does not work fast enough for new challenges.
- **(De novo) protein design** aims to *directly* engineer new proteins.



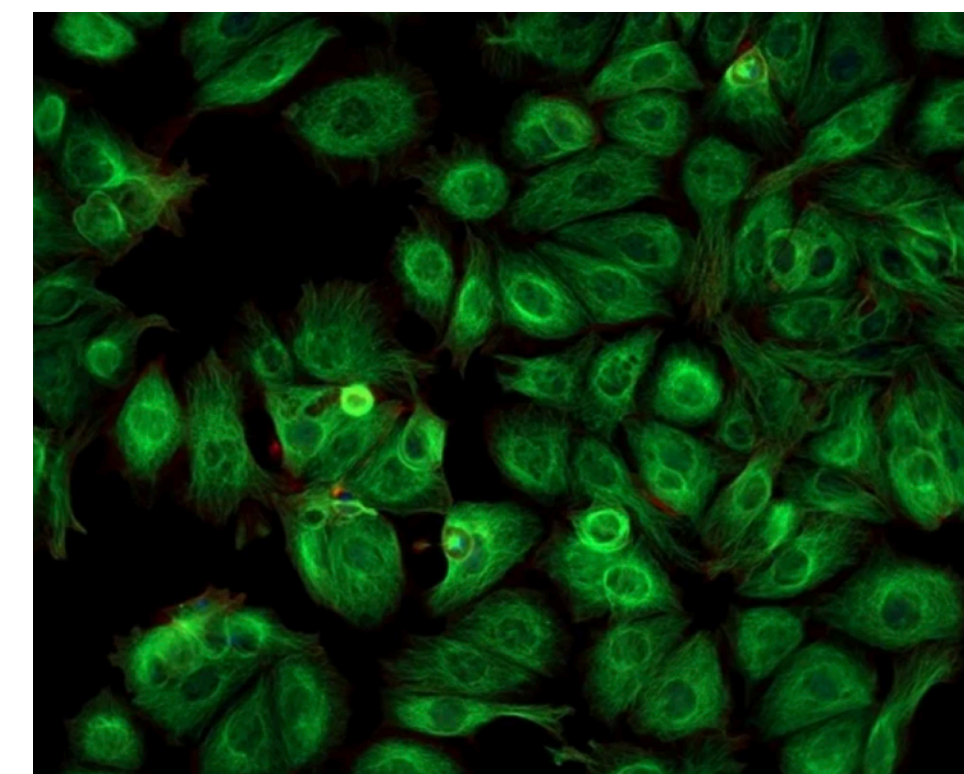
Vaccine & drug development

Image: Marsbars via iStock



Plastic degrading enzymes

Image: Martin Künsting/HZB



Fluorescence biosensors

Image: Caleb Foster via Shutterstock

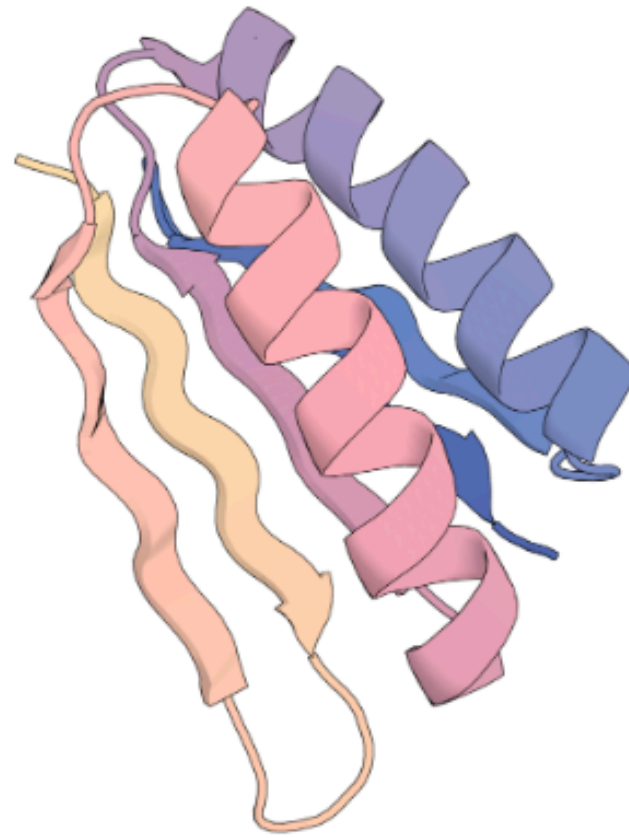


Genome editing

Image: Amanda Heidt via The Scientist

# AI-assisted protein design

1. Backbone generation



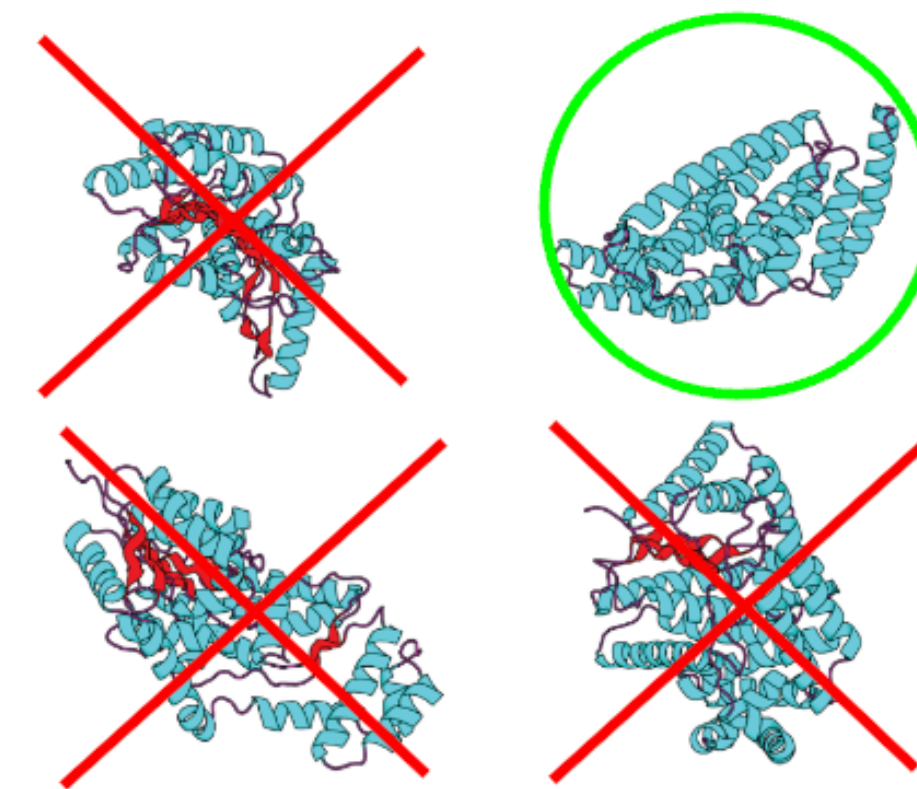
**Method:** None!

2. Sequence generation



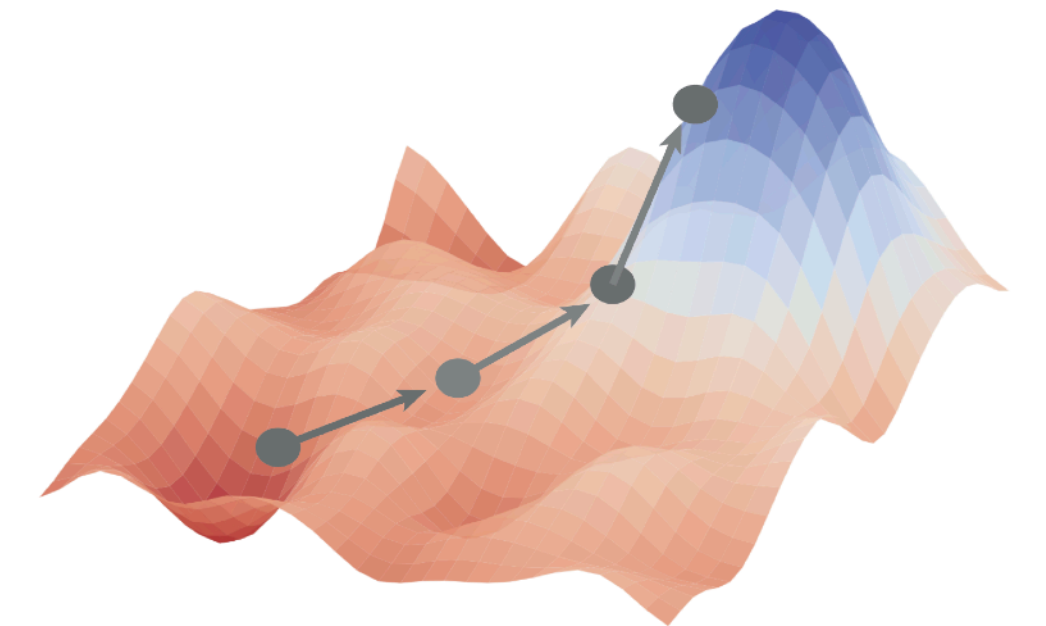
ProteinMPNN

3. Virtual screening



Protein folding

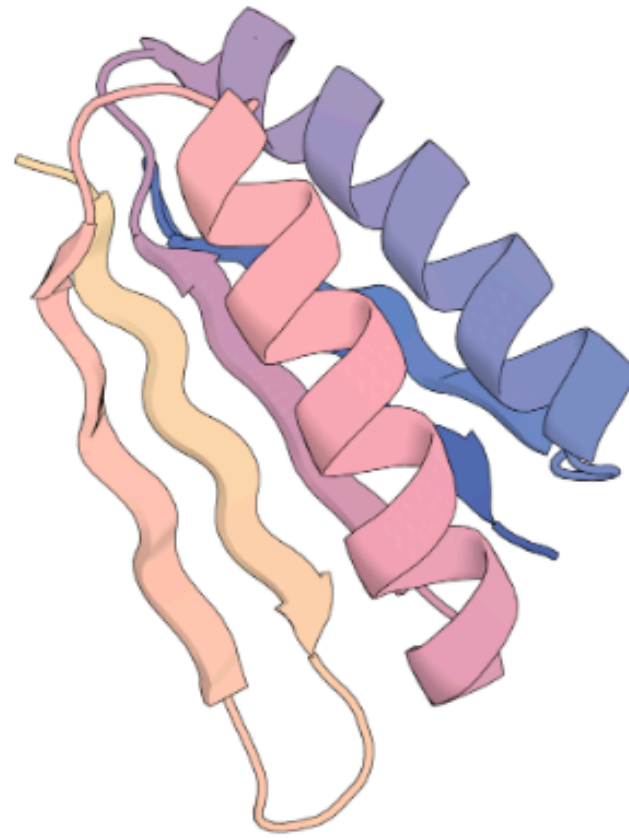
4. Property/fitness optimization



Directed evolution

# Probabilistic viewpoint

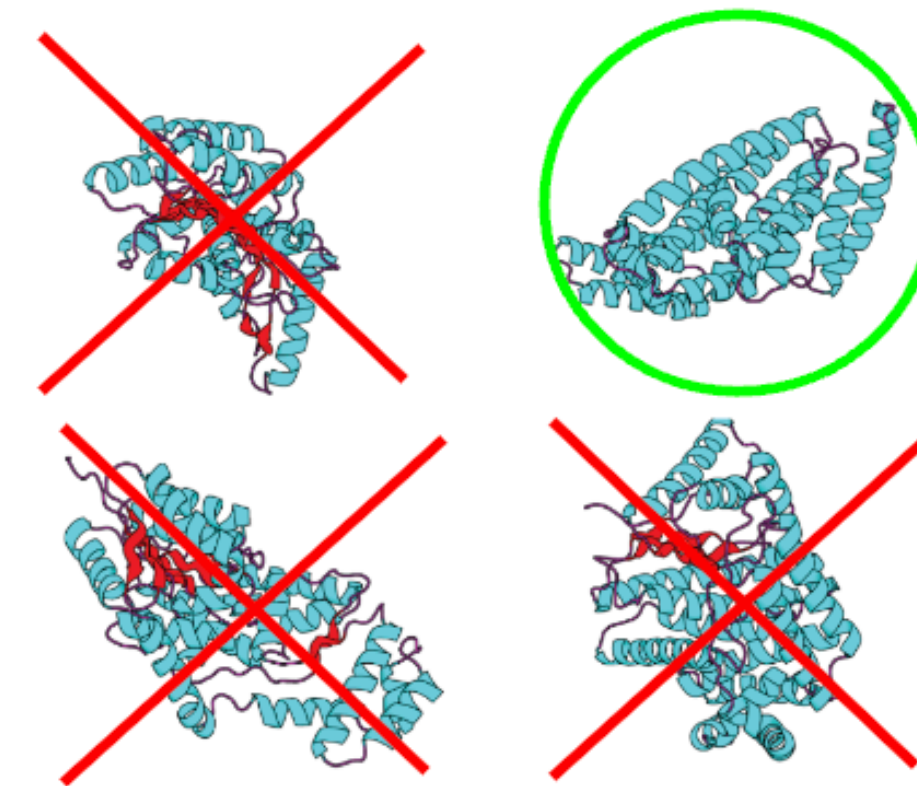
## 1. Backbone generation



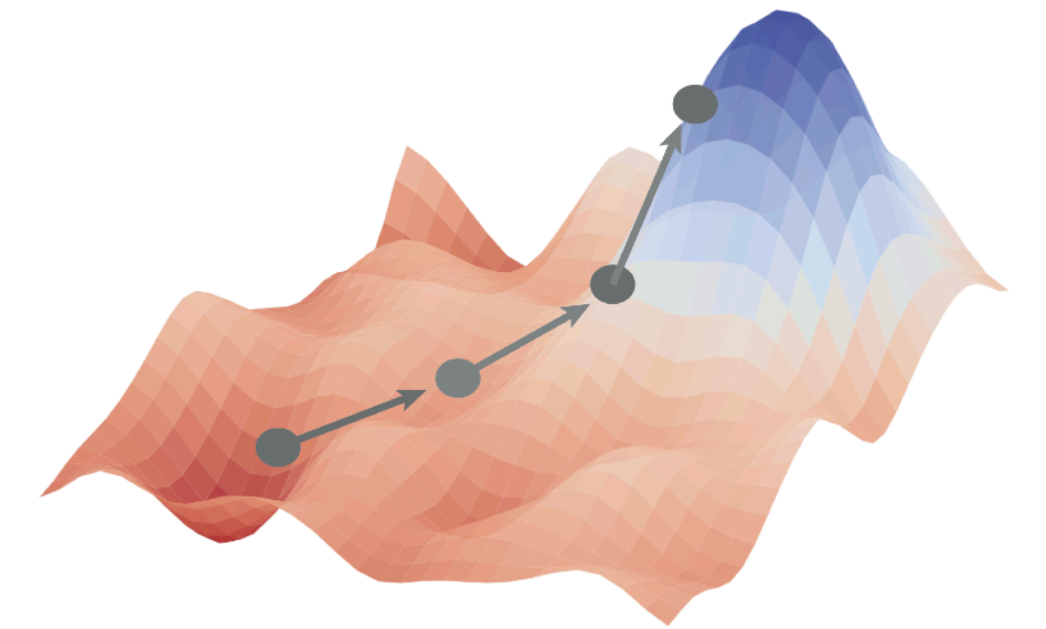
## 2. Sequence generation



## 3. Virtual screening



## 4. Property/fitness optimization



**Notation:** **bb:** backbone

**seq:** sequence

**des:** designability

**fit:** fitness

**Distribution:**  $p(\text{bb})$

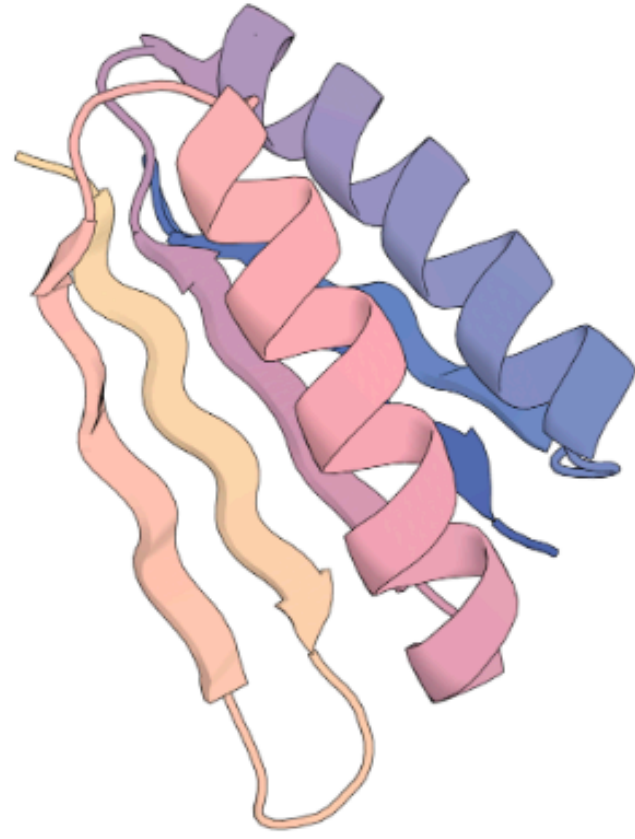
$p(\text{seq} | \text{bb})$

$p(\text{des} | \text{seq}, \text{bb})$

$p(\text{seq}; \text{fit}) \propto e^{\text{fit}(\text{seq})}$

# This talk

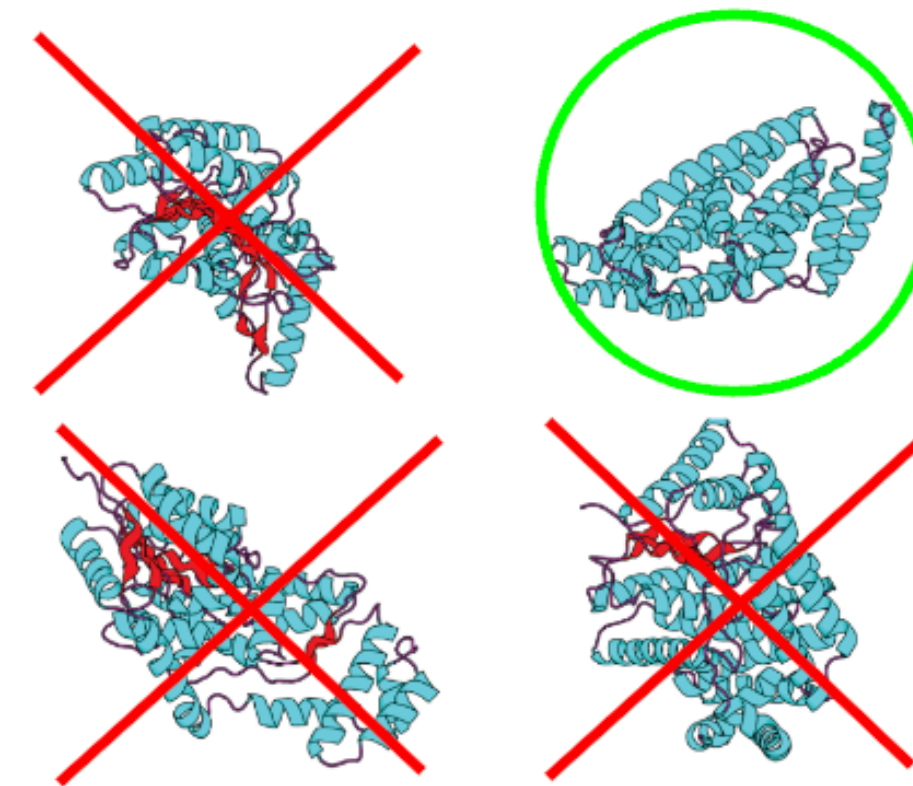
## 1. Backbone generation



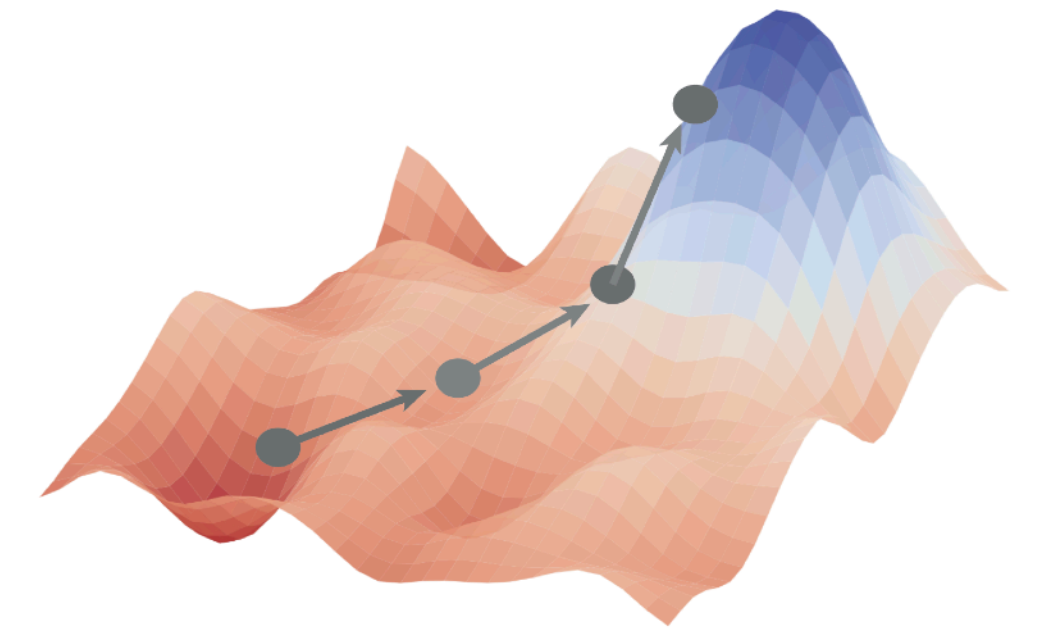
## 2. Sequence generation



## 3. Virtual screening



## 4. Property/fitness optimization



**Notation:** **bb:** backbone

**seq:** sequence

**des:** designability

**fit:** fitness

**Distribution:**  $p(\text{bb})$

$p(\text{seq} | \text{bb})$

$p(\text{des} | \text{seq}, \text{bb})$

$p(\text{seq}; \text{fit}) \propto e^{\text{fit}(\text{seq})}$

# Probabilistic protein design

## Why generative models

- **Exploration (sampling):** ability to generate *multiple* designs.
- **Composition:** combine and reuse models.
- **Uncertainty:** calibrate models with biological noise (i.e. evolution, experiments).

## Talk Outline:

	Method	Generative Model
1. FrameDiff	SE(3) diffusion	$p(\text{bb})$
2. RFdiffusion	SE(3) diffusion + RosettaFold2	$p(\text{bb}   \text{property})$
3. FrameFlow	SE(3) flow matching	$p(\text{bb})$

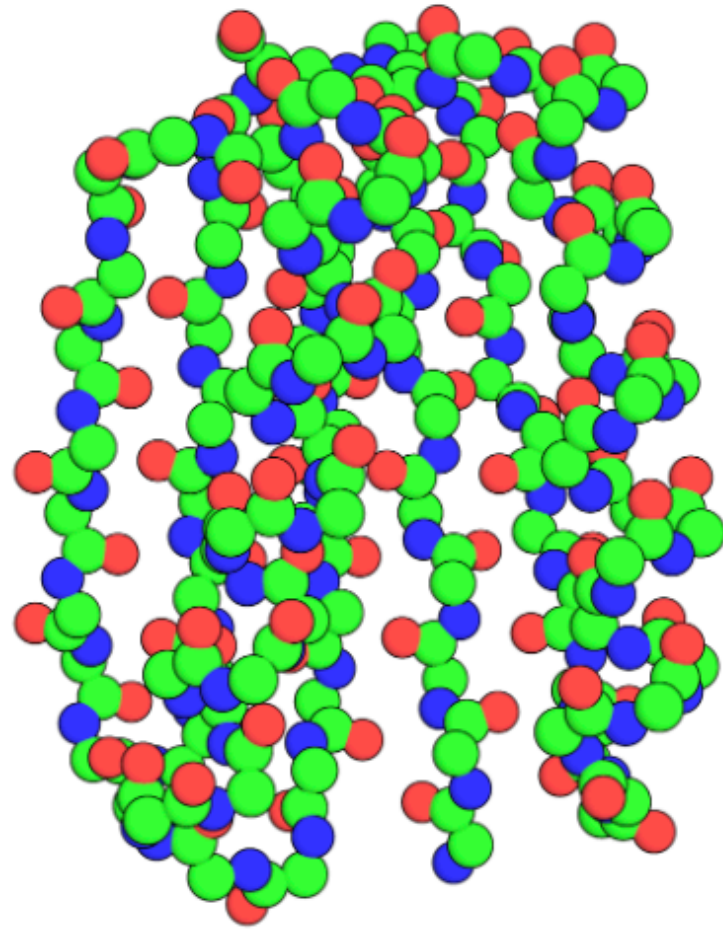
# FrameDiff

**Paper: SE(3) diffusion model with application to protein backbone generation**

**Jason Yim\*, Brian Trippe\*, Valentin De Bortoli\*, Emile Mathieu\*,  
Arnaud Doucet, Regina Barzilay, Tommi Jaakkola**

**ICML 2023**

# Structure representations

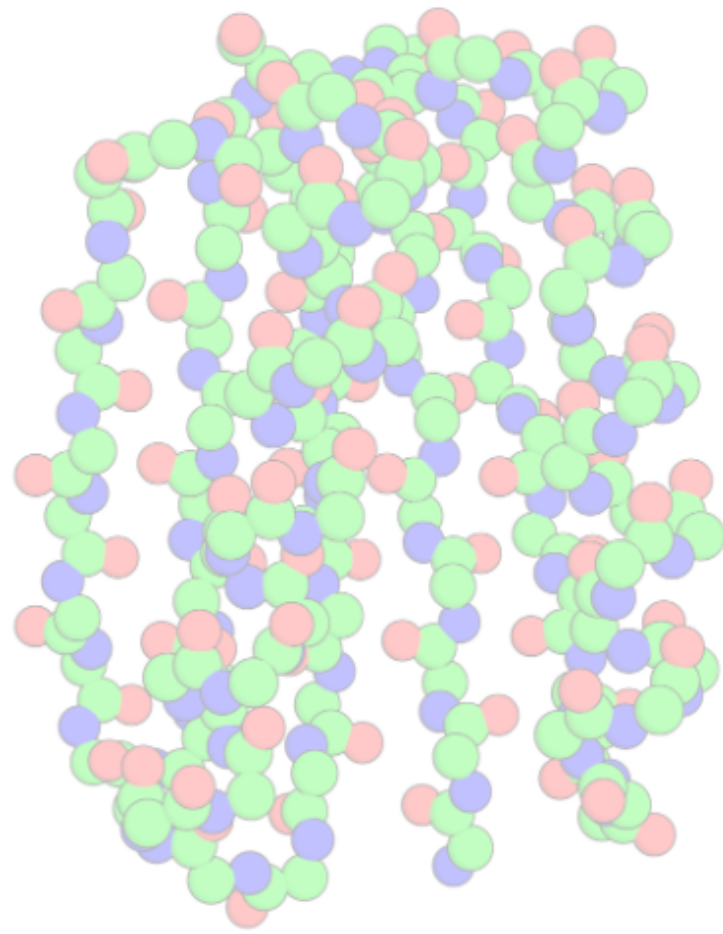


**Option #1:** model 3D  
coordinates of every atom.

- + **Precise control over atom placement.**
- **Bonds are not fixed.**

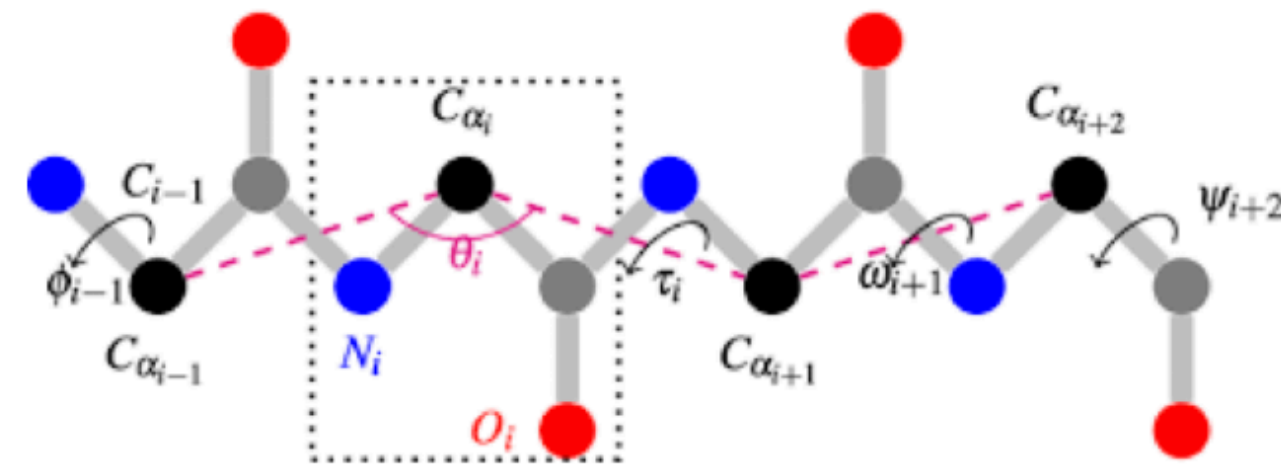


# Structure representations



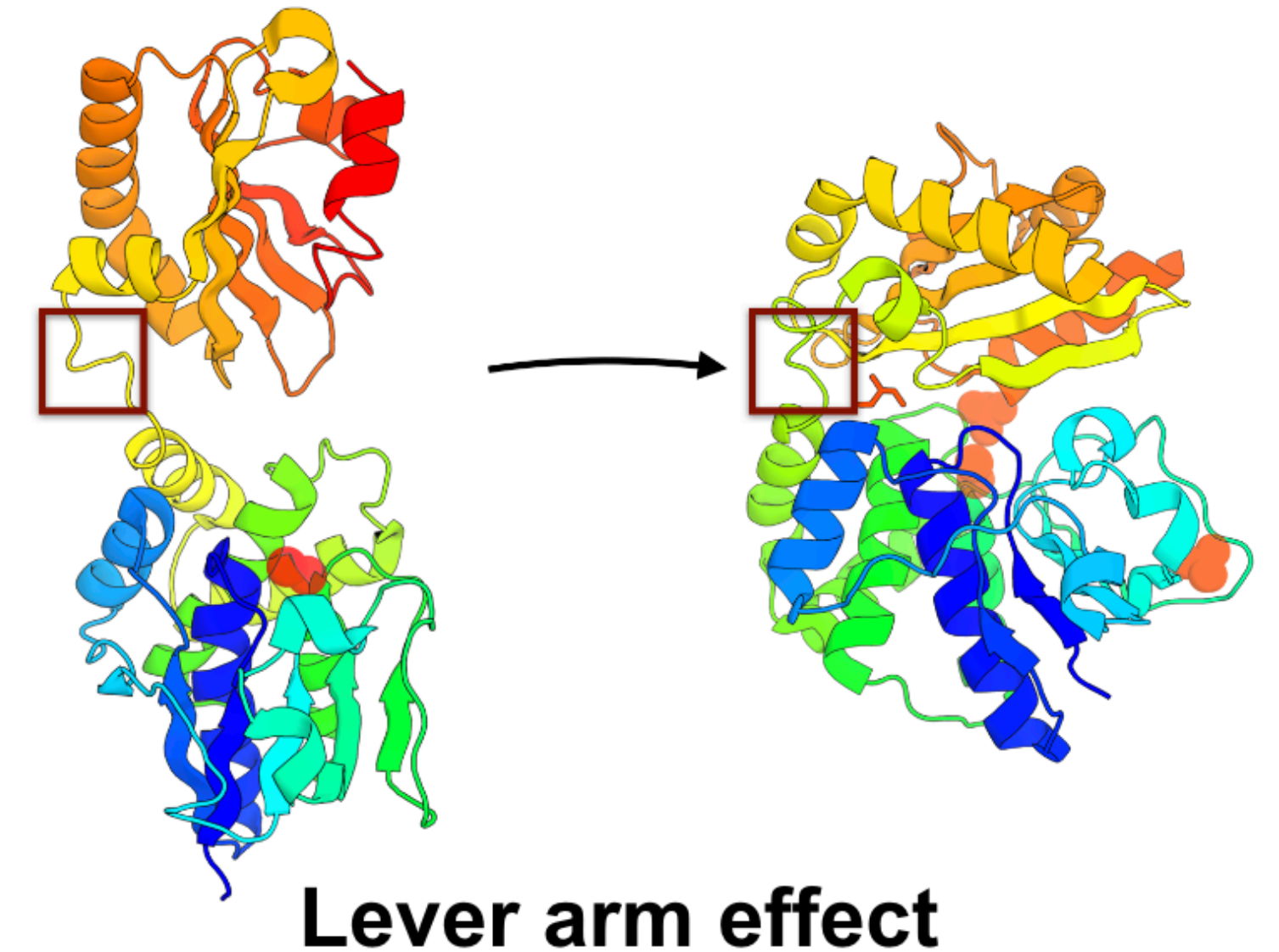
Option #1: model 3D coordinates of every atom.

- + Precise control over atom placement.
- Bonds are not fixed.



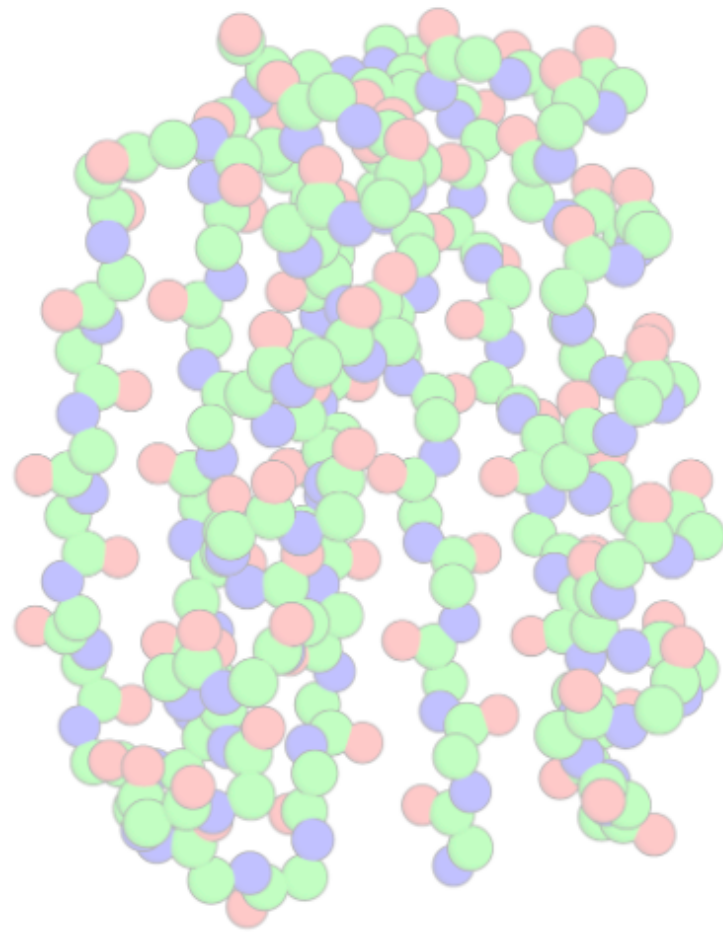
Option #2: model only torsion angles.

- + Bonds are fixed.
- Hard to control atom placement.



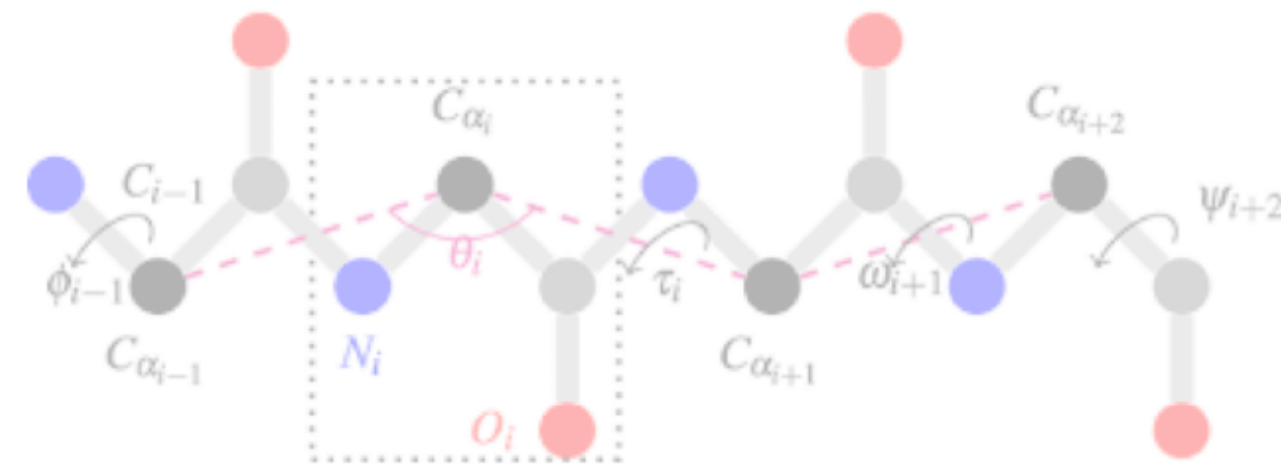
Lever arm effect

# Structure representations



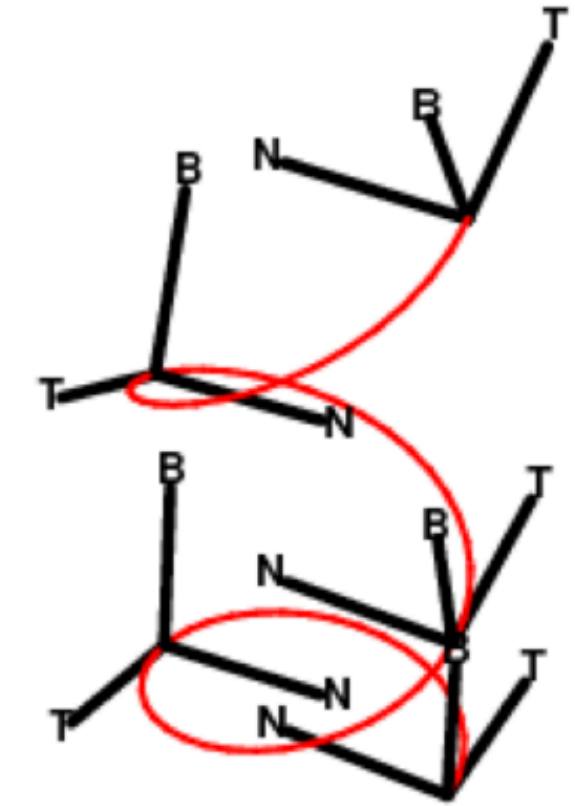
**Option #1:** model 3D coordinates of every atom.

- + Precise control over atom placement.
- Bonds are not fixed.



**Option #2:** model only torsion angles.

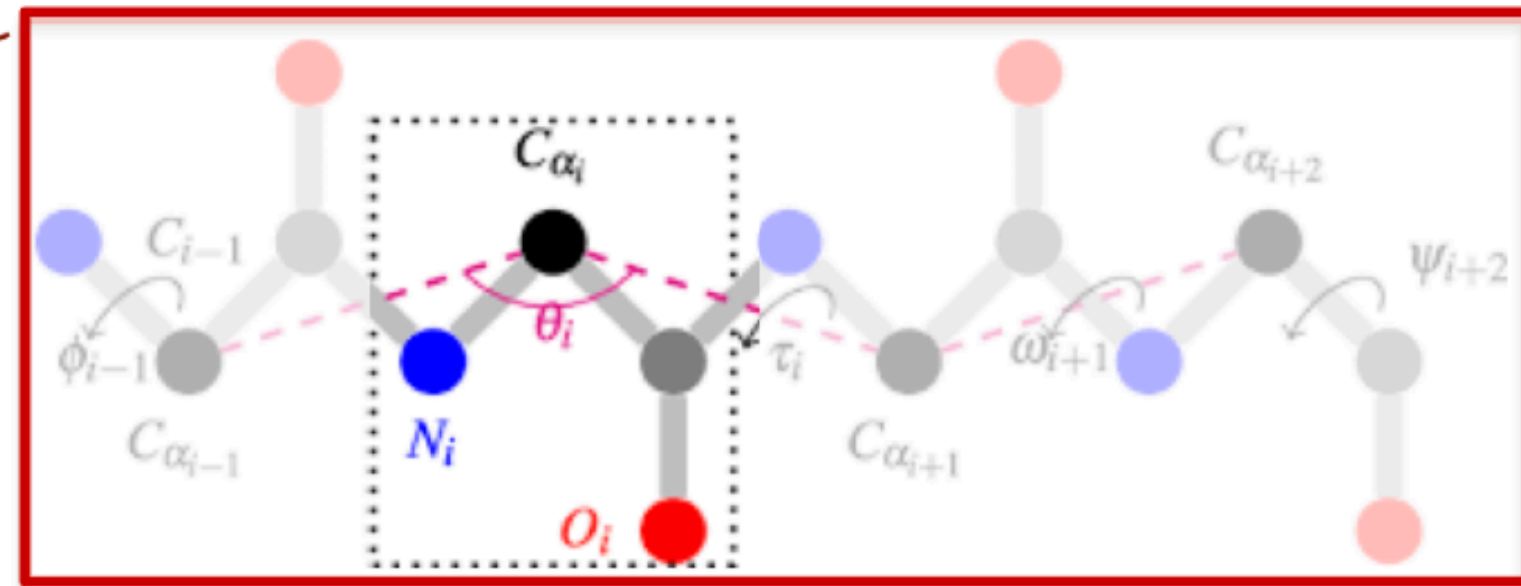
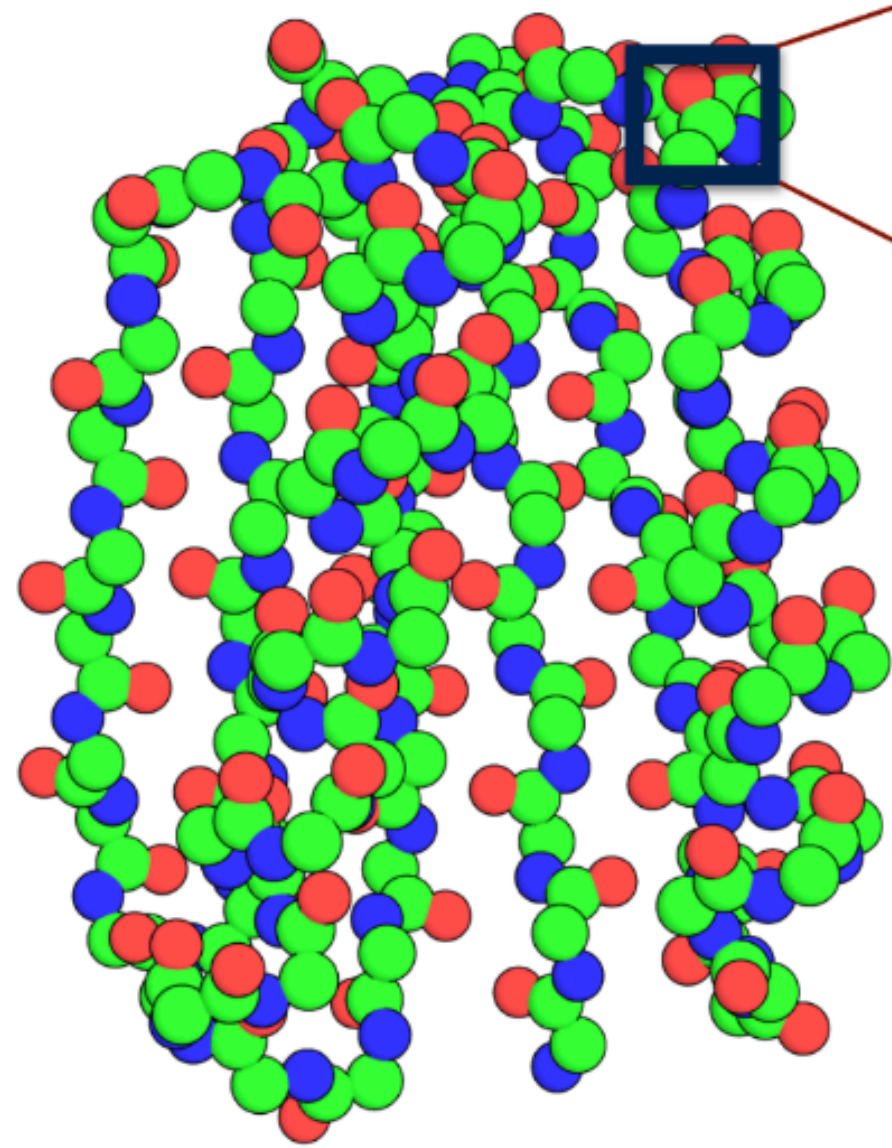
- + Bonds are fixed.
- Hard to control atom placement.



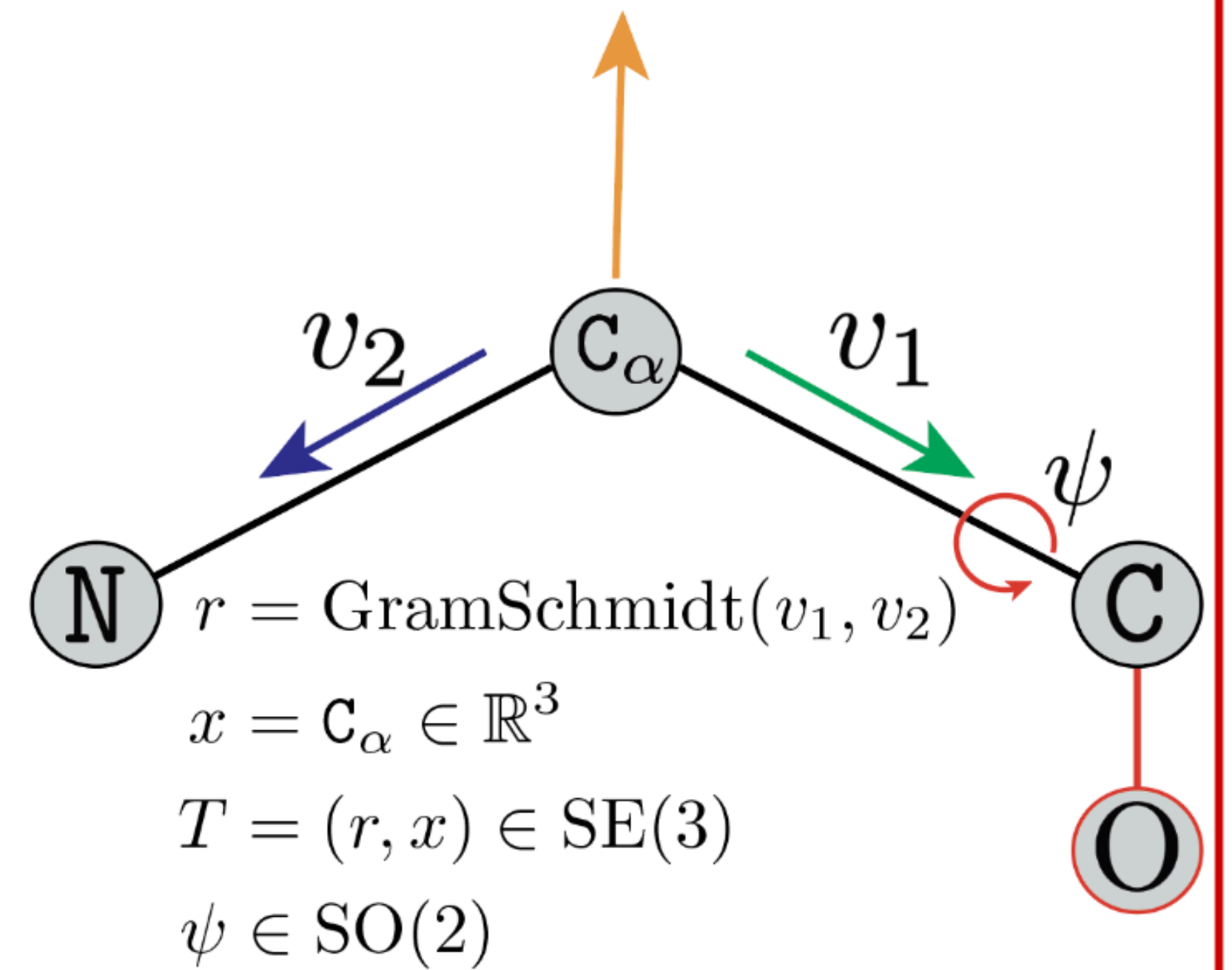
**Option #3:** model with frames along a chain.

- + 3 out of 4 bonds are fixed.
- + Precise control over frame placement
- more complicated math structure.

# Frame representation

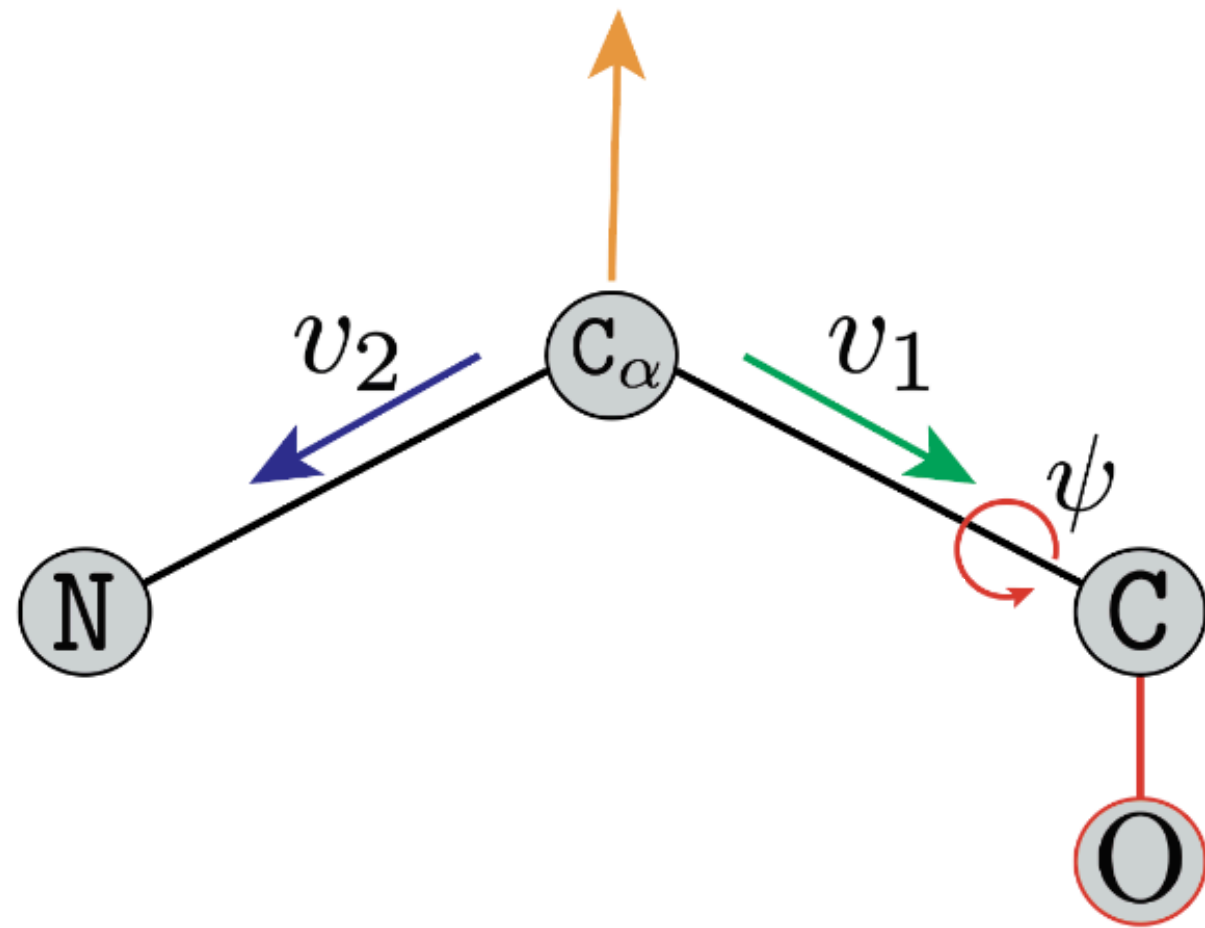


*Repeating atomic structure  
across residues.*

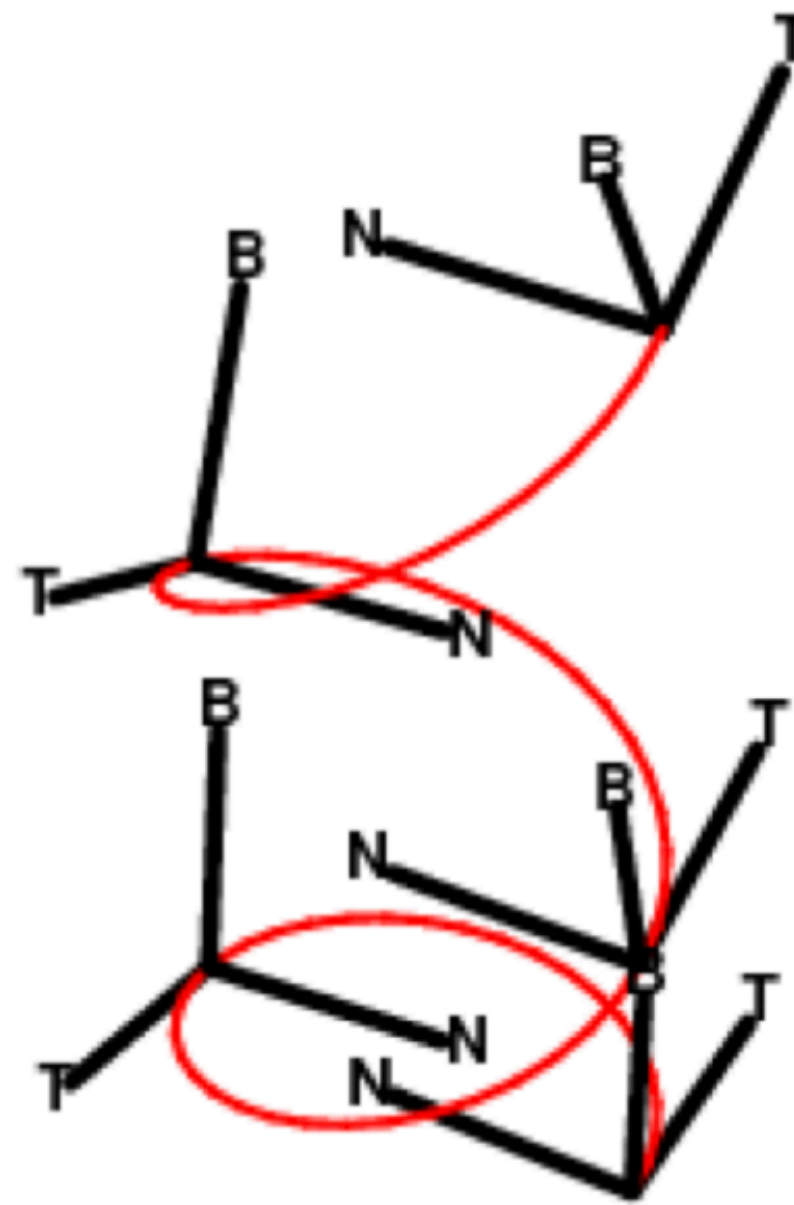


- *Each residue now represented as a Translation ( $C_{\alpha}$  coordinate) and Rotation.*
- *3 out of 4 bonds are automatically satisfied.*

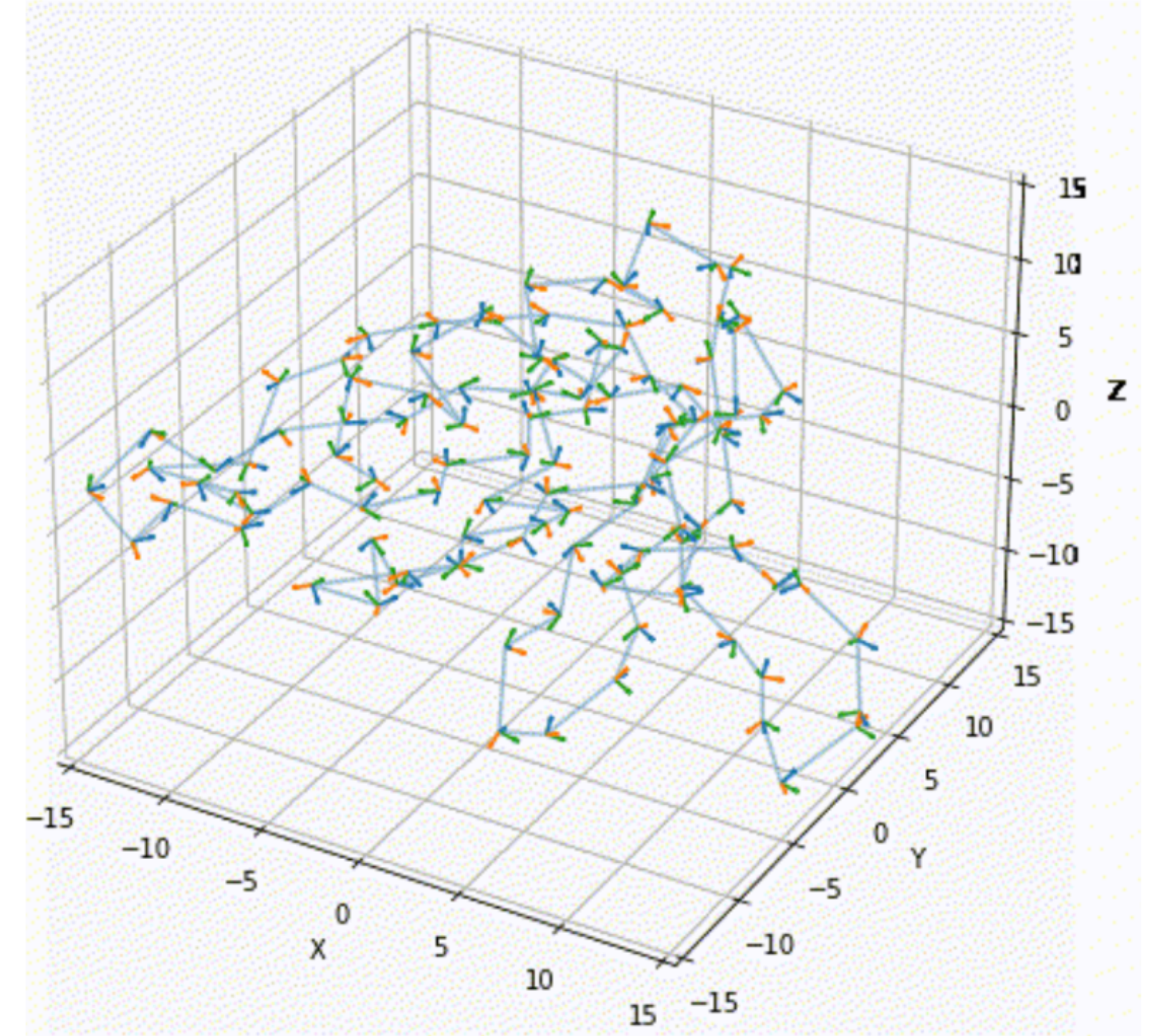
# Frame representation



(1) Single residue **frame**: translation and rotation

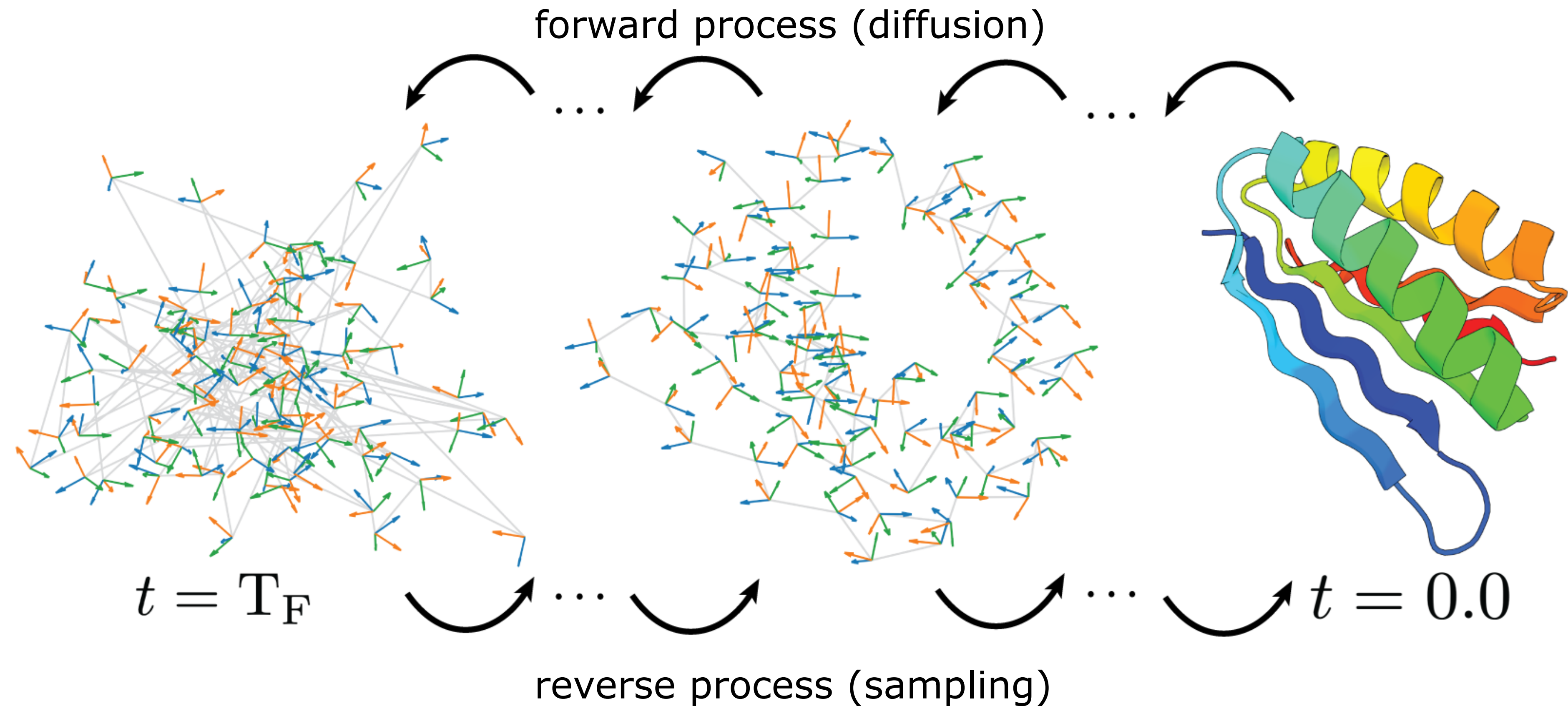


(2) Multiple **frames** along a chain



(3) Every backbone residue is represented with a **frame**.

# Structured diffusion on frames



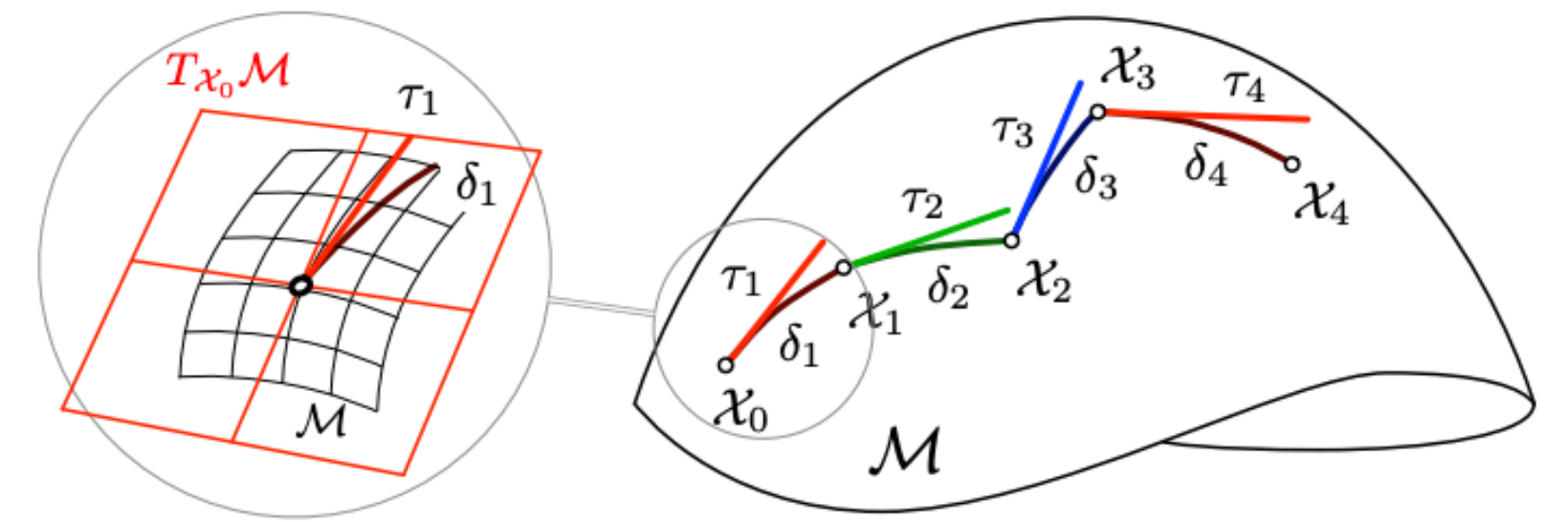
- We want SE(3) invariant densities (same protein, same likelihood); translation invariance based on explicit centering

# Structured diffusion on frames

## SE(3) diffusion construction

- SE(3) diffusion through manifold characterization:

- Brownian motion on SE(3) requires metric on  $T_x \text{SE}(3)$ .



Source: Sola et al

- Choice of metric on SE(3) is crucial: **left-invariant** or **right-invariant**.
- Only the left-invariant metric factorizes into a product manifold.**

$$\text{SE}(3) = \text{SO}(3) \times \mathbb{R}^3 \quad (\text{see Appendix D and Murray et al.})$$

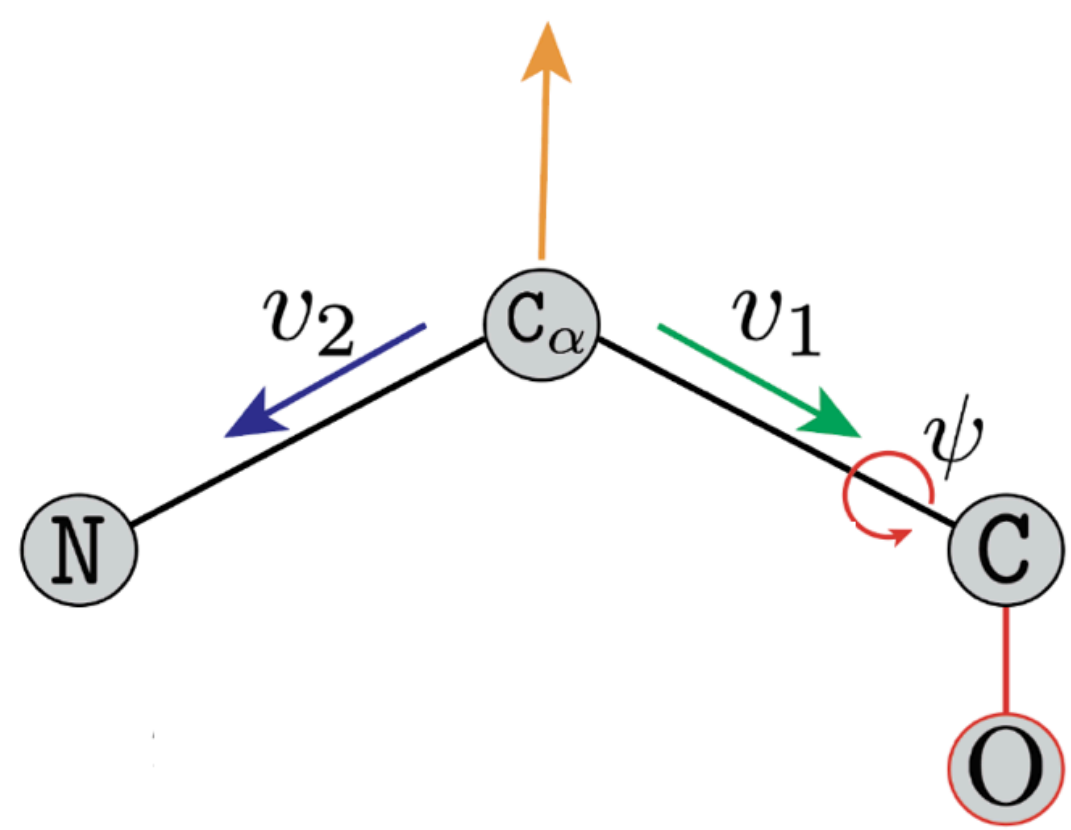
$$\text{where } \langle (R, x) \cdot (R', x') \rangle_{\text{SE}(3)} = \langle R, R' \rangle_{\text{SO}(3)} + \langle x, x' \rangle_{\mathbb{R}^3}$$

Independence! Also crucial for invariance of measure on  $\text{SE}(3)^N$  (see Prop 3.5)

# Structured diffusion on frames

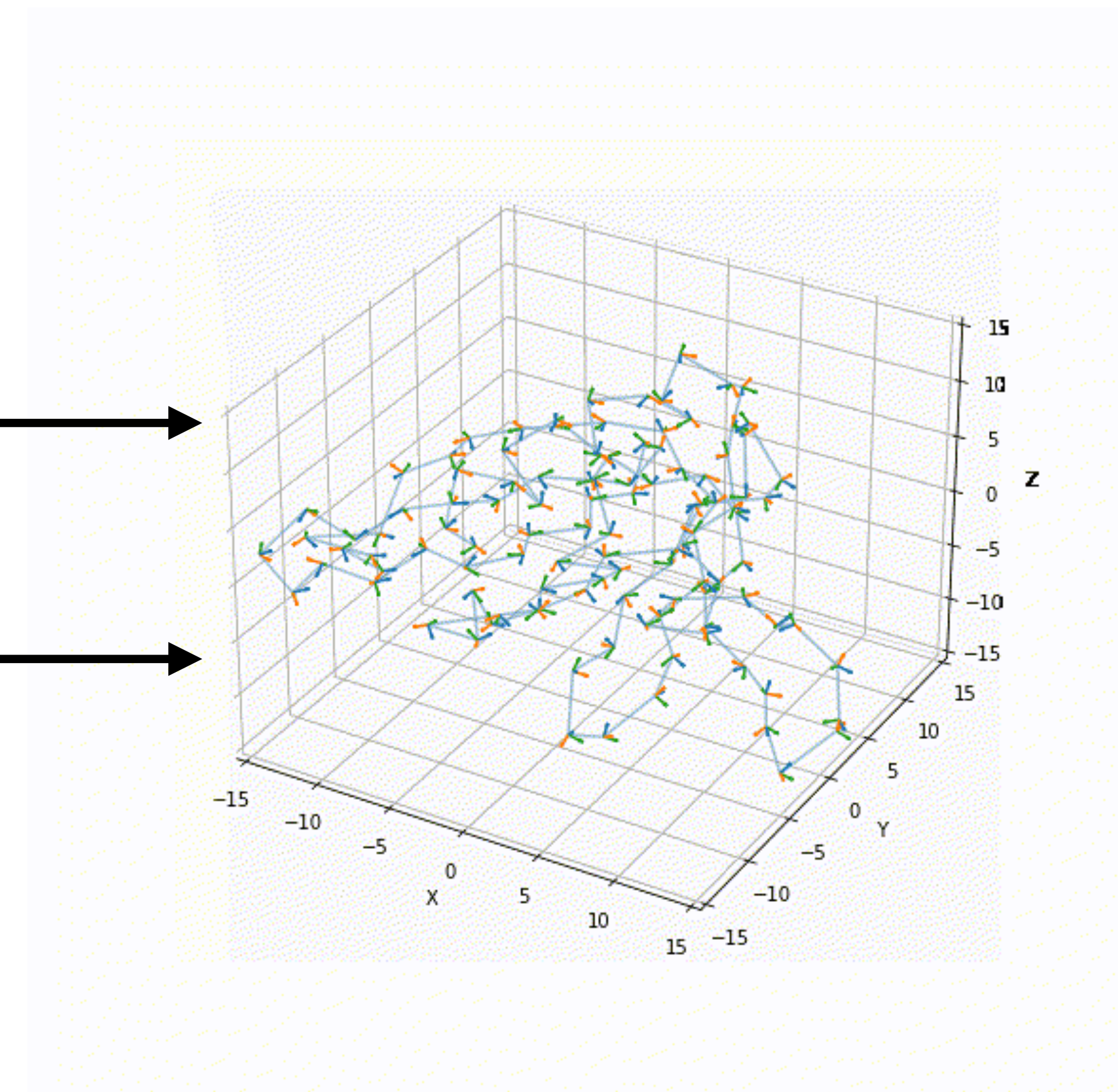
## How to diffuse a frame?

- As a **Riemannian manifold**,  $SE(3) = SO(3) \times \mathbb{R}^3$ .
- Translations and rotations can be **diffused independently**:



Diffuse translations  $x \in \mathbb{R}^3$

Diffuse rotation  $R \in SO(3)$



Frame  $(R, x) \in SO(3) \times \mathbb{R}^3$

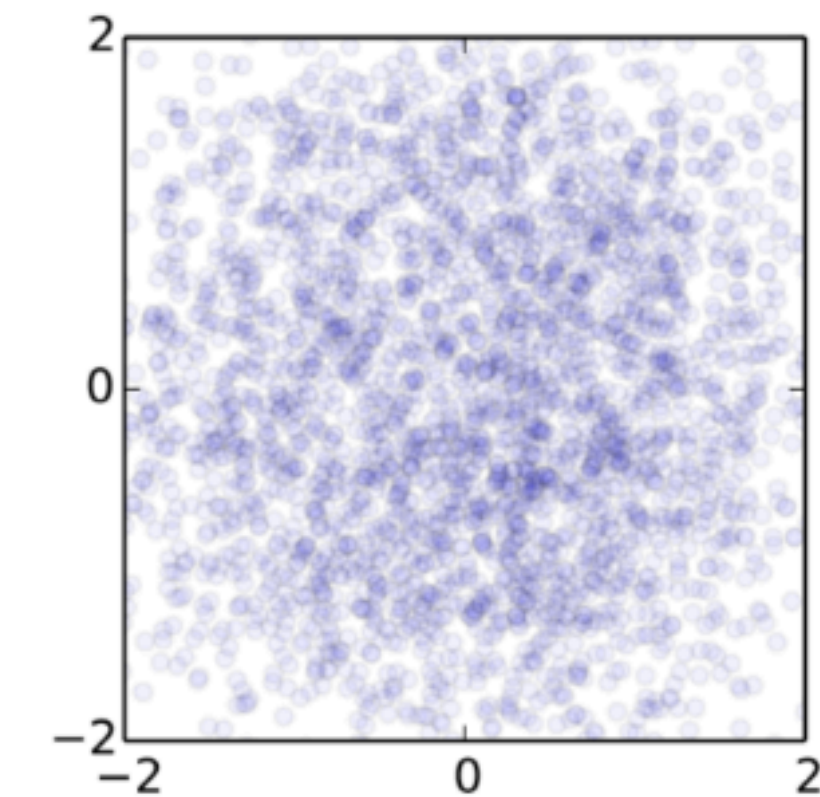
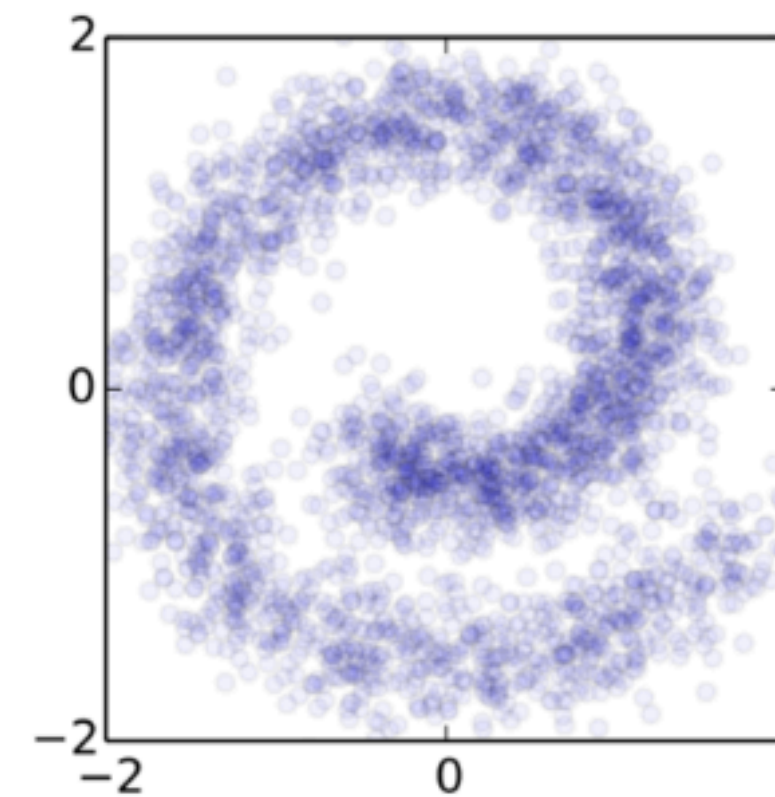
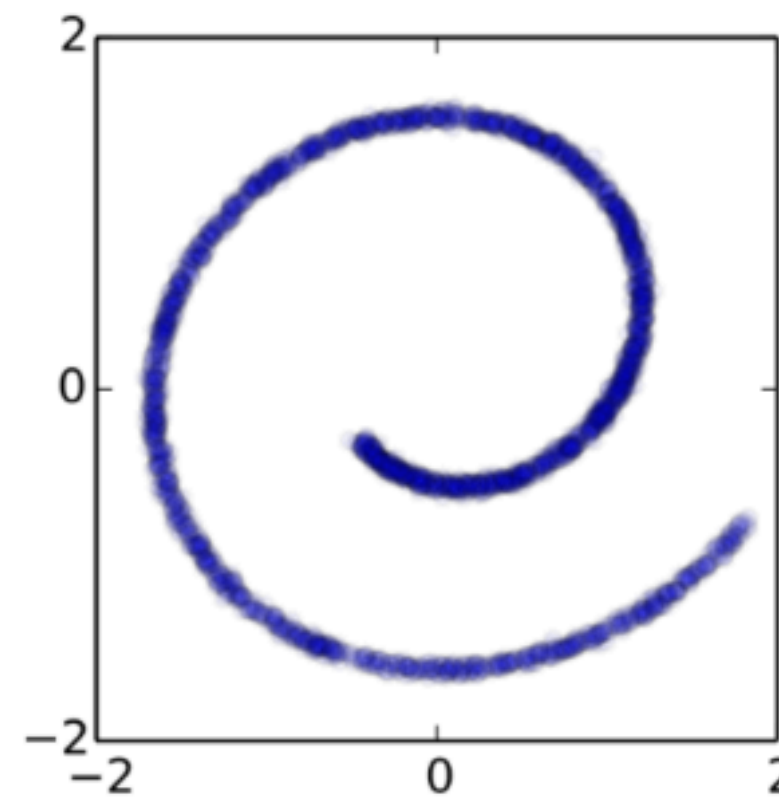
# Structured diffusion on frames

## How to diffuse a frame?

Diffuse translations  $x \in \mathbb{R}^3$

Brownian motion on  $\mathbb{R}^3$

$$p_{t|0}(x^{(t)} | x^{(0)}) = \mathcal{N}(x^{(t)}; \beta(t)x^{(0)}, \sigma(t))$$



(2D for visualization)

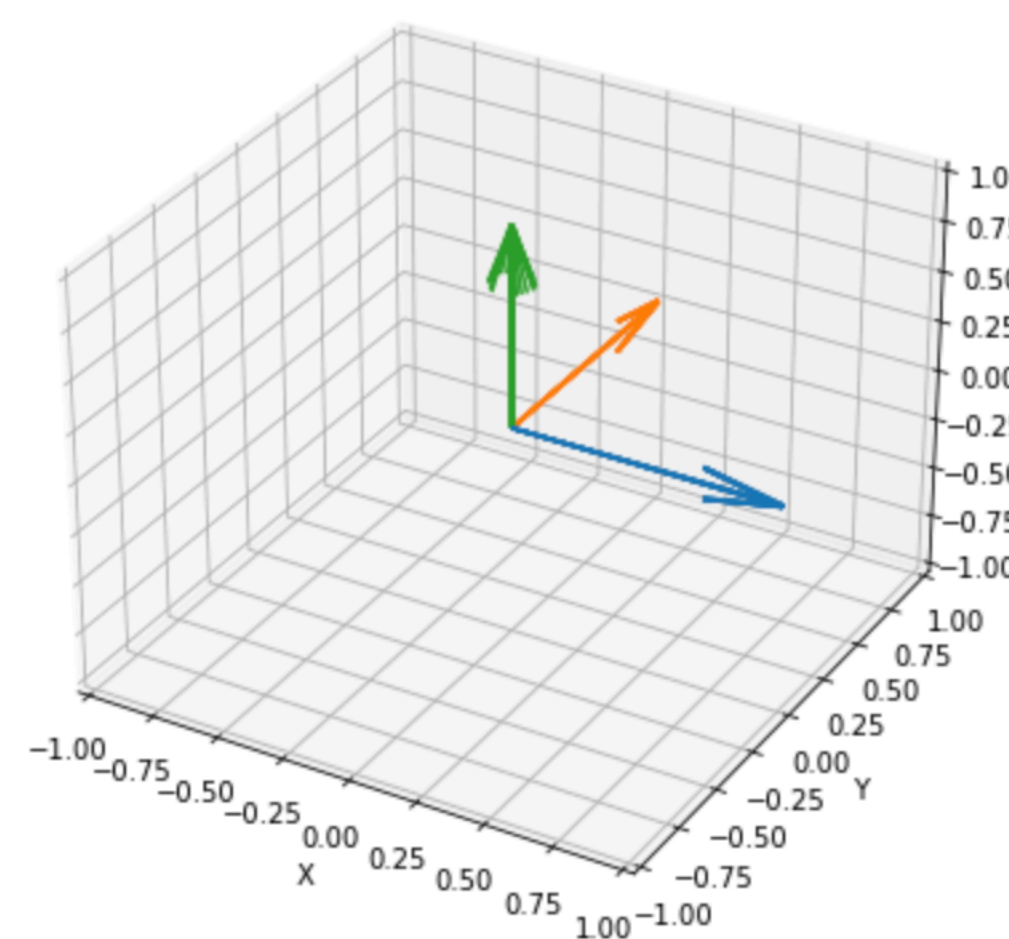
Source: Lilian Weng

Diffuse rotation  $R \in \text{SO}(3)$

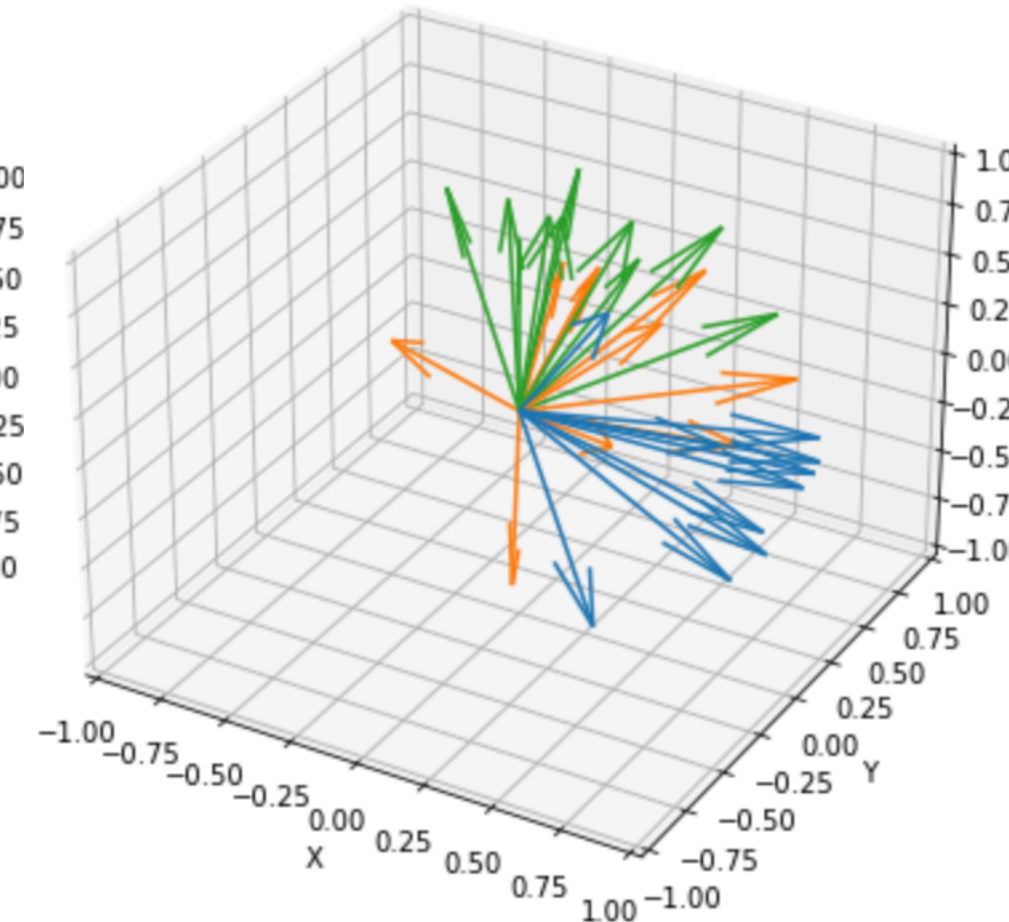
Brownian motion on  $\text{SO}(3)$

$$p_{t|0}(R^{(t)} | R^{(0)}) = \text{IGSO}_3(r^{(t)}; r^{(0)}, t)$$

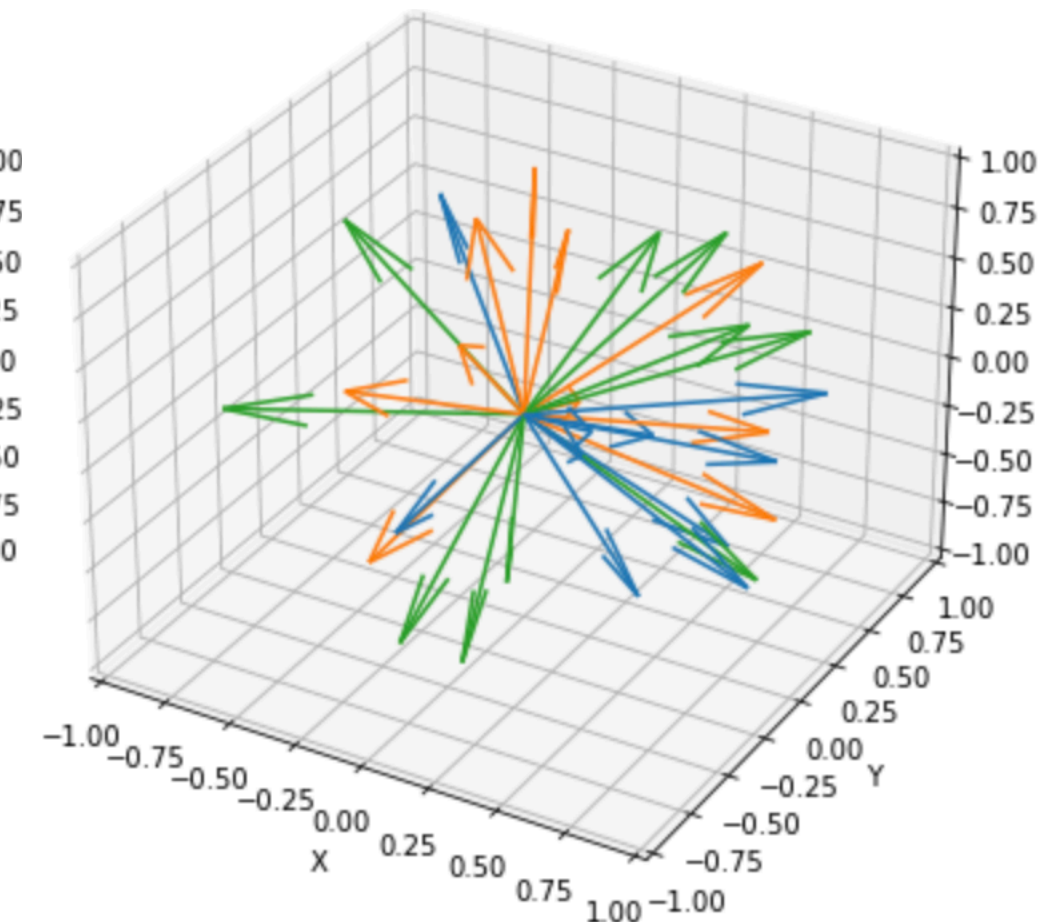
where  $r^{(t)} = \text{Log}(R^{(t)})$ ,  $r^{(0)} = \text{Log}(R^{(0)})$



$t = 0.0$



$t = 0.5$

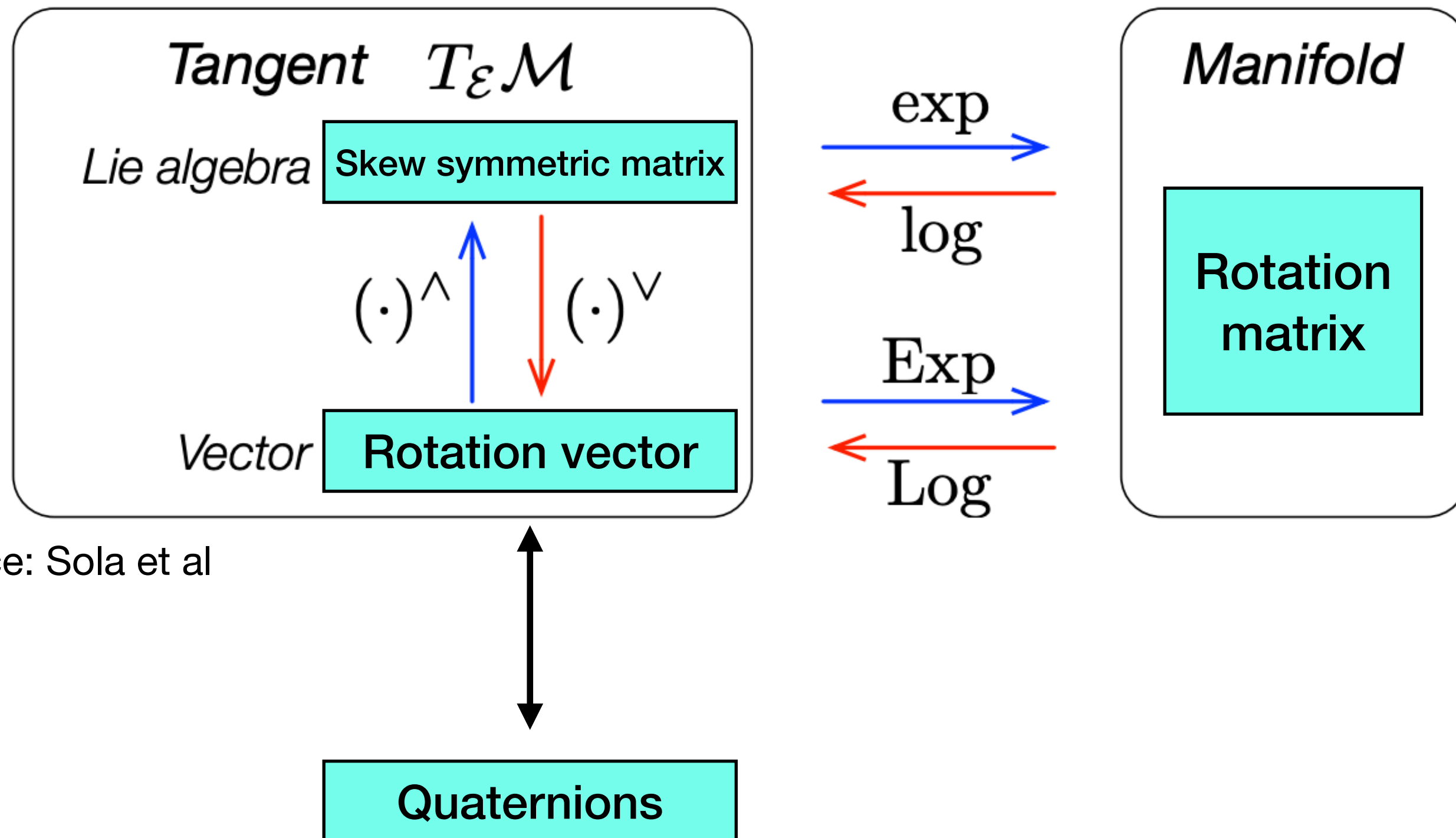


$t = T$



# Background on SO(3)

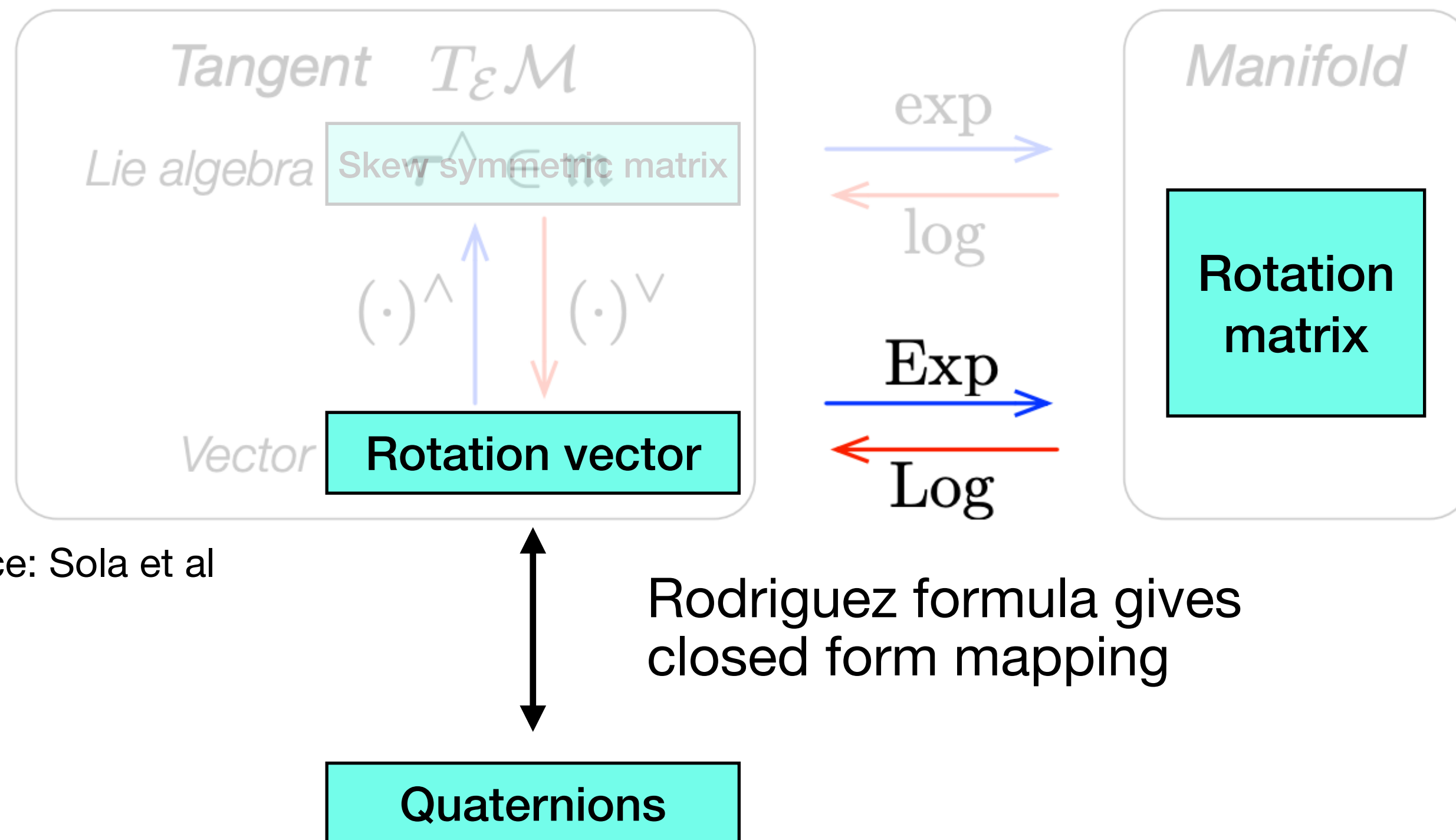
## Mappings in SO(3)



Source: Sola et al

# Background on SO(3)

## Mappings in SO(3)



Source: Sola et al

## Use cases

- **Rotation matrix:** orthogonal  $3 \times 3$  matrices
  - Form for multiplication.
  - *Cons:* can't add.
- **Rotation vector:** vectors in  $\mathbb{R}^3$ 
  - Allows addition and brownian motion.
  - *Cons:* degenerate.
- **Quaternion:**  $\mathbb{R}^4$ 
  - Numerically stable. Representation of neural network output.
  - *Cons:* can't add or multiply.

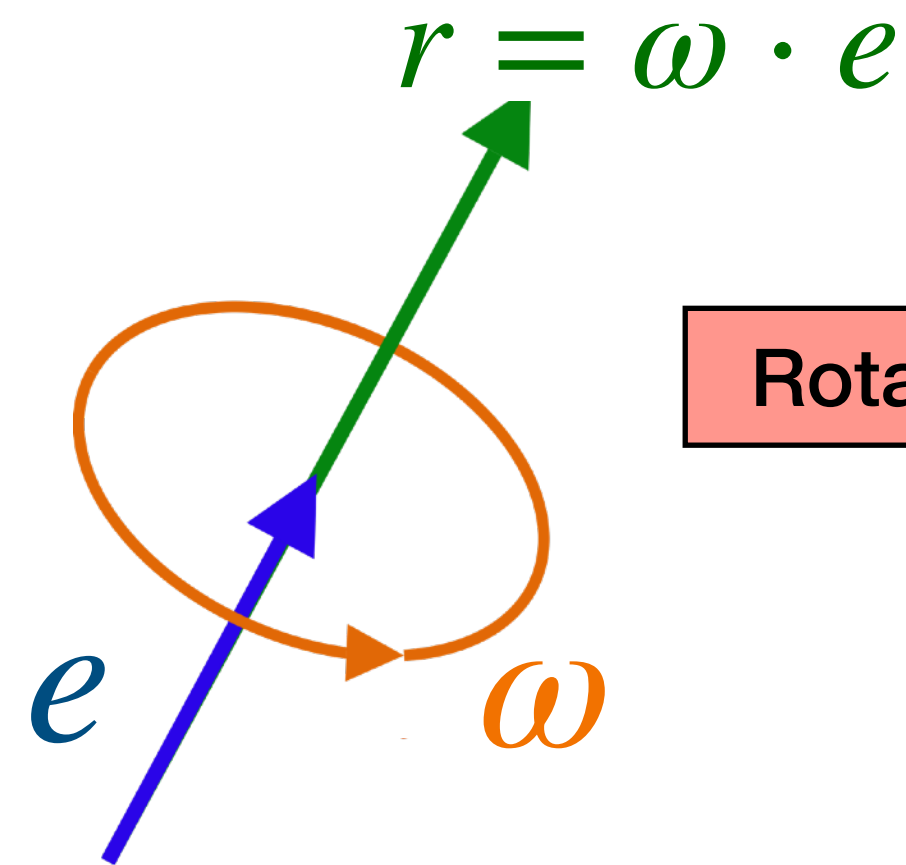
# Brownian motion on SO(3)

Diffuse rotation  $R \in \text{SO}(3)$

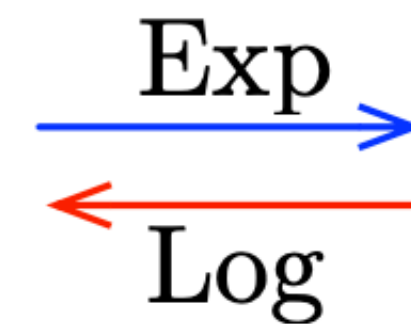
Brownian motion on SO(3)

$$p_{t|0}(R^{(t)} | R^{(0)}) = \text{IGSO}_3(r^{(t)}; r^{(0)}, t)$$

where  $r^{(t)} = \text{Log}(R^{(t)})$ ,  $r^{(0)} = \text{Log}(R^{(0)})$



Rotation vector

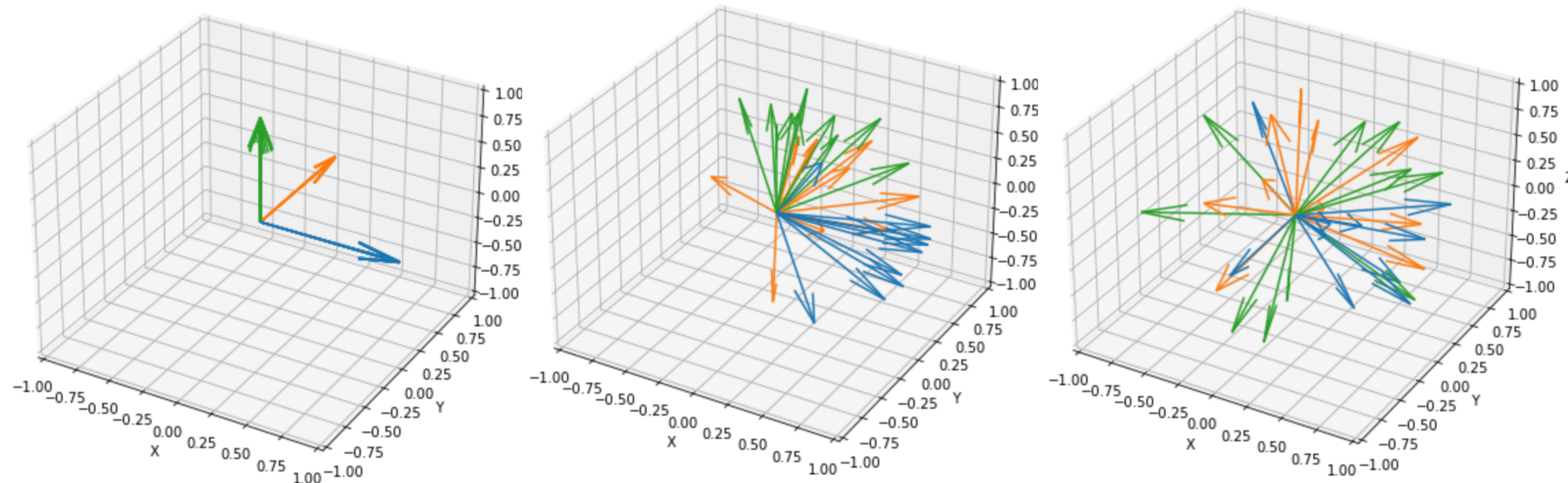


Rotation matrix

**IGSO(3) density:** heat kernel (brownian motion) on SO(3)

$$f(\omega, t) = \sum_{\ell \in \mathbb{N}} (2\ell + 1) e^{-\ell(\ell+1)t/2} \frac{\sin((\ell+1/2)\omega)}{\sin(\omega/2)}$$

Axis  $e$  sampled uniformly over unit sphere.



$t = 0.0$

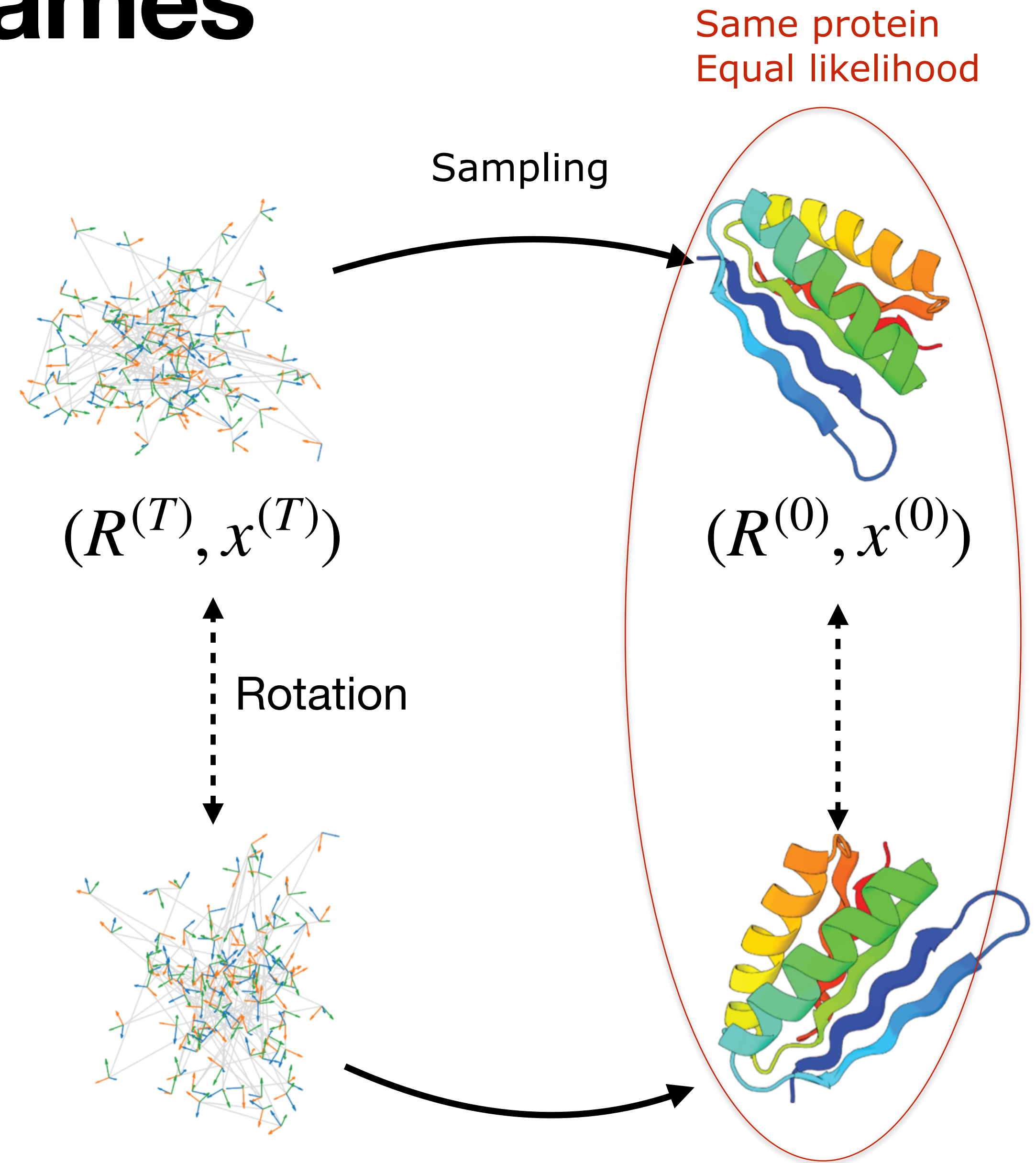
$t = 0.5$

$t = T$

# Structured diffusion on frames

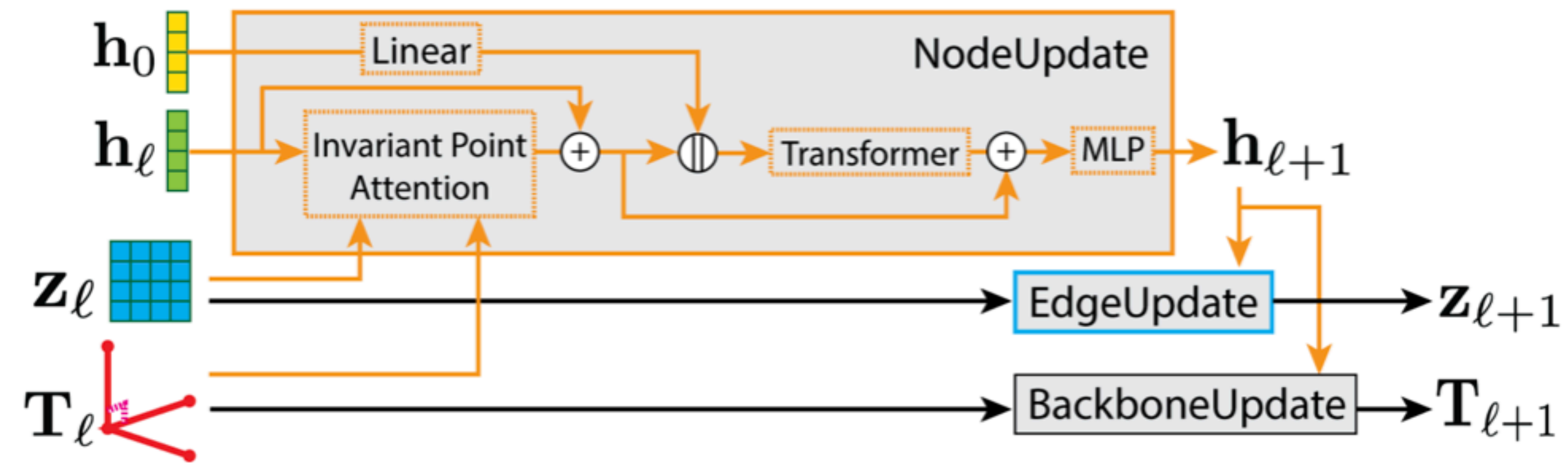
## $SE(3)^N$ invariance

- Forward process
  - Rotational invariance: by construction of distributions.
  - Translation invariance: by zero-centering.
- Reverse process
  - By learning a  $SE(3)^N$  equivariant score model.
  - We use Invariant Point Attention (IPA).



# FrameDiff

- SO(3) equivariant architecture leveraging IPA



- Model predicts  $\hat{T}^{(0)} = \text{FrameDiff}(T^{(t)})$  and its parameters are learned by score matching predicted and actual scores

$$s_\theta(T^{(t)}, t) := \nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | \hat{T}^{(0)})$$

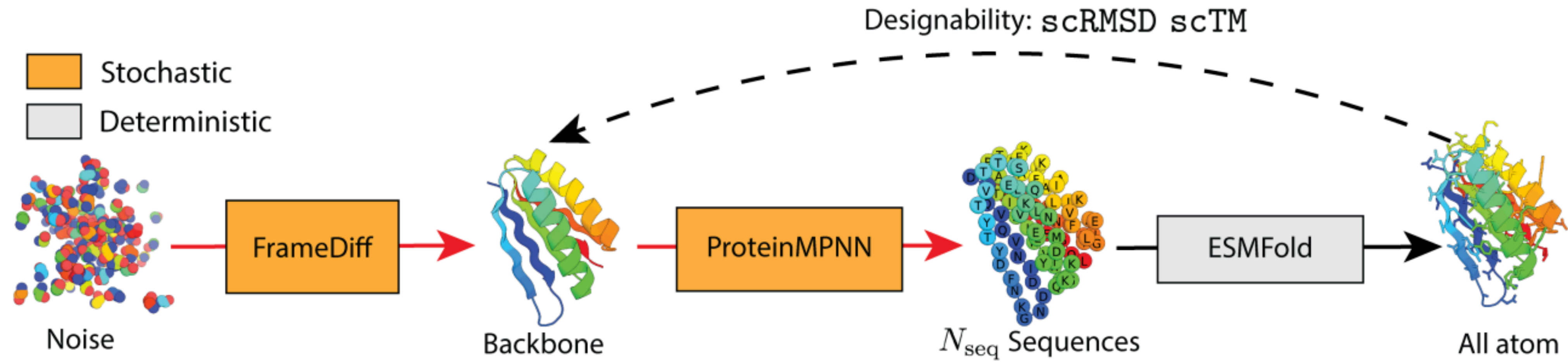
predicted frames

$$\text{minimize } \mathbb{E}_{T^{(0)}, t, T^{(t)}} [\|s_\theta(T^{(t)}, t) - \nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | T^{(0)})\|^2]$$

actual frames

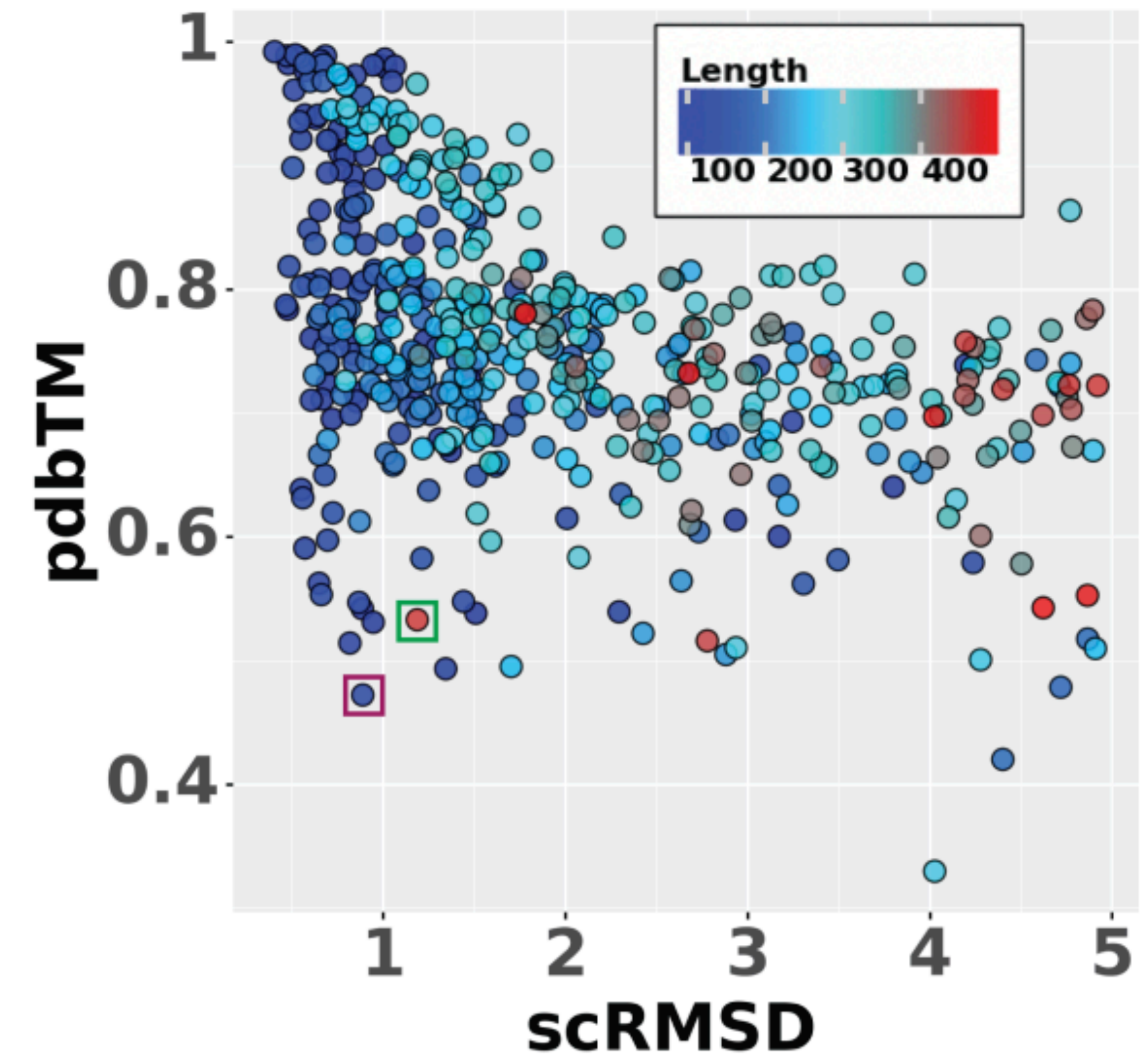
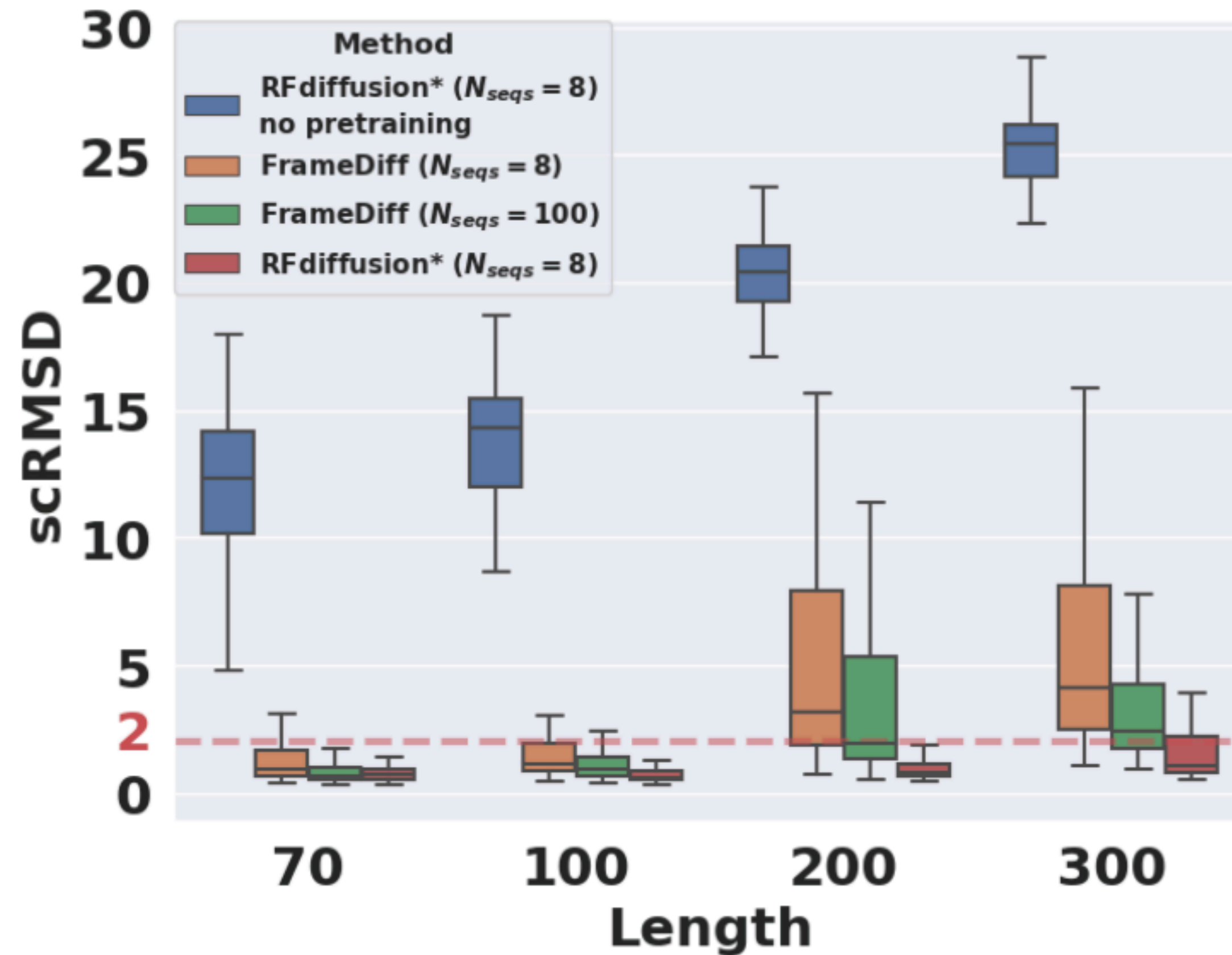
# FrameDiff evaluation

- **Data:** filtered backbone structures from PDB
  - Monomer (single chain)
  - Length between 60 to 512 residues
  - High secondary structure content (<50% loops)
  - Total: 19777 structures
- **Benchmarking:**



# FrameDiff results

- Pseudo-benchmarking since each diffusion model trains on different dataset.



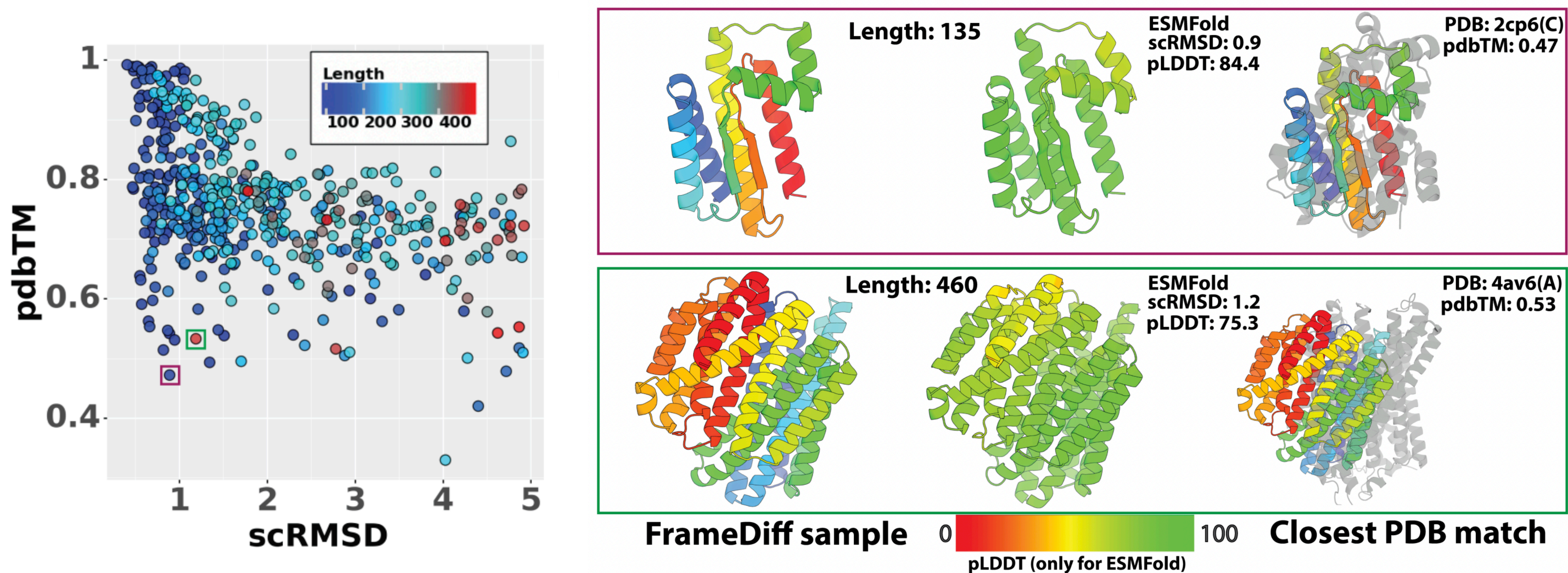
# FrameDiff sampling





# FrameDiff results

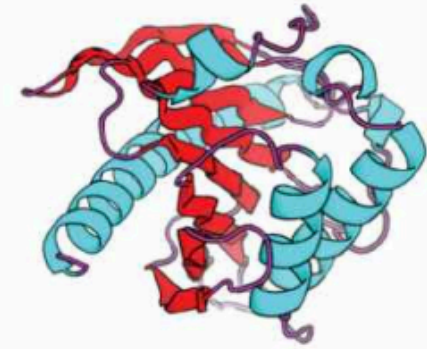
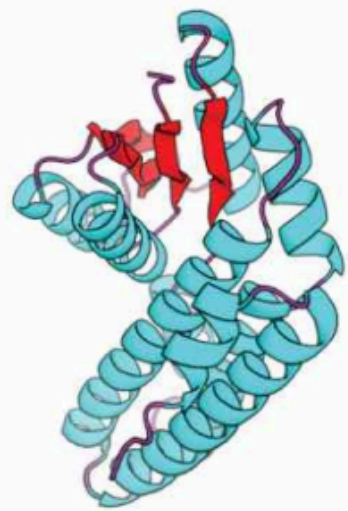
- In-silico* evidence of generalizing beyond PDB



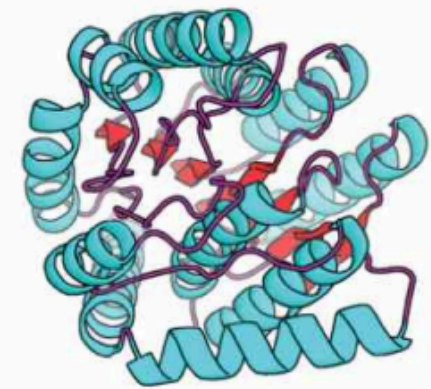
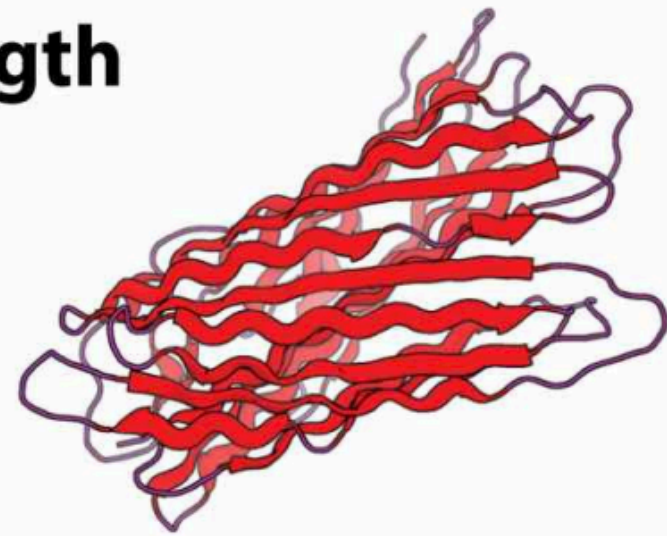
# Samples at noise scales

## Noise scale 0.1

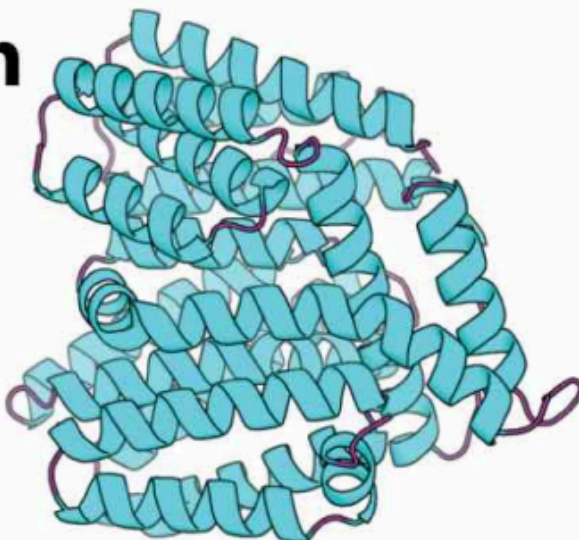
Length  
200



Length  
300

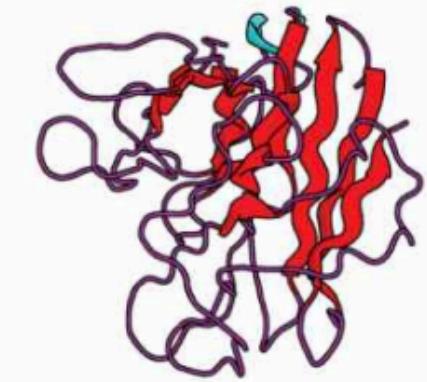


Length  
400

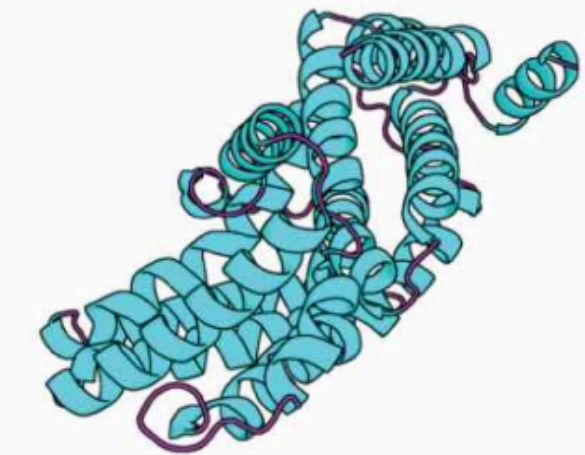
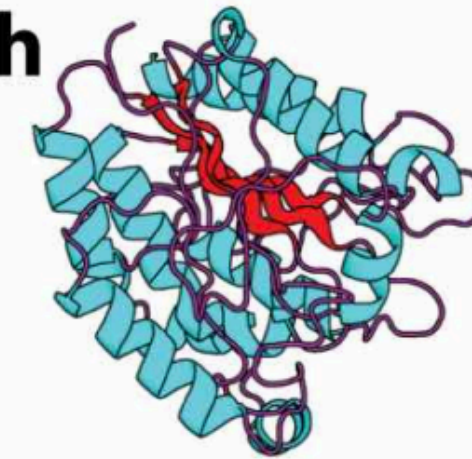


## Noise scale 1.0

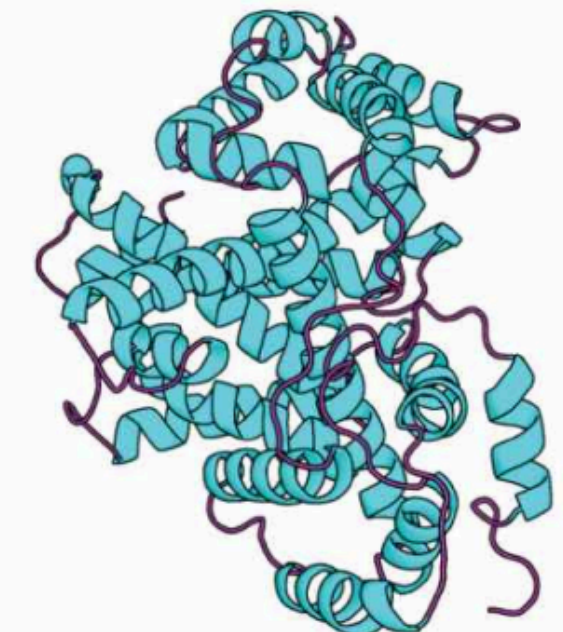
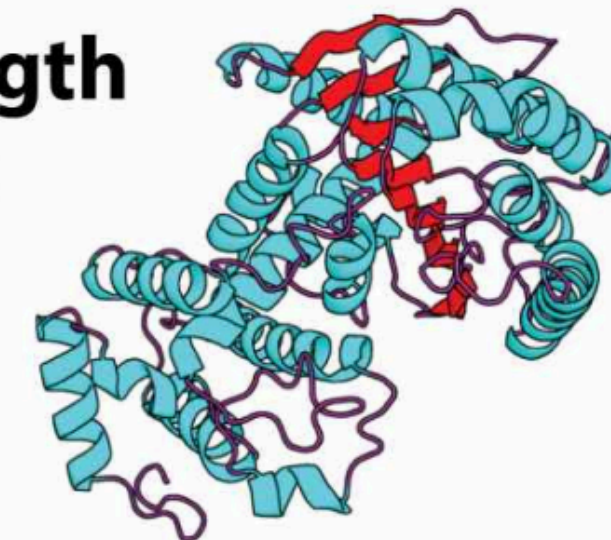
Length  
200



Length  
300



Length  
400



# Conclusion

- FrameDiff rigorously develops SE(3) diffusion.
- FrameDiff follows best practices in protein modeling from AF2 (frames, IPA).

## Limitations:

- Jointly generating sequence and side chains.
- Conditional generation, i.e. motif-scaffolding.

# RFdiffusion

**Paper: Broadly applicable protein design with RFdiffusion**

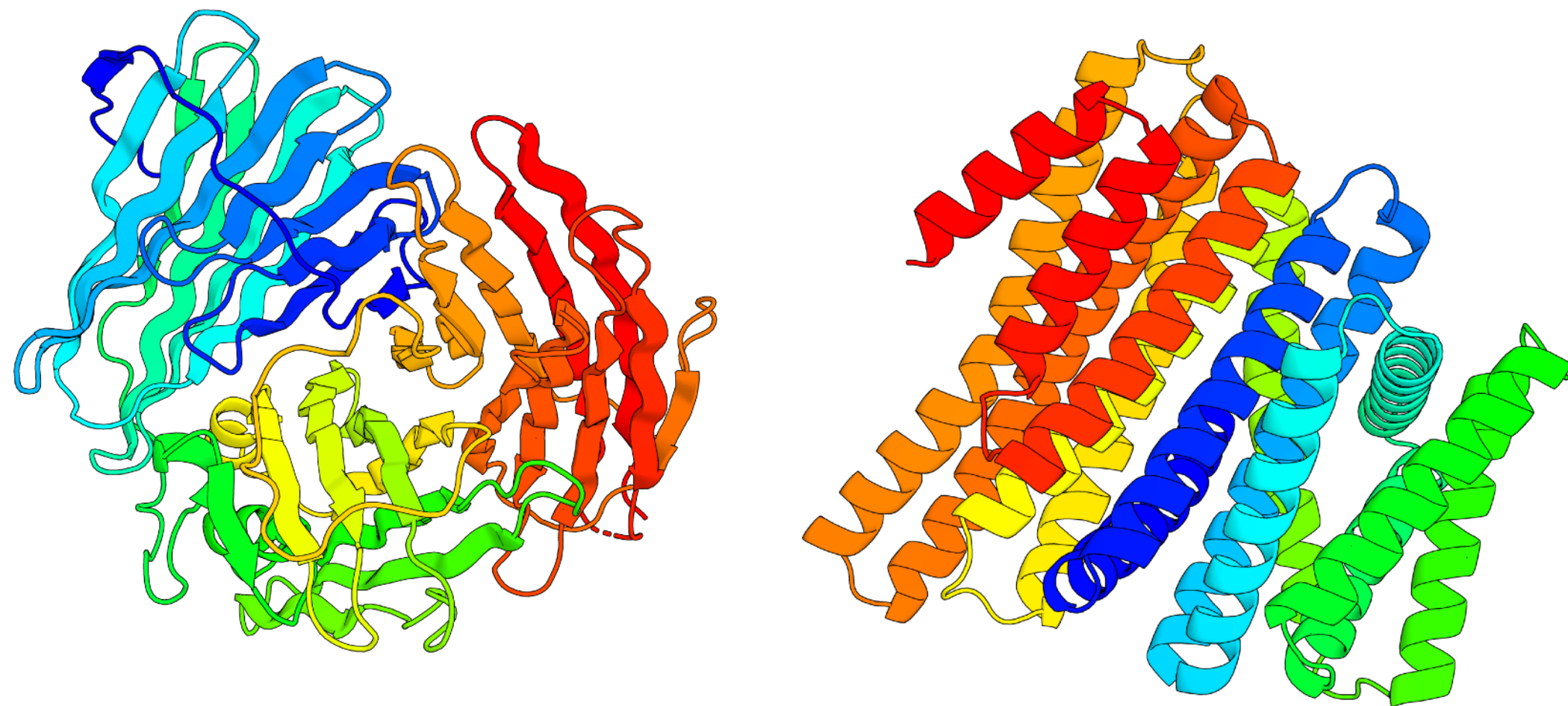
**Joseph Watson\*, David Juergens\*, Nathaniel Bennett\*, Brian Trippe\*, Jason Yim\*  
Helen Eisenach\*, Woody Ahern\*, and others**

**Preprint**

(see previous recorded talks of RFdiffusion)

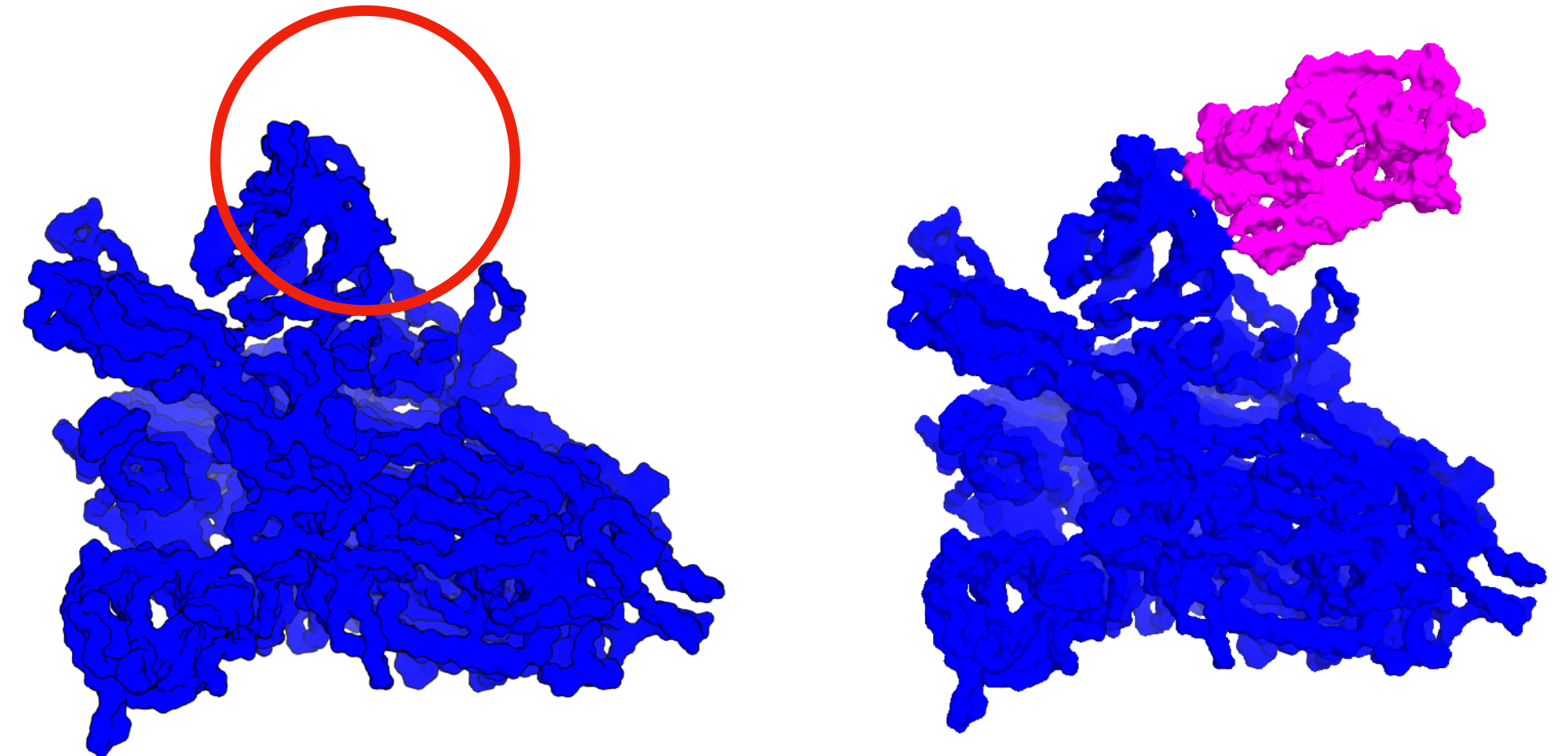
# RosettaFold diffusion

**FrameDiff:** *unconditional*  
SE(3) diffusion model



$P(x)$

**RFdiffusion:** *unconditional* and  
*conditional* SE(3) diffusion model

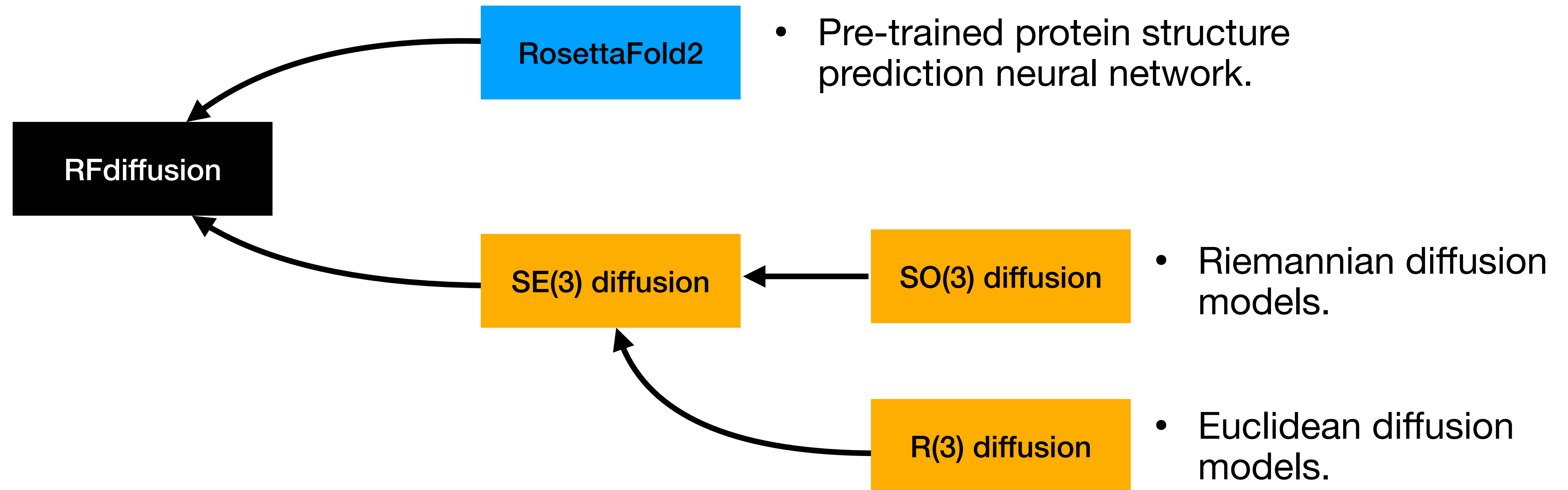


Condition  $y$

$P(x | y)$

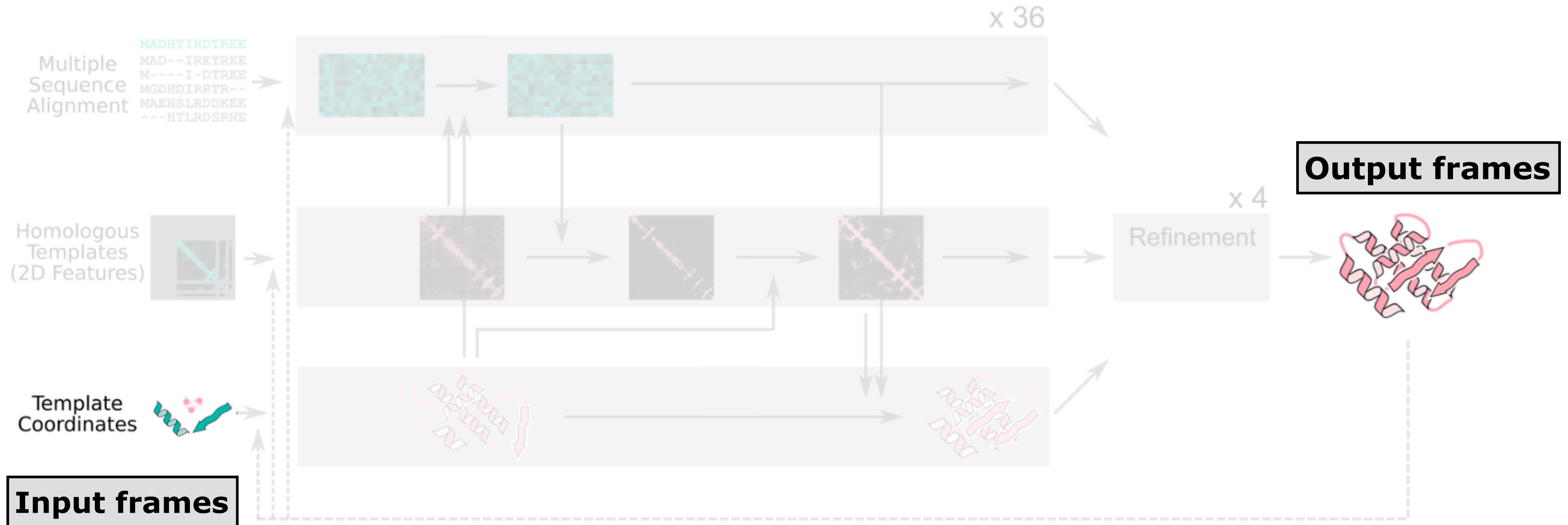
# RosettaFold diffusion

- RosettaFold diffusion is a culmination of multiple ideas.



# RosettaFold diffusion

- **Improvement #1:** pre-training with RoseTTAFold
- *Compatibility:* RosettaFold uses SE(3) frames as *input* and *output*.



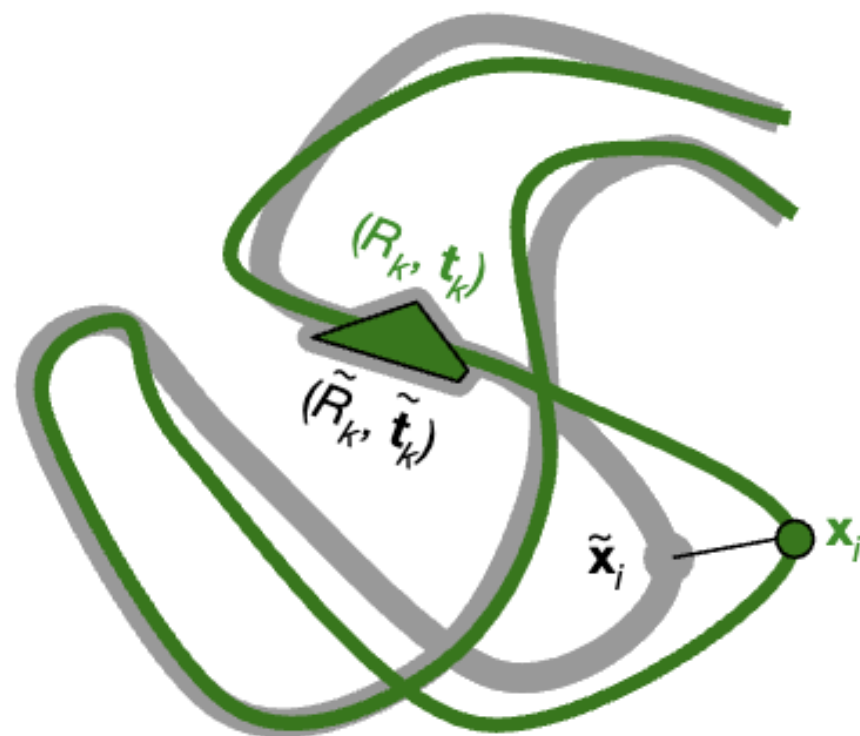
# RosettaFold diffusion

- **Improvement #1:** pre-training with RoseTTAFold
  - *Compatibility:* RosettaFold uses SE(3) frames as *input* and *output*.
  - *Learning:* structure prediction and SE(3) diffusion share *similar* objectives.

## Structure prediction loss:

Frame aligned point error (FAPE)

$$\sum_{i,j} \|\hat{T}_i^{-1} \hat{x}_j - T_i^{-1} x_j\|^2$$



## SE(3) Diffusion loss:

Denosing score matching (DSM)

$$\|\nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | \hat{T}^{(0)}) - \nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | T^{(0)})\|^2$$

Predicted frames                      True frames

Functions share the same minimizer but different gradients during training.

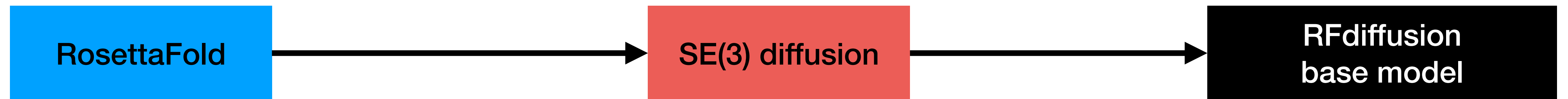


# RosettaFold diffusion

- **Training pipeline. Sampling follows FrameDiff procedure.**

Training with FAPE loss

SE(3) diffusion training\*



$$\|\nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | \hat{T}^{(0)}) - \nabla_{T^{(t)}} \log p_{t|0}(T^{(t)} | T^{(0)})\|_2^2 \Leftrightarrow \|\hat{x}^{(0)} - x^{(0)}\|_2^2 + \|\hat{R}^{(0)} - R^{(0)}\|_F^2$$

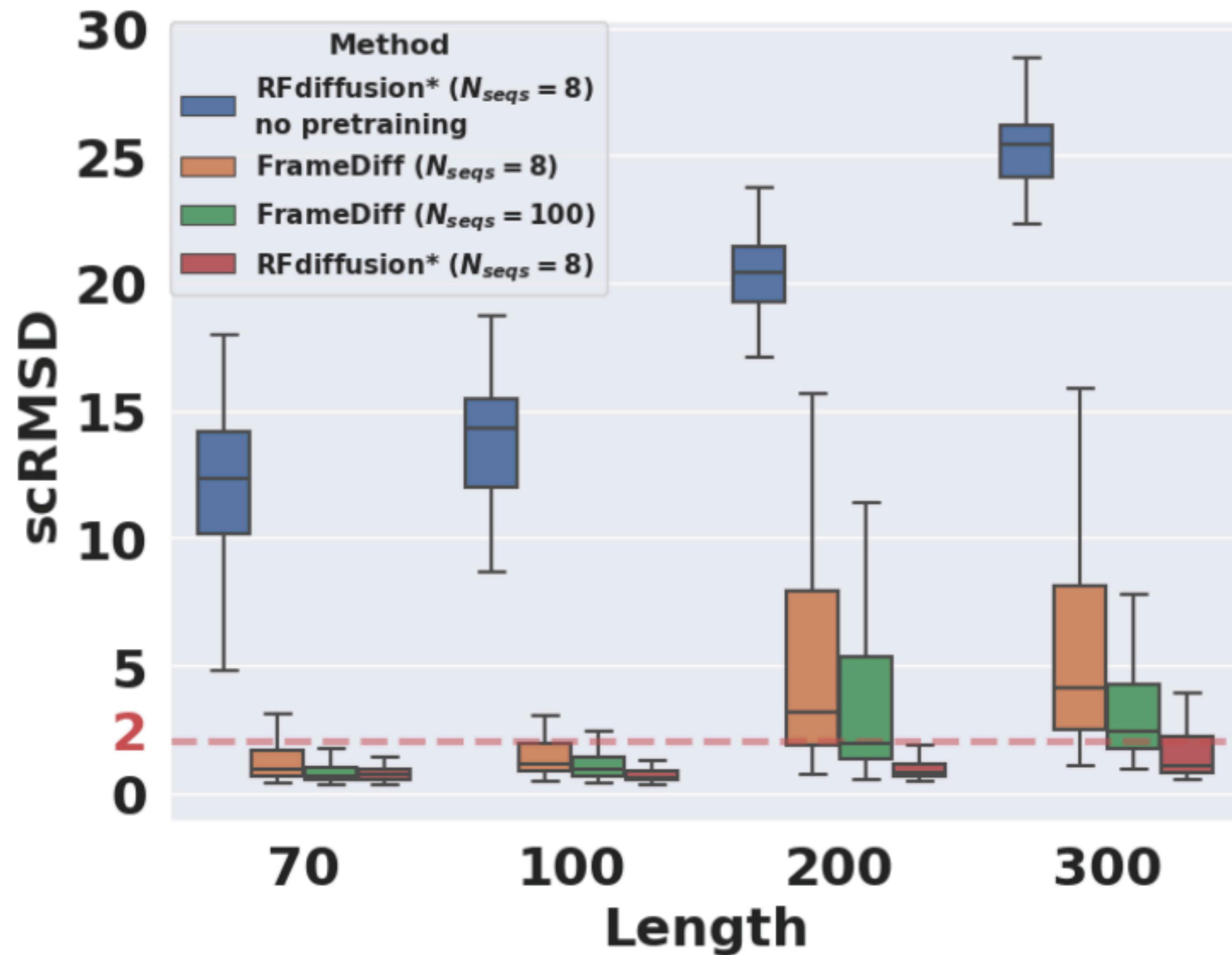
Denoising score matching (DSM).

DSM approximation used in training.  
Better agreement with FAPE loss.

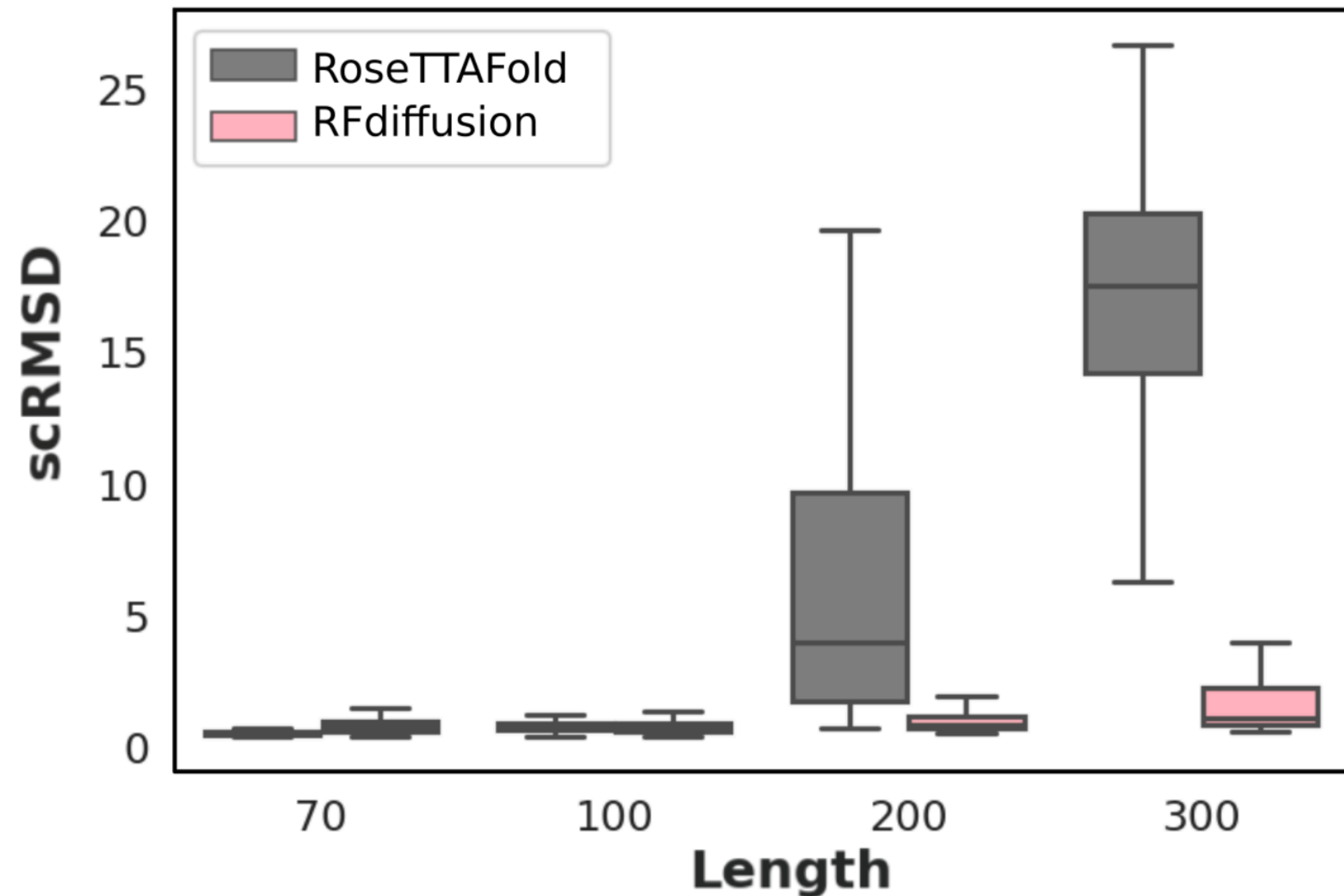
\* additional auxiliary losses are used. See supplementary section 4.

# RosettaFold diffusion

▸ Improved unconditional sampling

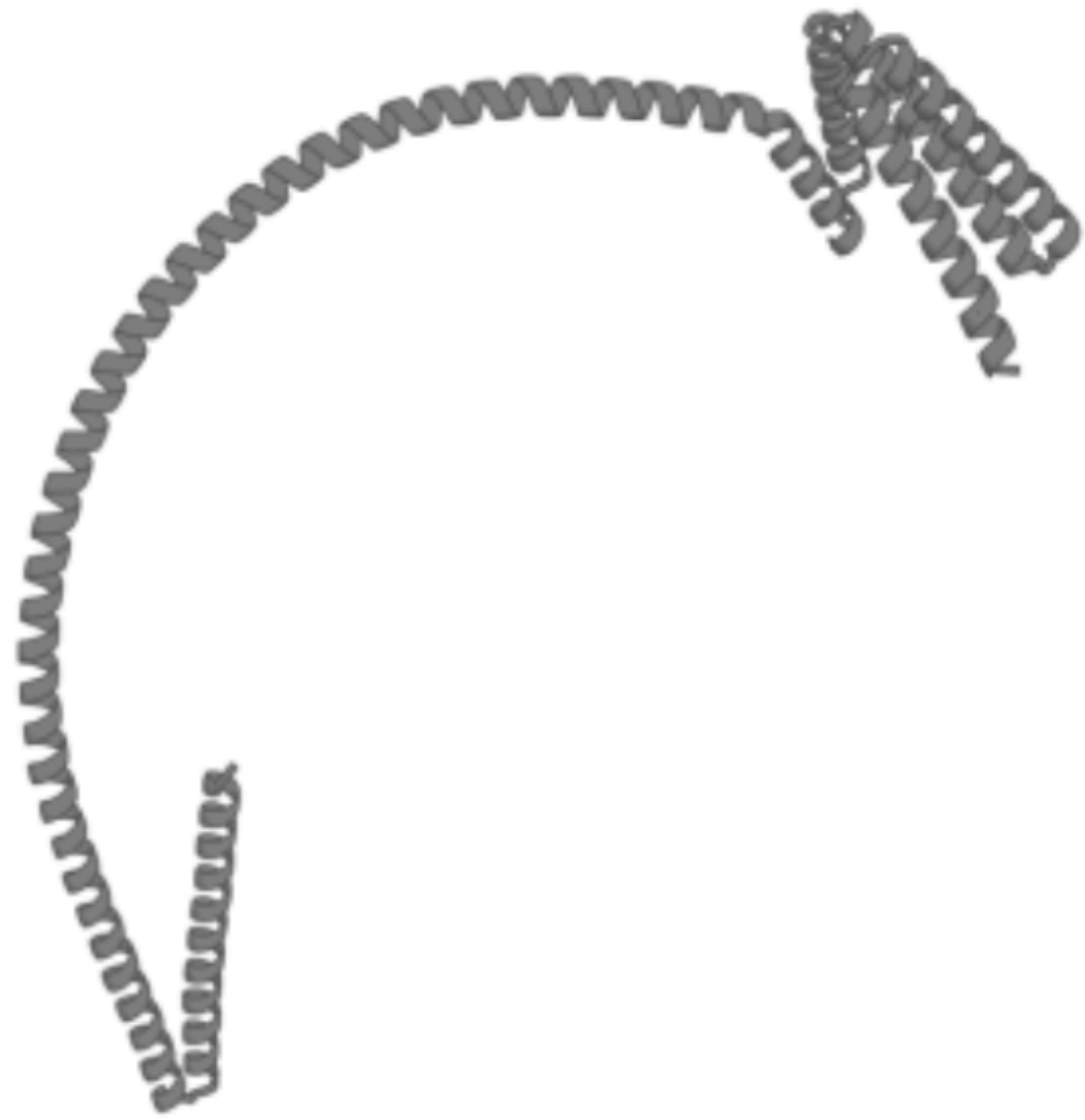


▸ Diffusion training is necessary

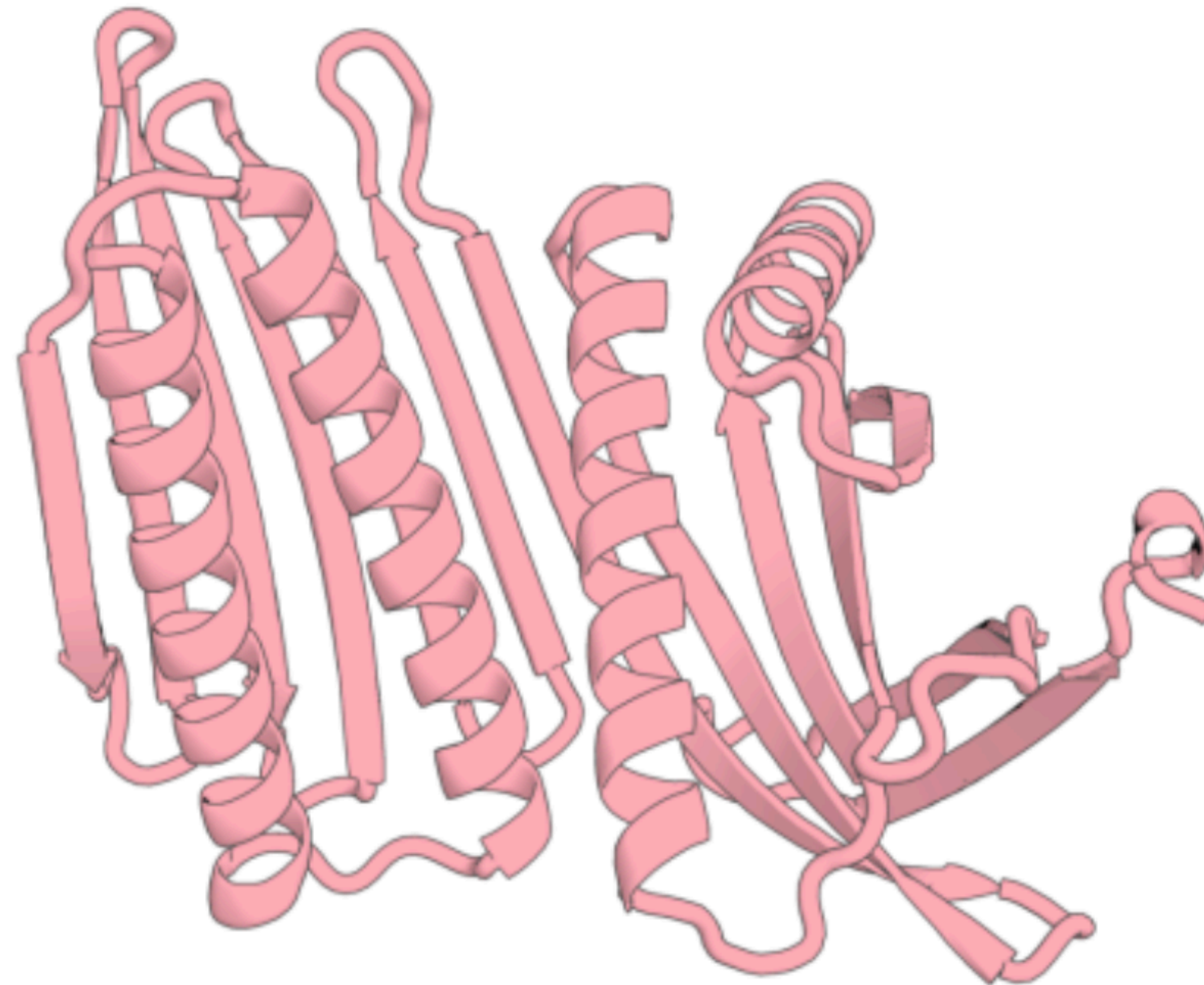


# Importance of pre-training RF

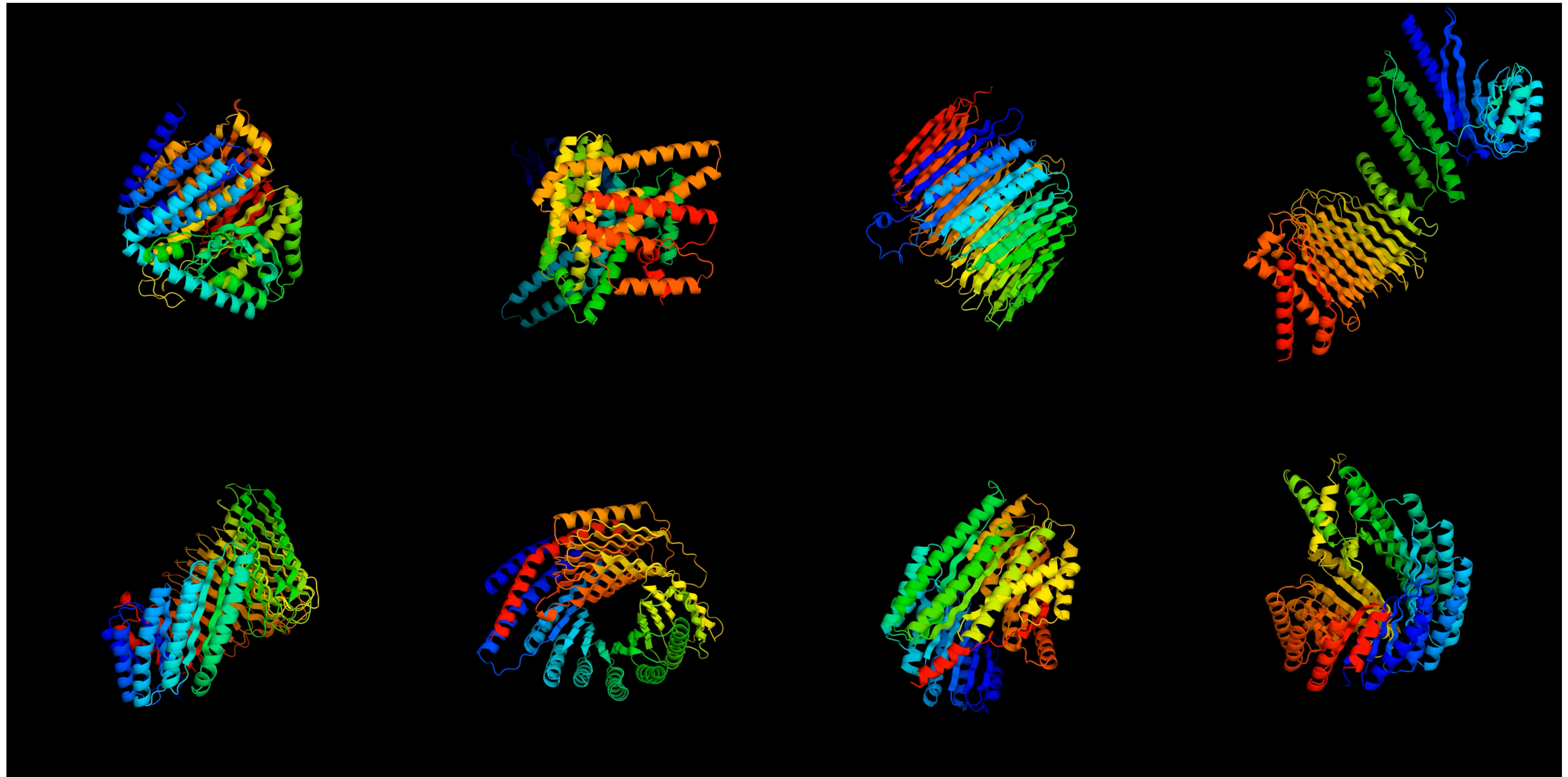
Without pre-training RF  
on protein folding



With pre-training

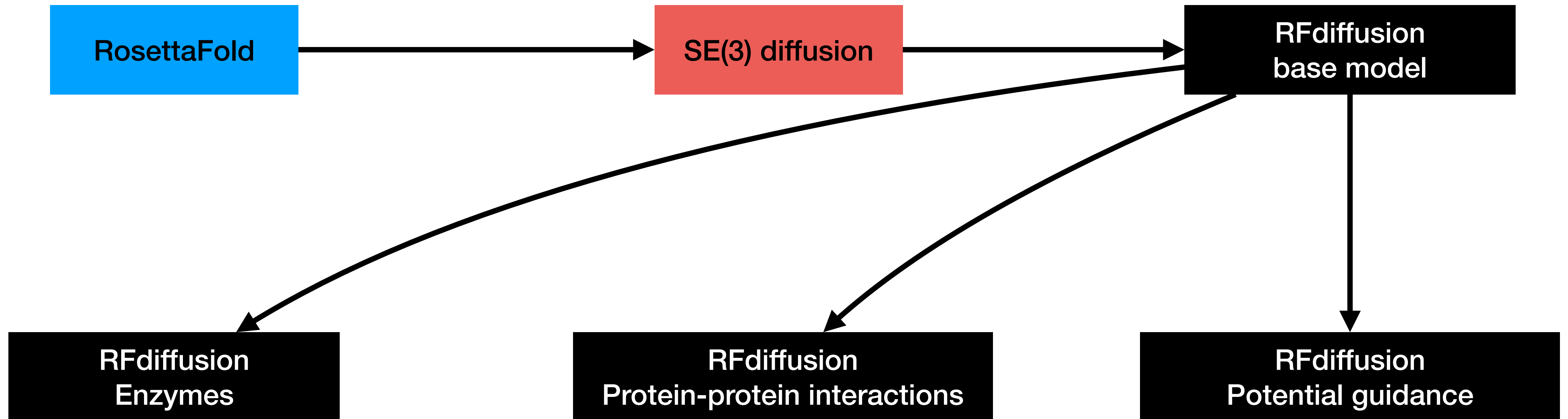


# Creative generation of large and diverse proteins



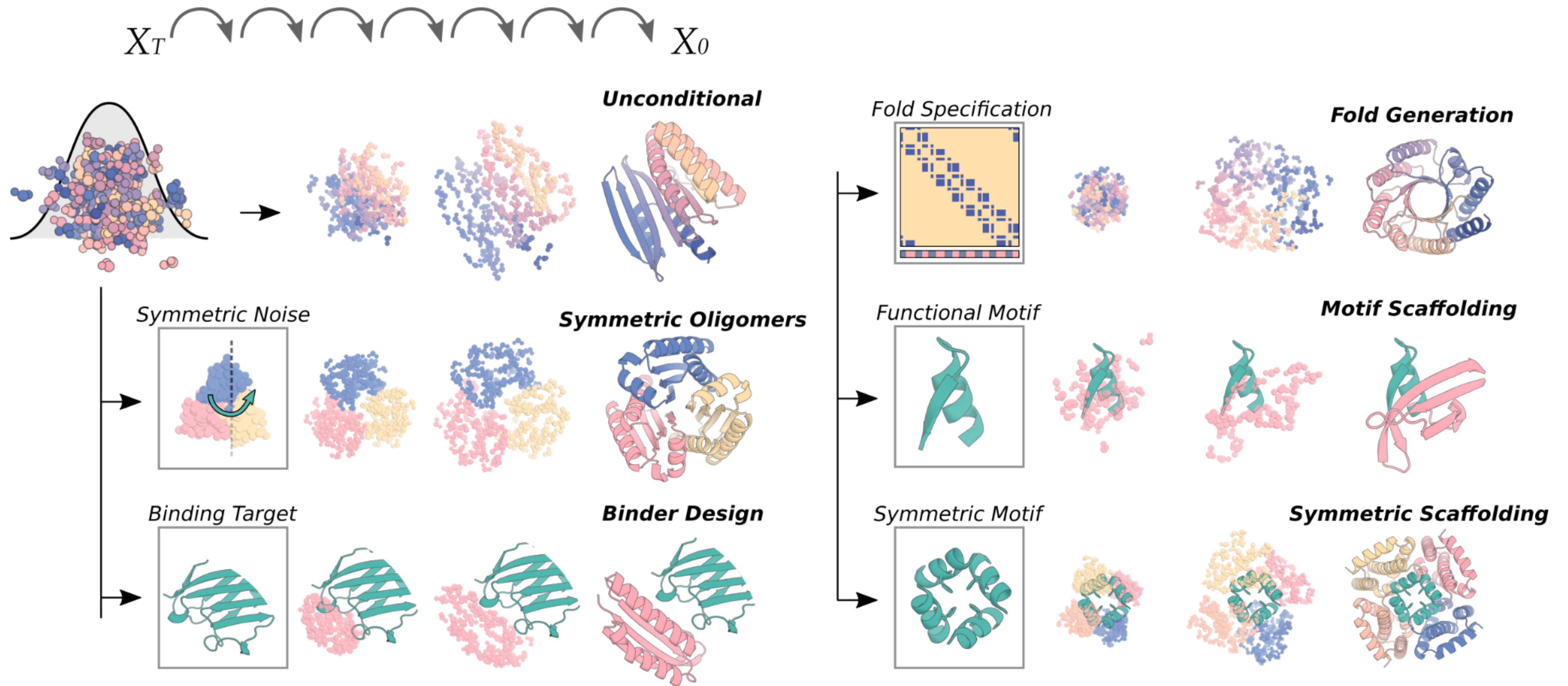
# RosettaFold diffusion

- **Improvement #2:** conditional training
  - *Fine tune* base model for different capabilities.



For fine-tuning details see supplementary section 4.2-4.4.

# RFdiffusion can be used across many tasks



# Conditional diffusion guidance

How to guide structures towards specific functions and higher quality?

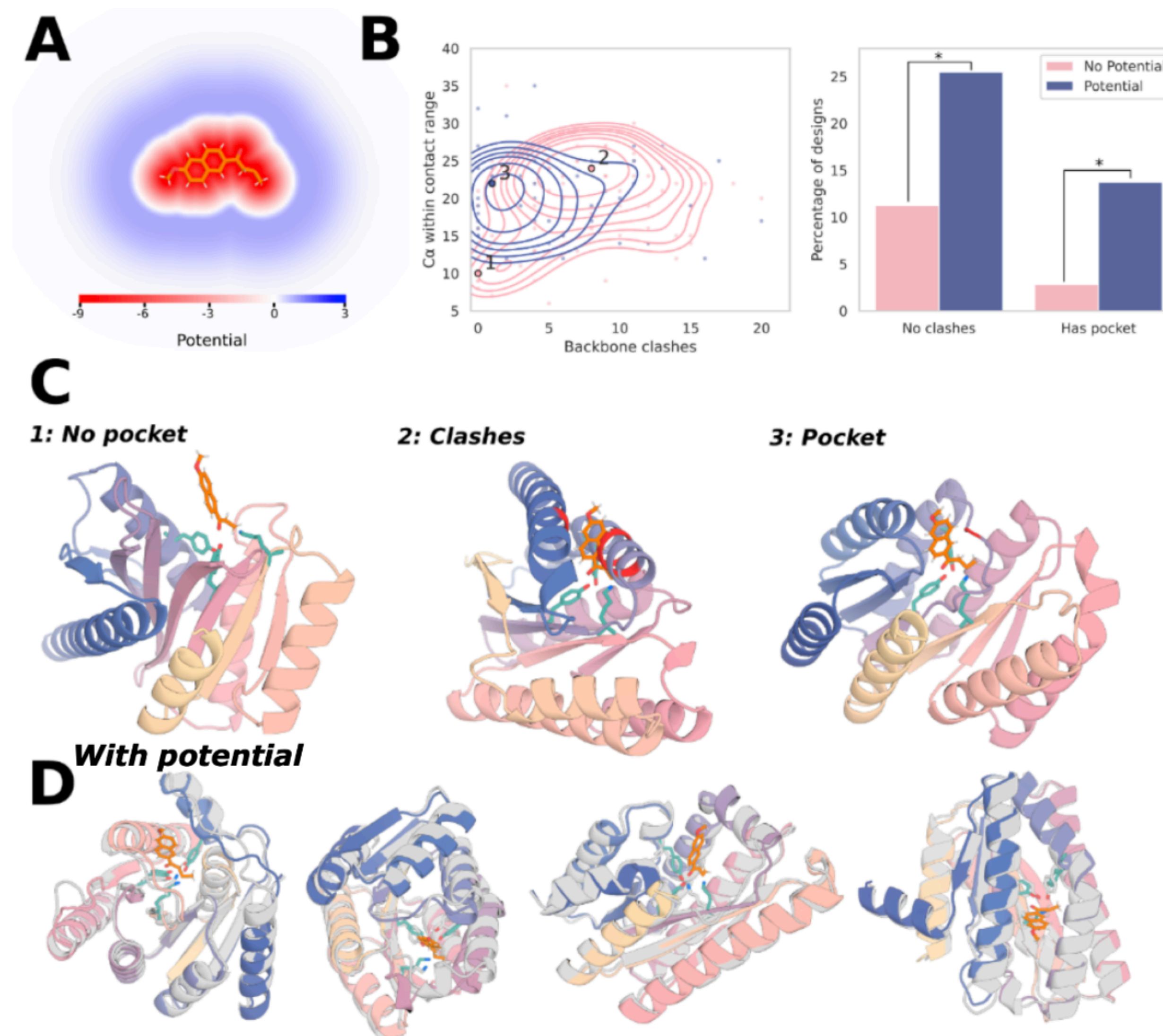
**Solution:** Inspired by classifier guidance, guide with potentials.

*Classifier guidance:*

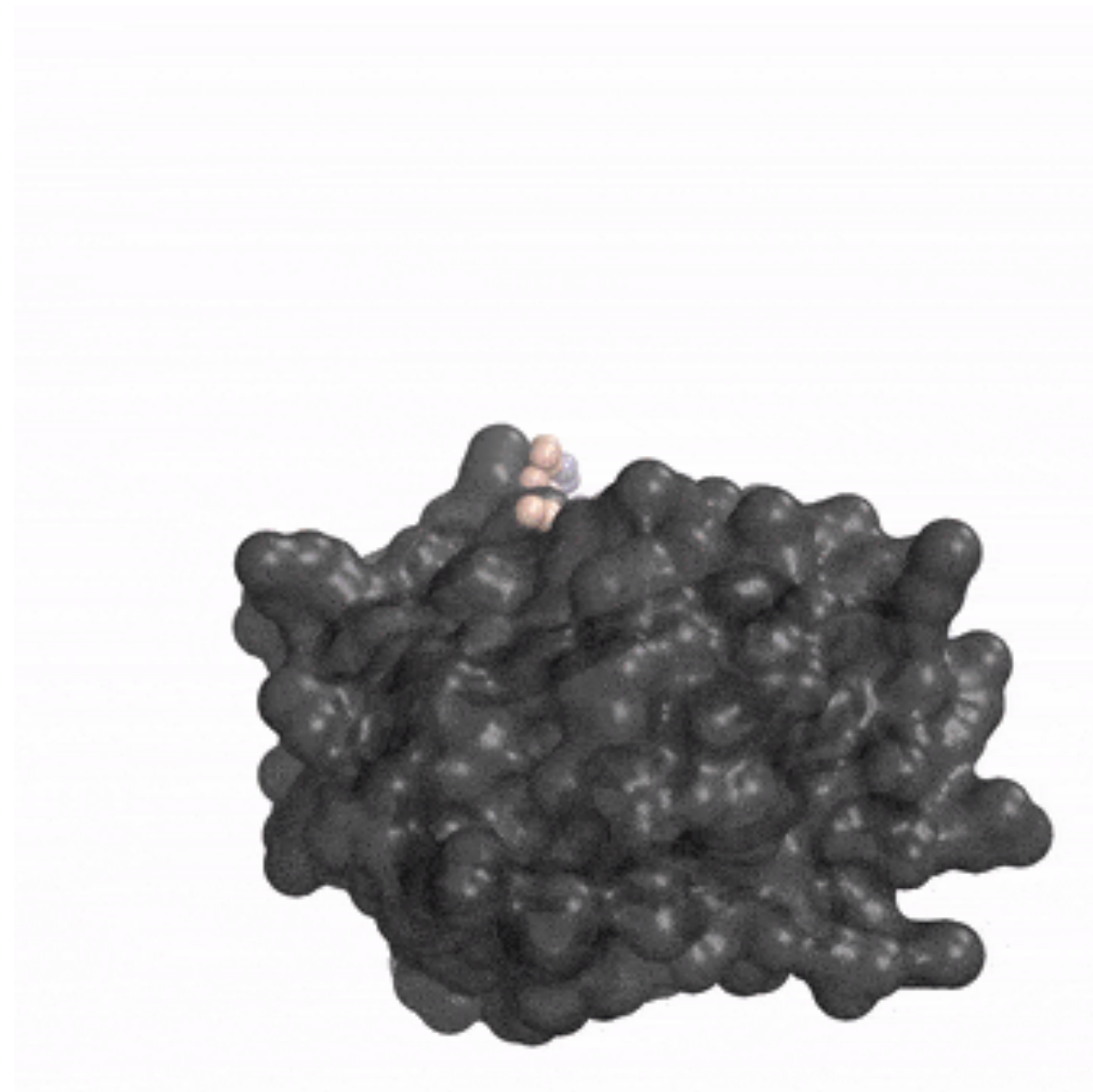
$$\nabla_{x^{(t)}} \log p(x^{(t)}) + \omega \nabla_{x^{(t)}} \log p(y = 1 | x^{(t)})$$

*Potential guidance:*

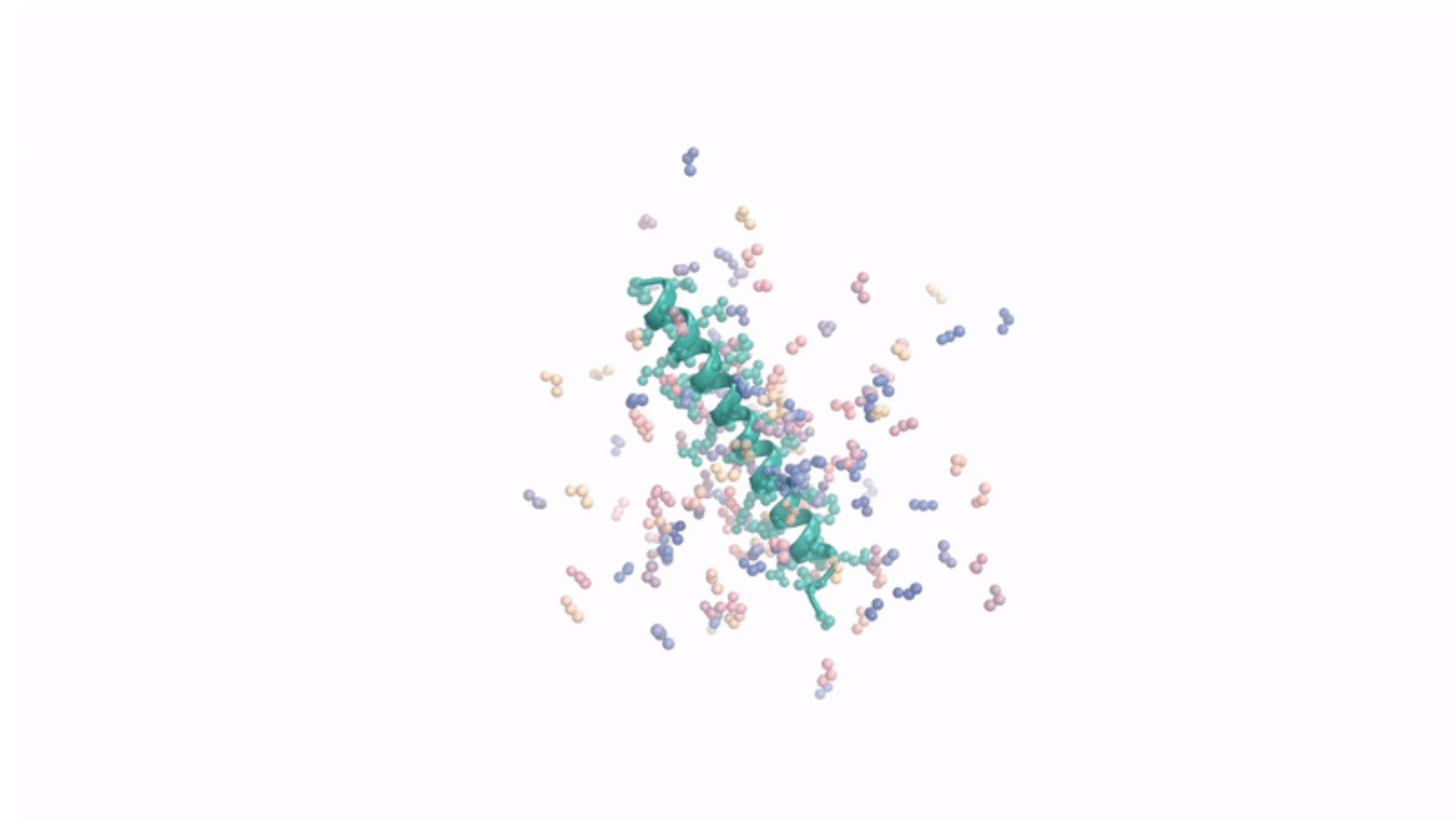
$$\nabla_{x^{(t)}} \log p(x^{(t)}) + \omega \nabla_{x^{(t)}} P(x^{(t)})$$



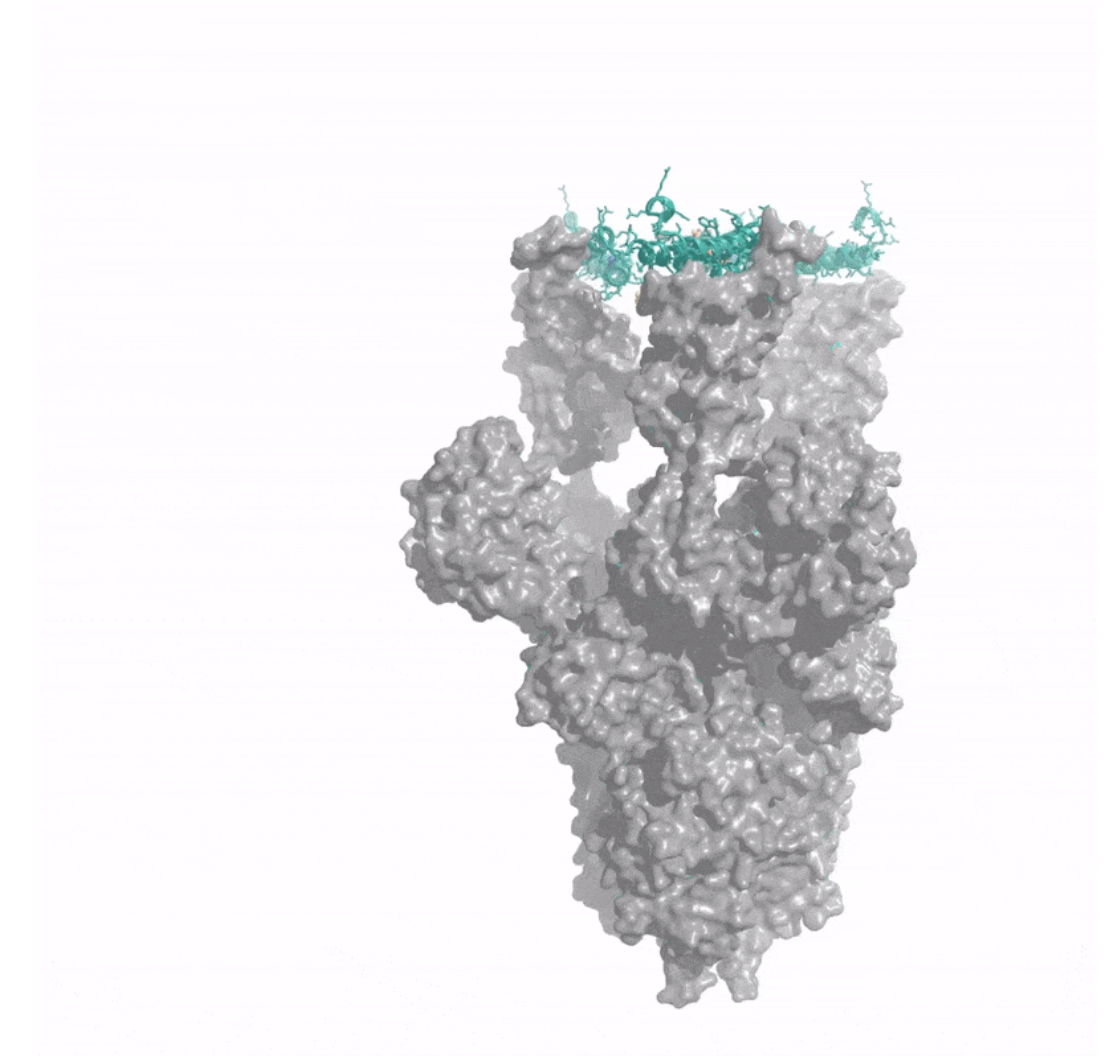
# Versatility across important applications



Binder generation



Scaffolding

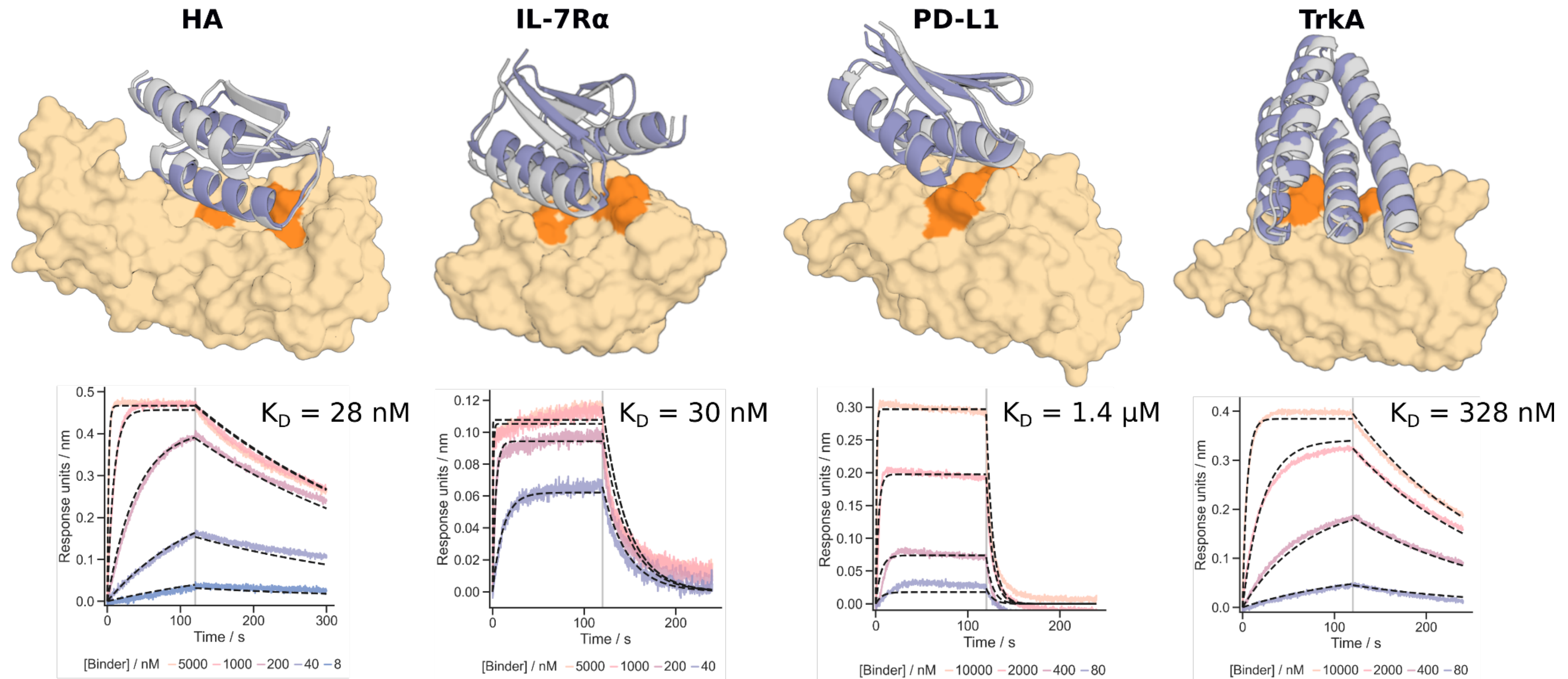


Symmetric complex  
binder and scaffolding

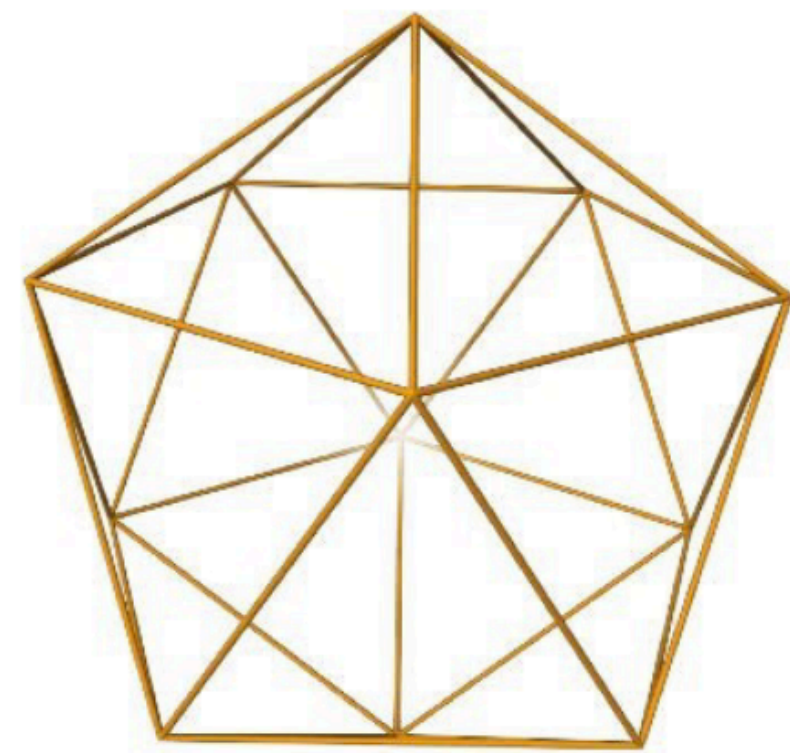


# Wet-lab validation

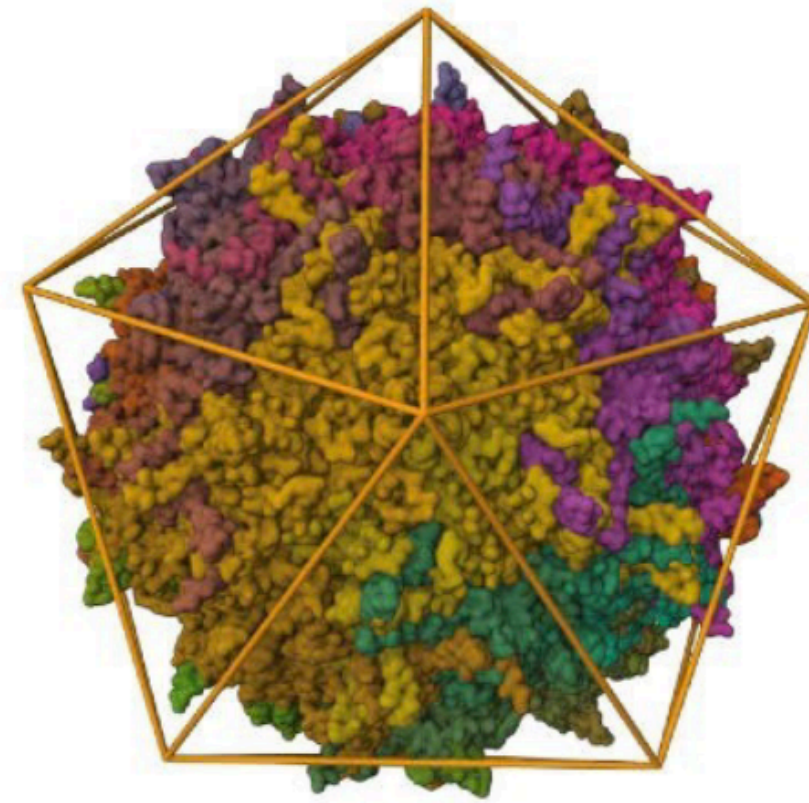
## De novo binder design



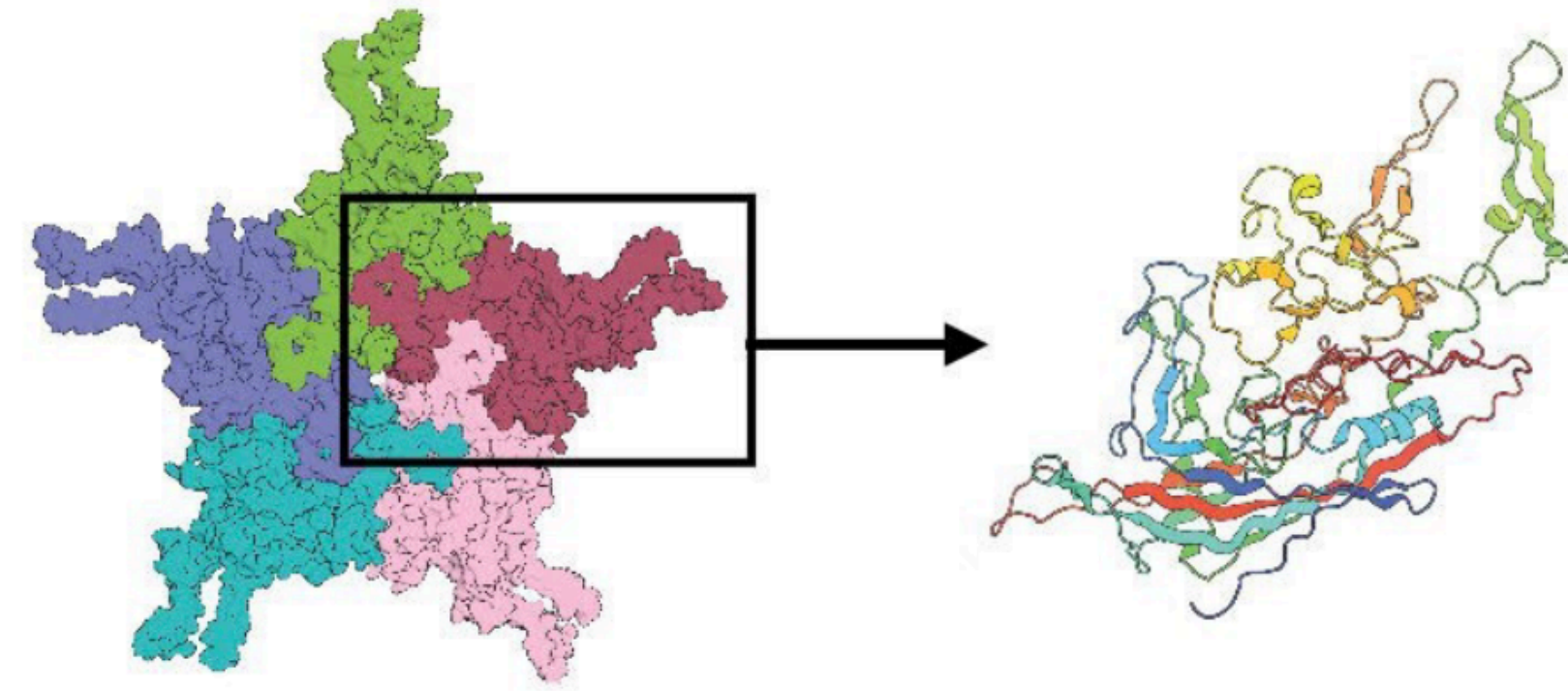
# Symmetric complex design



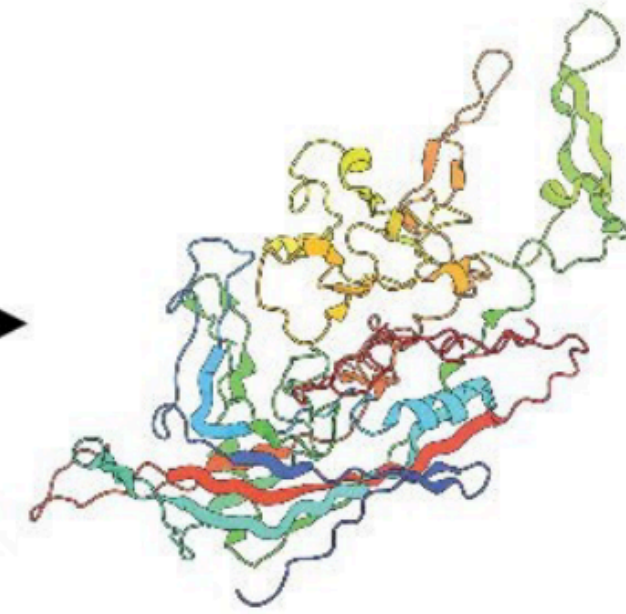
(a) Regular icosahedron



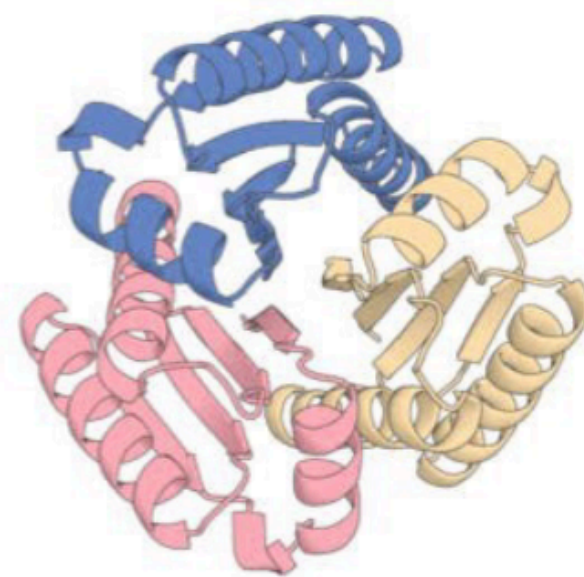
(b) AAV-DJ structure PDB: 3J1Q



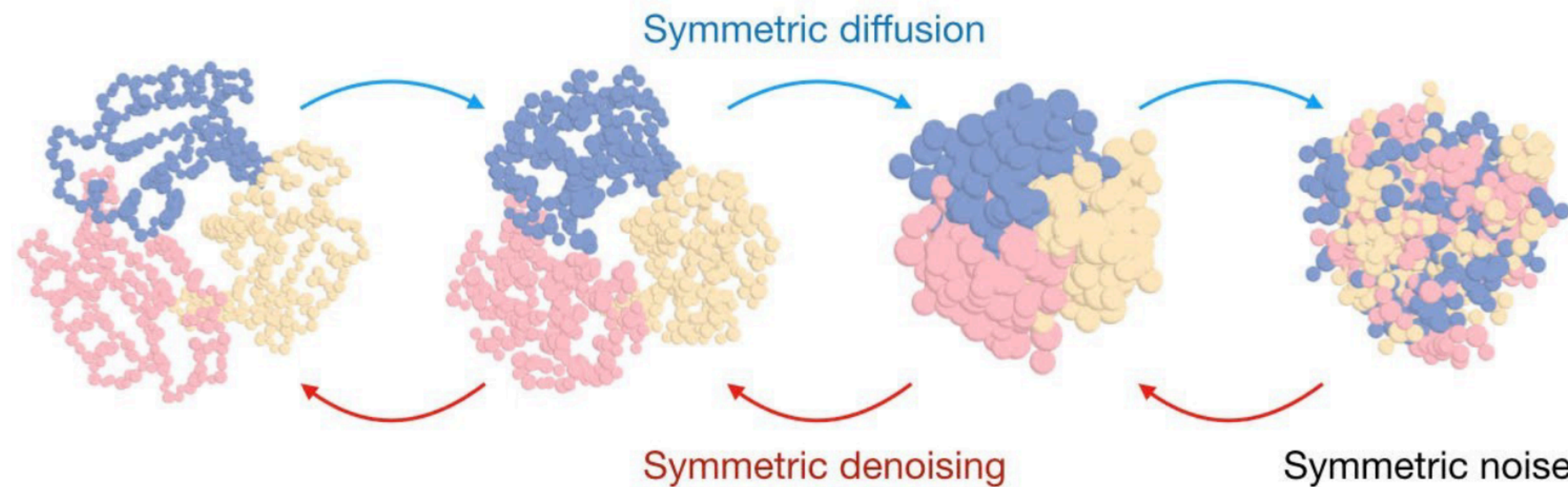
(c) 1 of 12 pentamer faces on AAV-DJ



(d) Asymmetric unit of AAV-DJ



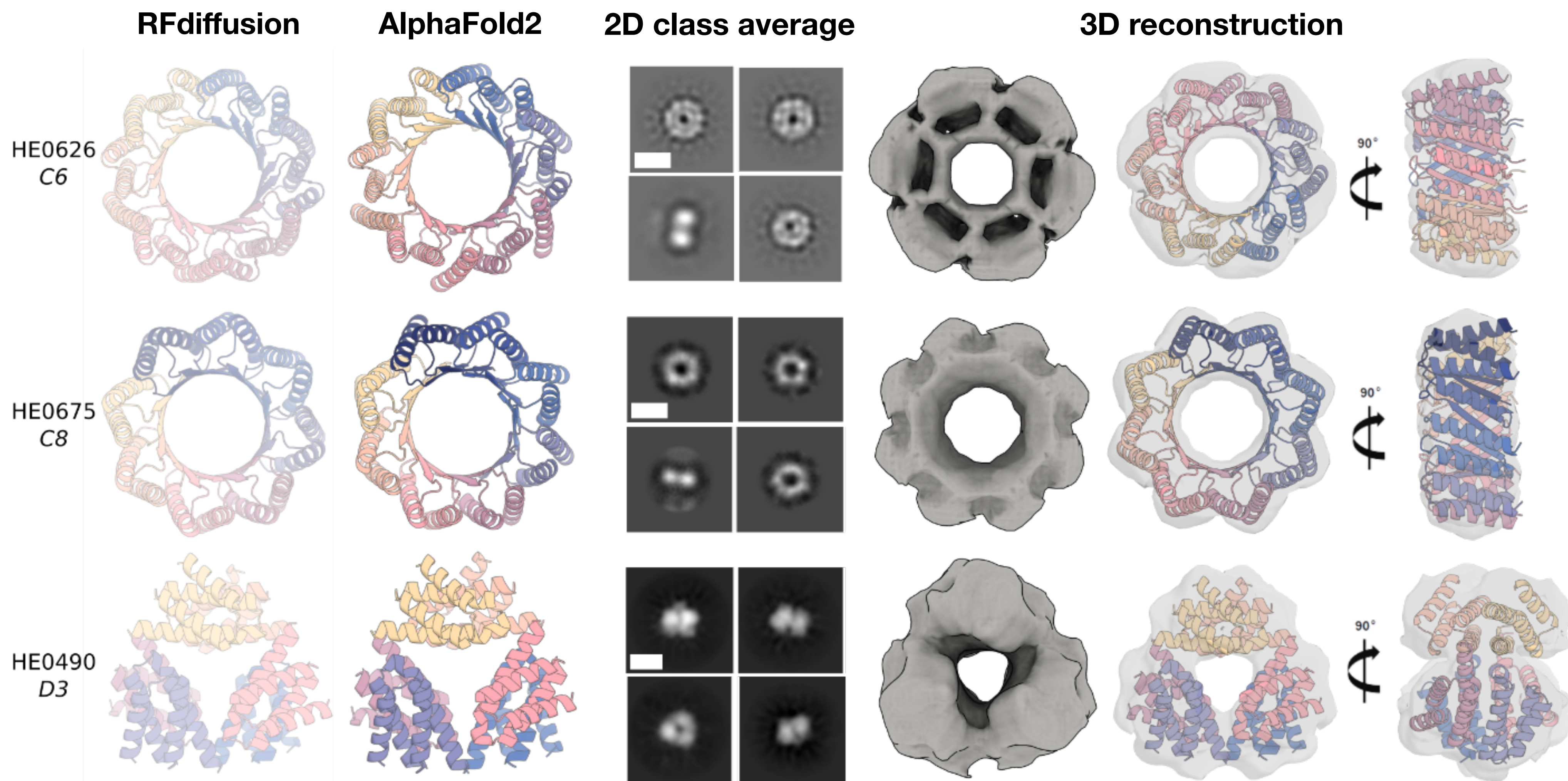
(e) C3 symmetric complex



(f) Symmetric noising with SDE and symmetric denoising with neural network

# Wet-lab validation

## Symmetric complex design



# Future directions

- **All-atom:** data, architecture, generative framework.
- **Lightweight, faster, tunable:** pre-training? fine-tuning? 1-step generation?

# FrameFlow

**Paper: Fast protein backbone generation with SE(3) flow matching**

**Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, Jose Jimenez-Luna, Sarah Lewis, Victor Garcias Satorras, Bastiaan S. Veiling, Regina Barzilay, Tommi Jaakkola, Frank Noe**

# Flow matching

## Lipman et al 2022

- Learn a Continuous Normalizing Flow (CNF)  $\phi_t$  by learning a time-dependent vector field  $v_t$ .

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x))$$
$$\phi_0(x) = x$$

- Parameterizes a probability density path

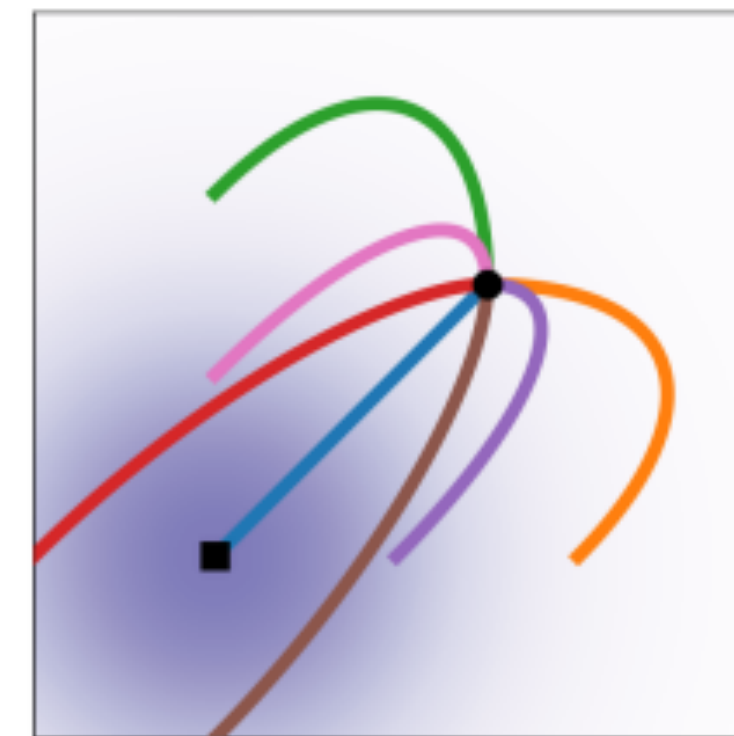
$$p_t = [\phi_t]_* p_0$$

$$[\phi_t]_* p_0(x) = p_0(\phi_t^{-1}(x)) \det \left[ \frac{\partial \phi_t^{-1}}{\partial x}(x) \right]$$

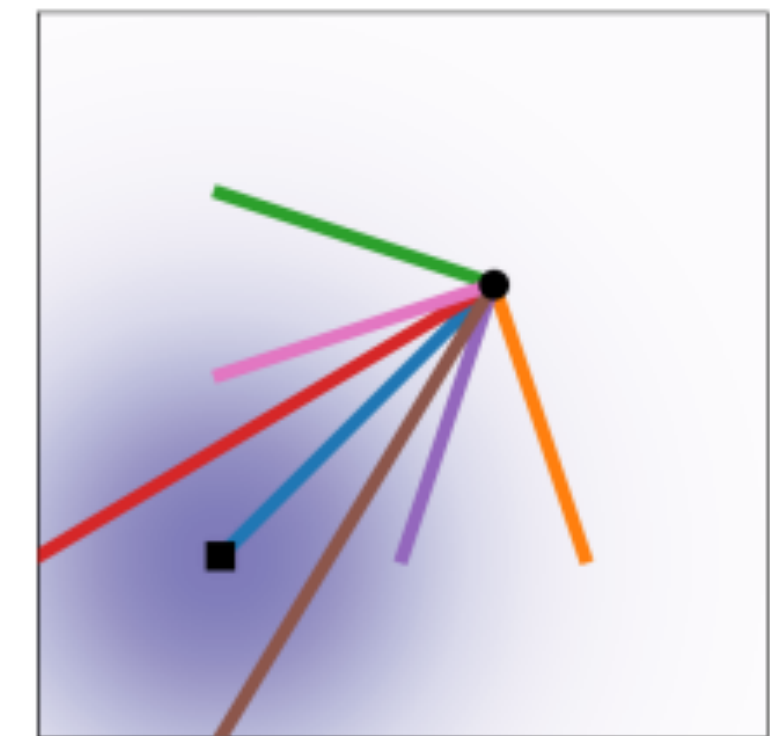
- Flow matching objective

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, p_t(x)} \|v_t(x) - u_t(x)\|^2$$

- Connection with diffusion.
  - Probability flow ODE is a CNF.
  - VP / VE diffusion have CNF equivalent.
  - Each instance corresponds to different path/schedule.



Diffusion path

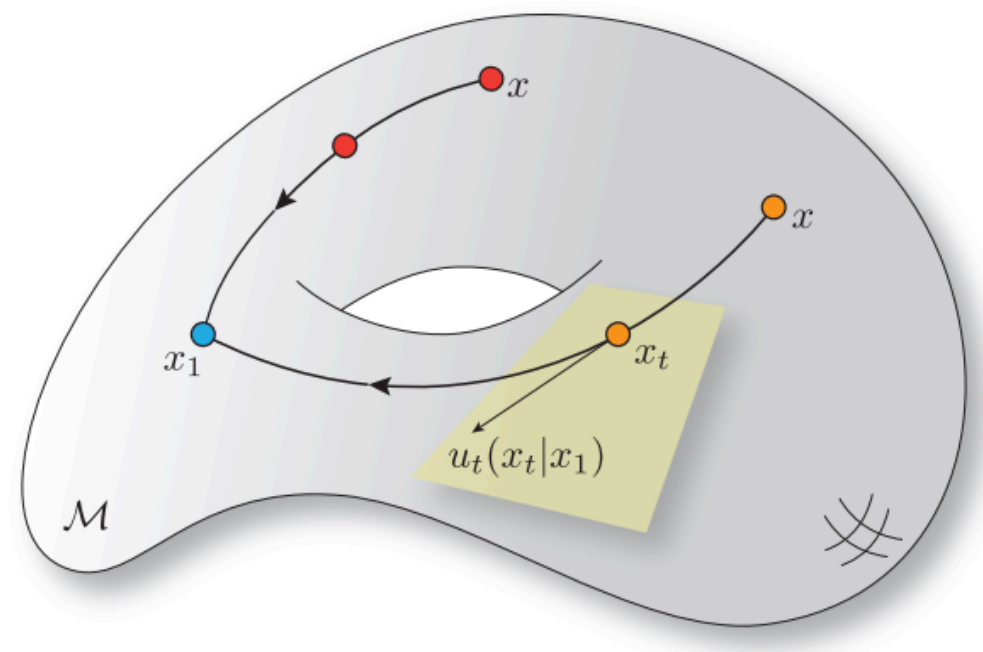


Straight (optimal transport)  
flow matching path

# Riemannian flow matching

Chen et al 2023

- Flow matching over general manifolds.



- Practical differences to diffusion.

- No need to solve SDE and define heat kernel.
- Only requires solving geodesic.

$$x_t = \exp_{x_0}(t \log_{x_0}(x_1)), \quad t \in [0, 1]$$

- And vector field.

$$u_t(x|x_1) = \frac{d \log \kappa(t)}{dt} d(x, x_1) \frac{\nabla d(x, x_1)}{\|\nabla d(x, x_1)\|_g^2}$$

	Simulation-free on simple geo.	Closed-form target vector field	Does not require divergence
Ben-Hamu et al. [5]	✓	-	✗
De Bortoli et al. [15] (DSM)	✗	✗	✓
De Bortoli et al. [15] (ISM)	✗	-	✗
Huang et al. [28]	✗	-	✗
Riemannian FM ( <i>Ours</i> )	✓	✓	✓

# Relative schedule

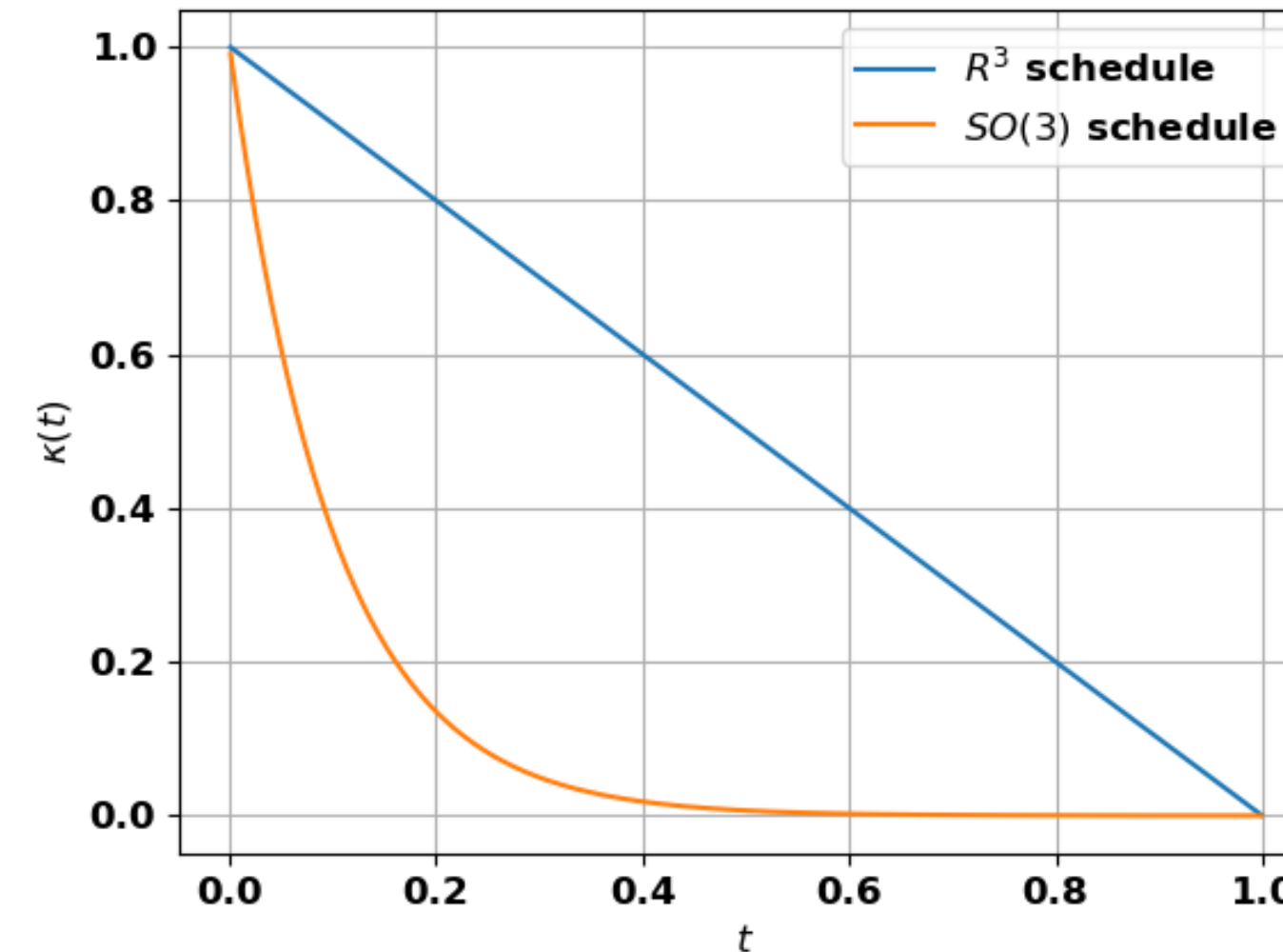
- SE(3) flow matching is flow matching on  $R^3$  and  $SO(3)$ .

$$\text{Translations } (\mathbb{R}^3) : x_t = (1 - t)x_0 + tx_1$$

$$\text{Rotations } (SO(3)) : r_t = \exp_{r_0} (t \log_{r_0}(r_1))$$

- Here both  $R^3$  and  $SO(3)$  follow the same linear schedule  $\kappa(t) = 1 - t$
- We found having a different rotation schedule  $\kappa(t) = e^{-tc}$  to be **crucial** for good performance.

$$r_t = \exp_{r_0} \left( (1 - e^{-ct}) \log_{r_0}(r_1) \right)$$
$$v_r^{(n)} = c \log_{r_t^{(n)}} \left( \hat{r}_1^{(n)} \right).$$





# Sample quality with different schedules

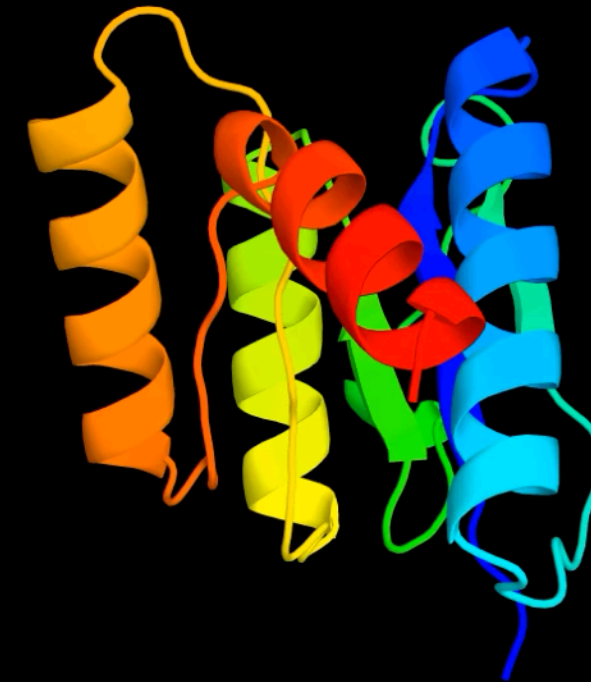
Translations:  $\kappa(t) = 1 - t$ ,

Rotations:  $\kappa(t) = 1 - t$



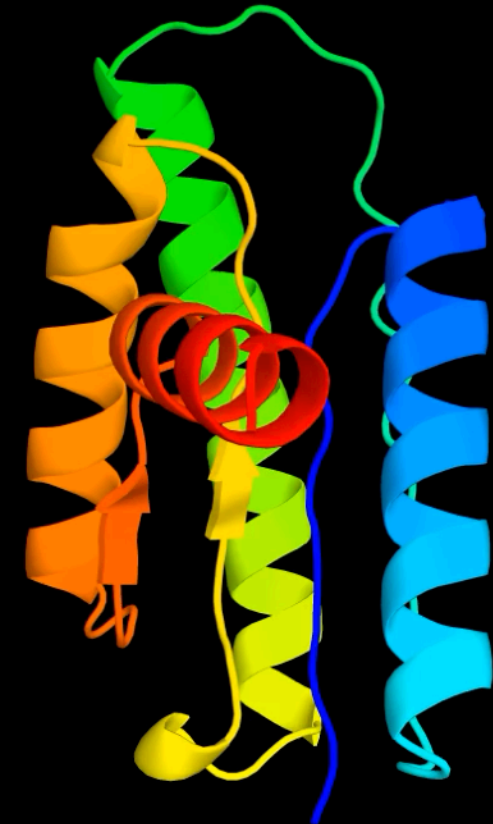
Translations:  $\kappa(t) = 1 - t$ ,

Rotations:  $\kappa(t) = e^{-5t}$



Translations:  $\kappa(t) = 1 - t$ ,

Rotations:  $\kappa(t) = e^{-10t}$

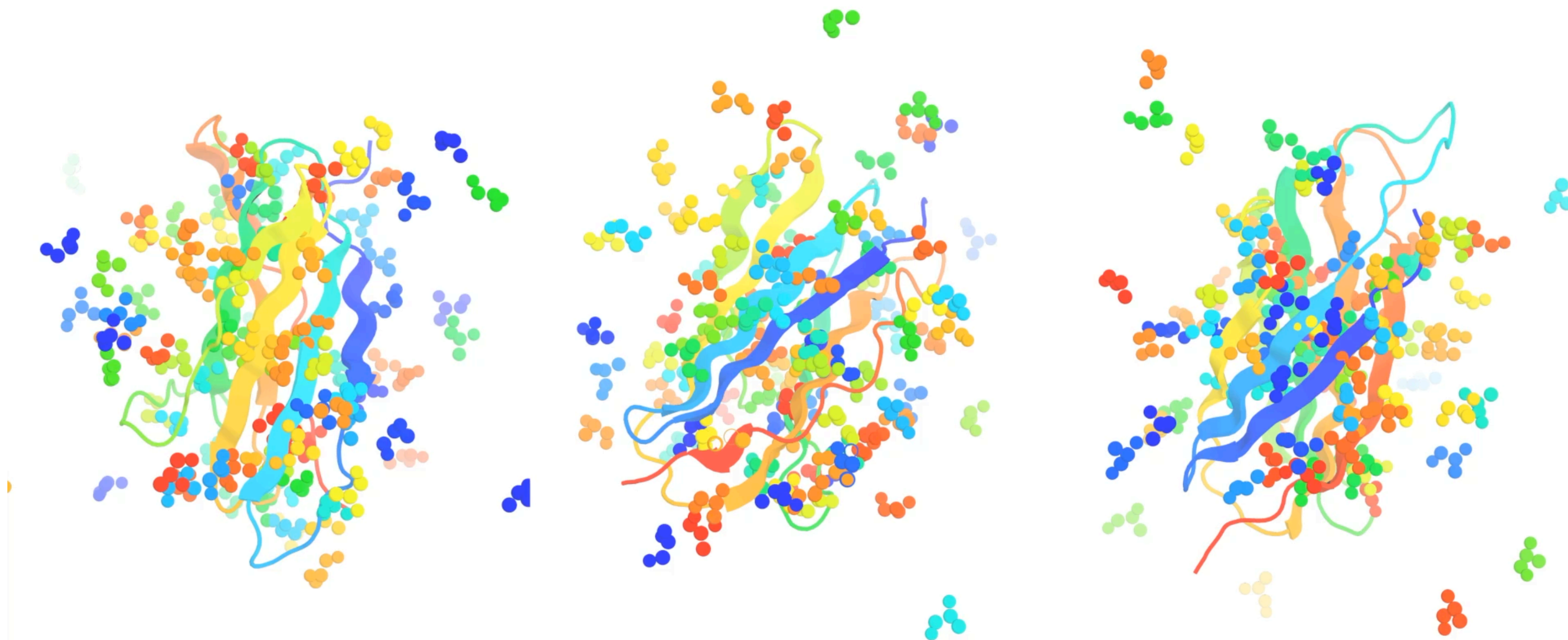


Translations:  $\kappa(t) = 1 - t$ ,

Rotations:  $\kappa(t) = e^{-100t}$



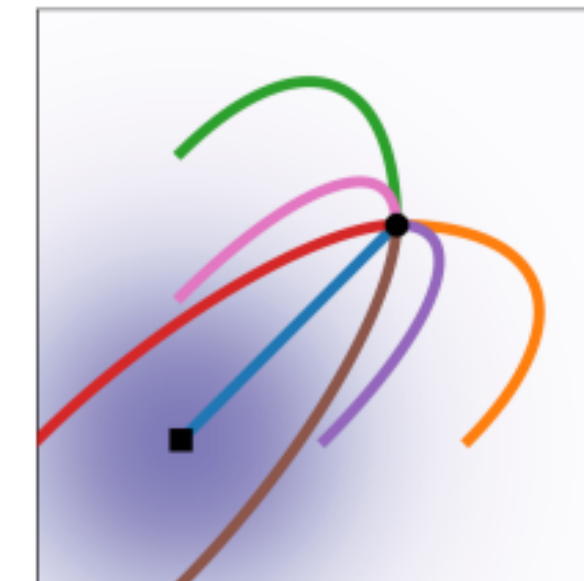
# Trajectories



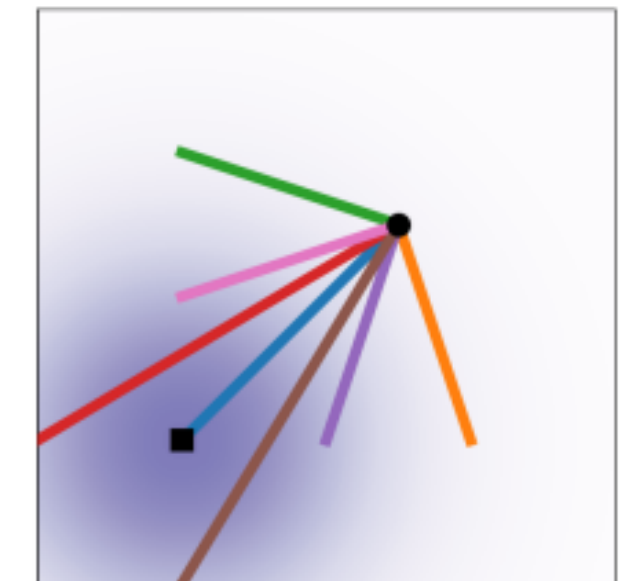
Diffusion SDE

Diffusion ODE

Flow ODE



Diffusion path



Straight (optimal transport)  
flow matching path

# Results

## Takeaways

1. Flow matching can sample with 100 or 10 time steps.
2. Sample quality is higher.

Model		Timesteps	Designability ( $\uparrow$ )
GENIE	SDE	1000	0.22
		750	0.11
		500	0.0
FrameDiff	SDE	500	0.42
		100	0.39
FrameDiff	ODE	100	0.34
		10	0.17
FrameFlow	ODE	500	0.81
		100	0.77
		10	0.33

# Outstanding questions

- **Learning ODE/SDEs on multimodal data.**
- **SDE vs. ODE**
- **Scheduling**

# Acknowledgements

- **FrameDiff** collaborators: Brian Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet.
- **RFdiffusion** collaborators: Joseph Watson, David Juergens, Nathaniel Bennett, Helen Eisenach, Woody Ahern, ..., David Baker.
- **FrameFlow** collaborators: Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, Jose Jimenez-Luna, Sarah Lewis, Victor Garcias Satorras, Bastiaan S. Veiling, Regina Barzilay, Tommi Jaakkola, Frank Noe.
- **Advisors:** Regina Barzilay, Tommi Jaakkola

