Graph Denoising Diffusion for Inverse Protein Folding

Kai Yi

UNSW Data Science Hub (uDASH)
School of Mathematics and Statistics
University of New South Wales

UNSW SYDNEY

uDASH UNSW Data Science Hub

Joint with Bingxin Zhou (SJTU), Yiqing Shen (JHU), Pietro Liò (Cambridge), Yu Guang Wang (UNSW)

Outline

- Introduction
- Protein Graph Representation & Inverse Folding
- Graph Denoising Diffusion for Inverse Protein Folding (GraDe-IF)
- Experiment & Evaluation

Outline

- Introduction
- Protein Graph Representation & Inverse Folding
- Graph Denoising Diffusion for Inverse Protein Folding (GraDe-IF)
- Experiment & Evaluation

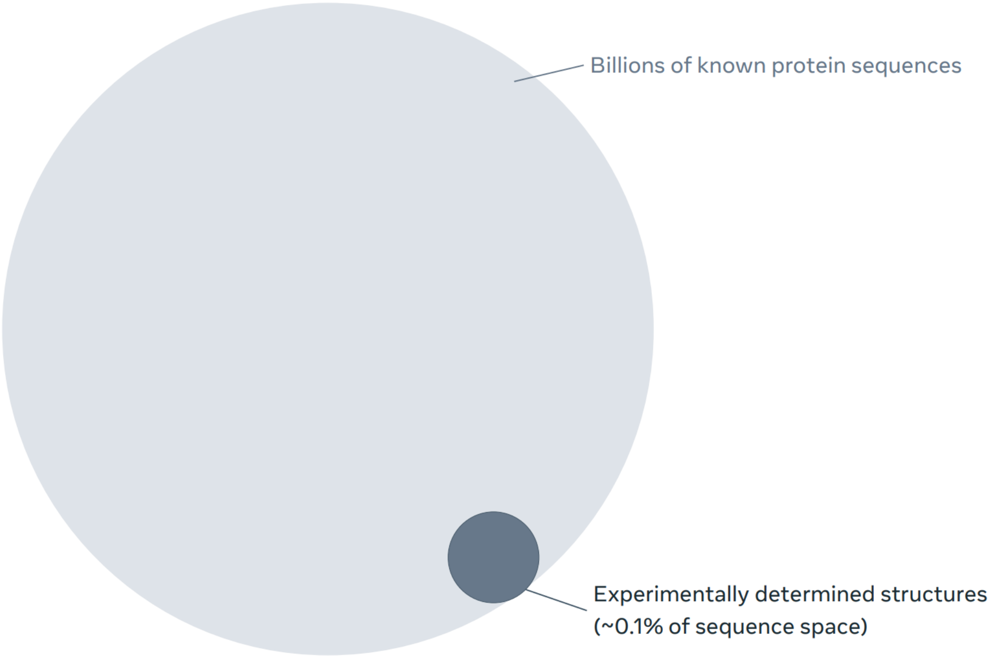
Introduction

www.nature.com/nmeth / January 2022 Vol.19 No. 1
nature methods
Method of the Year 2021:
Protein structure prediction

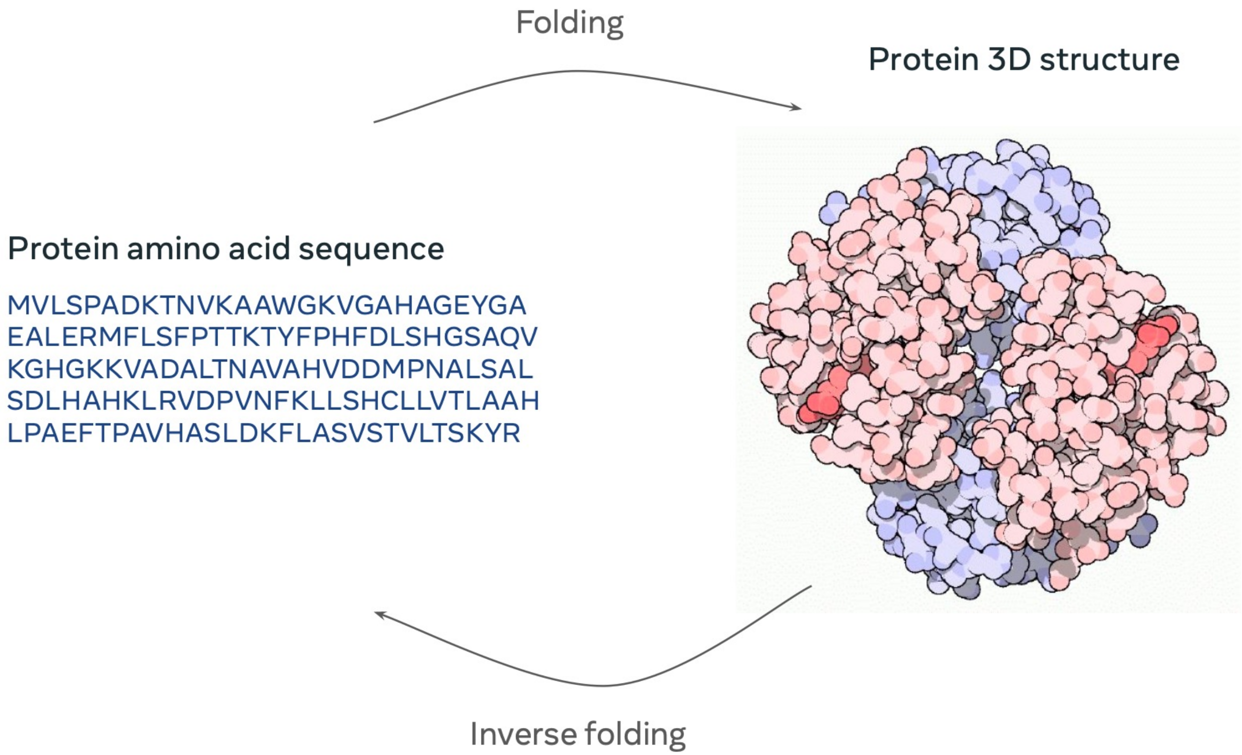
AlphaFold2 (2021)

SARS-CoV-2 is evolving fast. What will it do next? p. 844
Effects on surface water from hydraulic fracturing pp. 853 & 896
Baby bats babble like human babies p. 923
Science
$15
20 AUGUST 2021
sciencemag.org
AAAS
PREDICTING
STRUCTURES
Deep learning accurately folds proteins p. 871

RoseTTAFold (2021)


Billions of known protein sequences
Experimentally determined structures
(~0.1% of sequence space)

Introduction



Folding

Protein 3D structure

Protein amino acid sequence

MVLSPADKTNVKAAWGKVGAHAGEYGA
EALERMFLSFPTTKTYFPHFDLSHGSAQV
KGHGKKVADALTNAVAHVDDMPNALSAL
SDLHAHKLRVDPVNFKLLSHCLLVTLAAH
LPAEFTPAVHASLDKFLASVSTVLTSKYR

Inverse folding

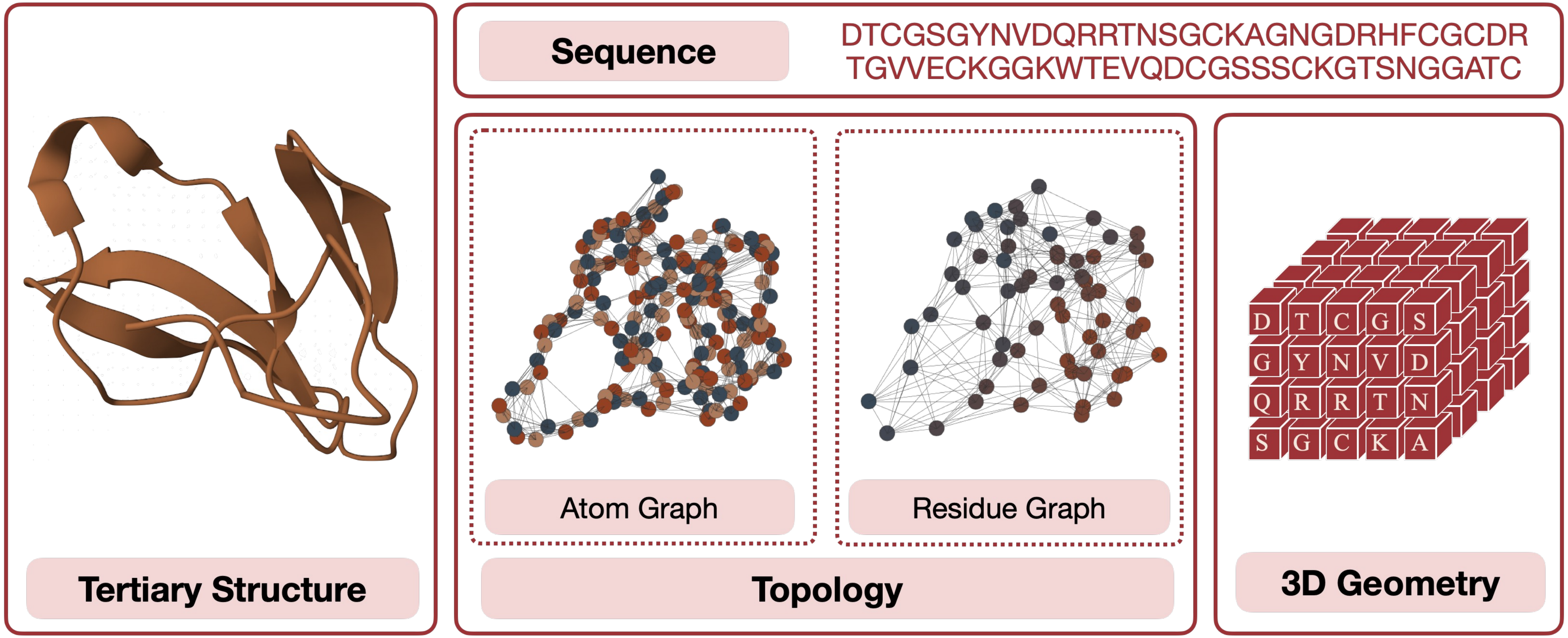
- Structure determine function
- Many sequence can fold on one similar structure (524 different sequence fold into 1MBN)
- Diffusion model can generate diverse sample given the same input

Hsu, Chloe, et al. "Learning inverse folding from millions of predicted structures." . PMLR, 2022.

Outline

- Introduction
- Protein Graph Representation & Inverse Folding
- Graph Denoising Diffusion for Inverse Protein Folding (GraDe-IF)
- Experiment & Evaluation

Protein Representation



Sequence

DTCGSGYNVDQRRTNSGCKAGNGDRHFCGCDR
TGVVECKGGKWTEVQDCGSSSCKGTSNGGATC

Atom Graph

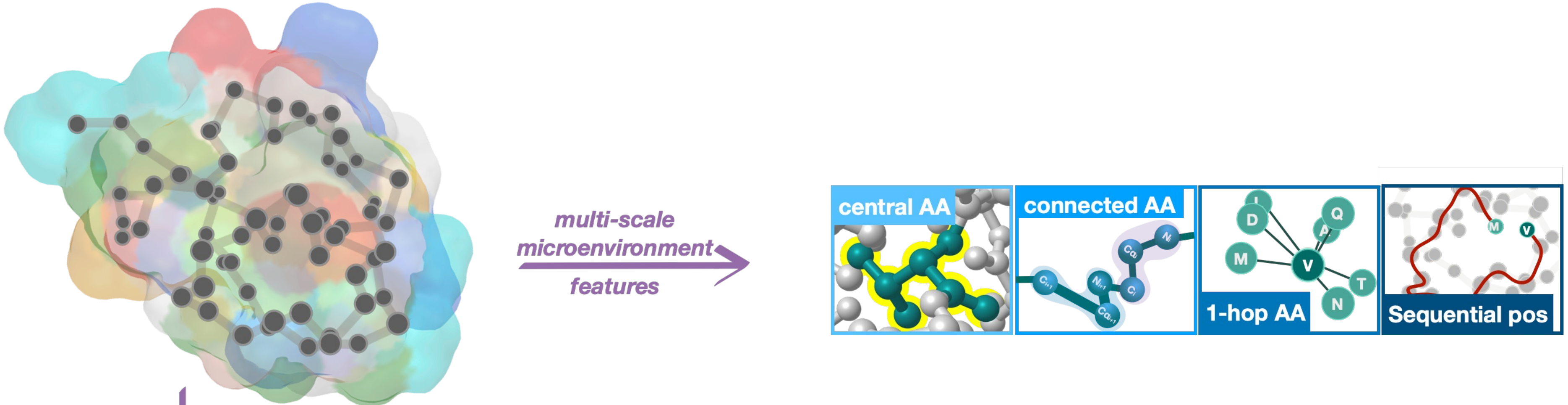
Residue Graph

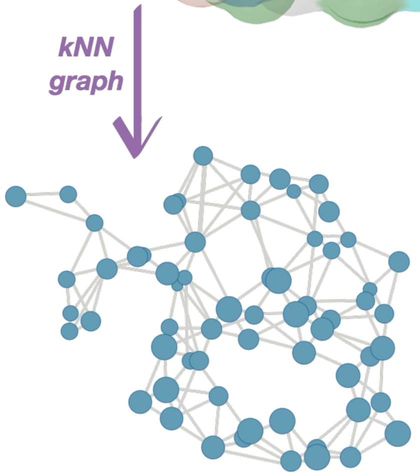
Tertiary Structure

Topology

3D Geometry

Protein Graph





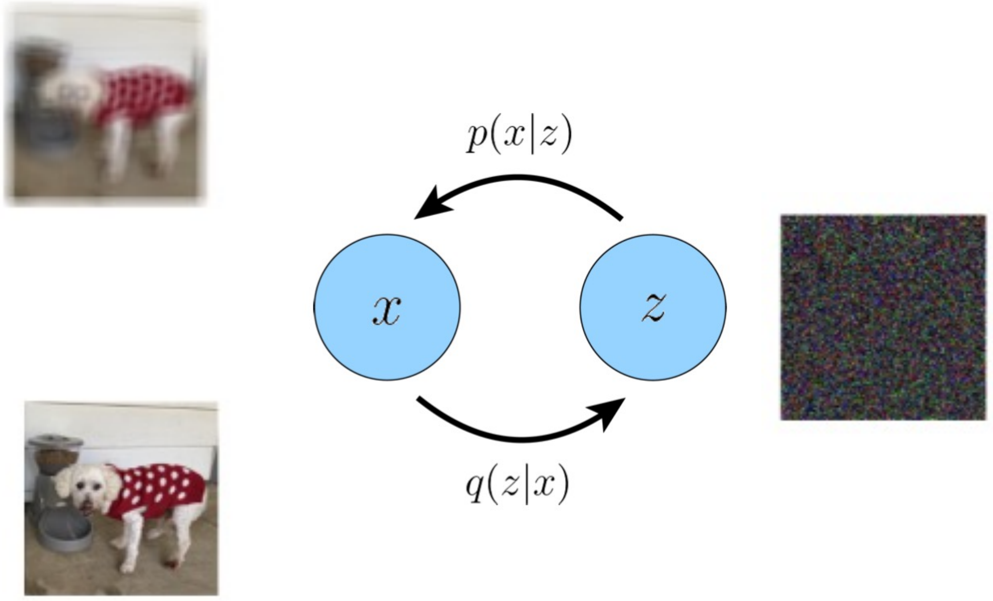
$$\mathcal{G} = \{X, A, E\}$$

Inverse Folding:

$$p(X^{aa} \mid X^{pos}) = p_\theta(X^{aa} \mid A, E) \quad \text{where} \quad X^{aa} \in \mathbb{R}^{N\times 20}$$

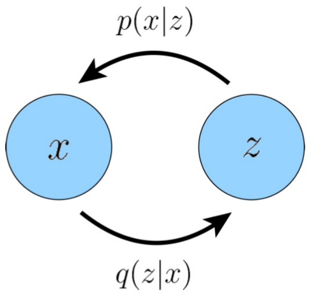
Diffusion Generative Model

$$
\begin{aligned}
\log p(\boldsymbol{x}) &= \log \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \\
&= \log \int \frac{p(\boldsymbol{x}, \boldsymbol{z}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} d\boldsymbol{z} \\
&= \log \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right] \\
&\geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})]}_{\text{reconstruction term}} - \underbrace{D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p(\boldsymbol{z})\right)}_{\text{prior matching term}}
\end{aligned}
$$

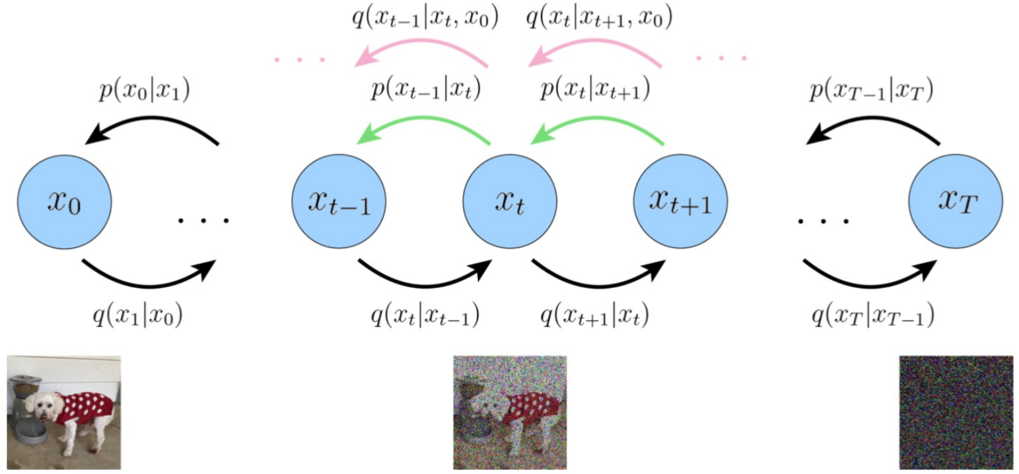


Diffusion Generative Model

$$\log p(\boldsymbol{x}) \quad \geq \mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})] - D_{KL}\left(q_{\phi}(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p(\boldsymbol{z})\right)$$



$$\begin{aligned}
\log\ p(\boldsymbol{x}) &= \log \int p(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \\
&= \log \int \frac{p(\boldsymbol{x}_{0:T}) q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} d\boldsymbol{x}_{1:T} \\
&= \log\ \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right] \\
&\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right] \\
&\dots \\
&= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log\ p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)] - D_{\mathrm{KL}}\left(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\right) - \\
&\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right)\right]
\end{aligned}$$



Outline

- Introduction
- Protein Graph Representation & Inverse Folding
- Graph Denoising Diffusion for Inverse Protein Folding (GraDe-IF)
- Experiment & Evaluation

Define Diffusion Process in Discrete Space

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)] - D_{\mathrm{KL}}\big(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)\big)$$

$$- \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}\big[D_{\mathrm{KL}}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\big)\big]$$

$$\arg \underset{\boldsymbol{\theta}}{min}\, \mathbb{E}_{t \sim U\{2,T\}}\Big[\mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}\big[D_{\mathrm{KL}}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\big)\big]\Big]$$

- $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ requires a closed form

- $q(\boldsymbol{x}_T)$ should independent with x

- $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\,,\ \boldsymbol{x}_0)$ should have closed form

Define Diffusion Process in Discrete Space

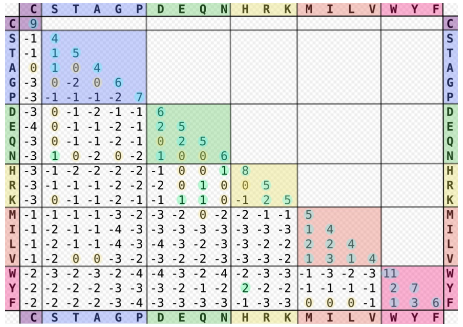
Diffusion Process:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}^{aa}) = \boldsymbol{x}^{aa}\overline{\boldsymbol{Q}}_t \qquad \overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_1\boldsymbol{Q}_2 \dots \boldsymbol{Q}_t \qquad x^{aa} \in \mathbb{R}^{1\times 20}$$

Transition Matrix

$$\boldsymbol{Q}_t = \alpha_t \boldsymbol{I} + (1-\alpha_t)1_d\, 1_d^{\top}/d$$

(Uniform Diffusion Kernel)



	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

(BLOSUM Matrix)

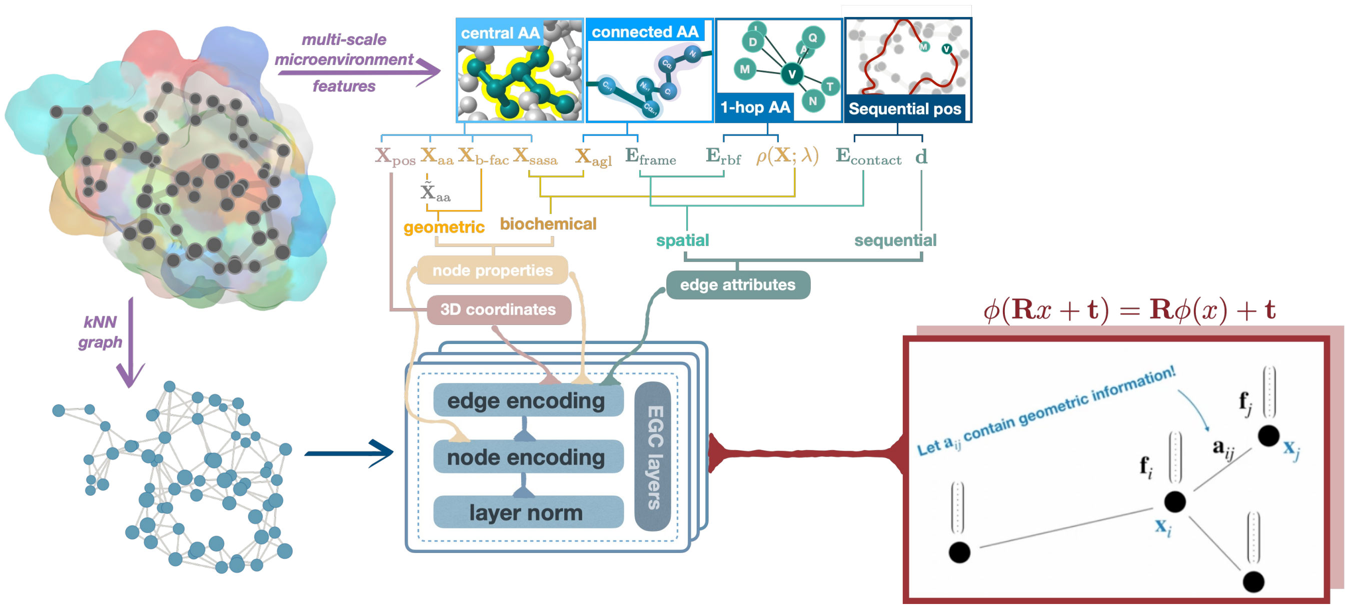
Posterior Distribution

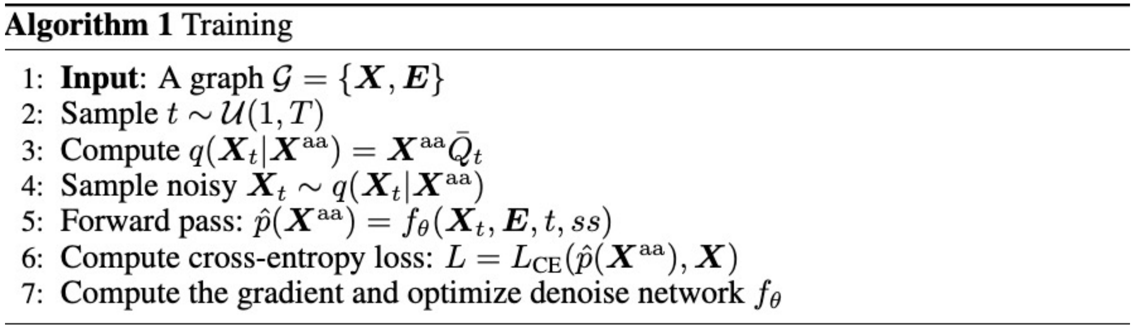
$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}^{aa}) = \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}^{aa})\, q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}^{aa})}{q(\boldsymbol{x}_t \mid \boldsymbol{x}^{aa})}$$

$$= \mathrm{Cat}\left(\boldsymbol{x}_{t-1} \mid \frac{\boldsymbol{x}_t Q_t^{\top} \odot \boldsymbol{x}^{aa}\bar{Q}_{t-1}}{\boldsymbol{x}^{aa}\bar{Q}_t \boldsymbol{x}_t^{\top}}\right)$$

Parameterized Generative Process : $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) \propto \sum_{\hat{\boldsymbol{x}}^{\mathrm{aa}}} q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}^{\mathrm{aa}})\hat{p}_\theta(\boldsymbol{x}^{\mathrm{aa}} \mid \boldsymbol{x}_t)$

Denoise Neural Network

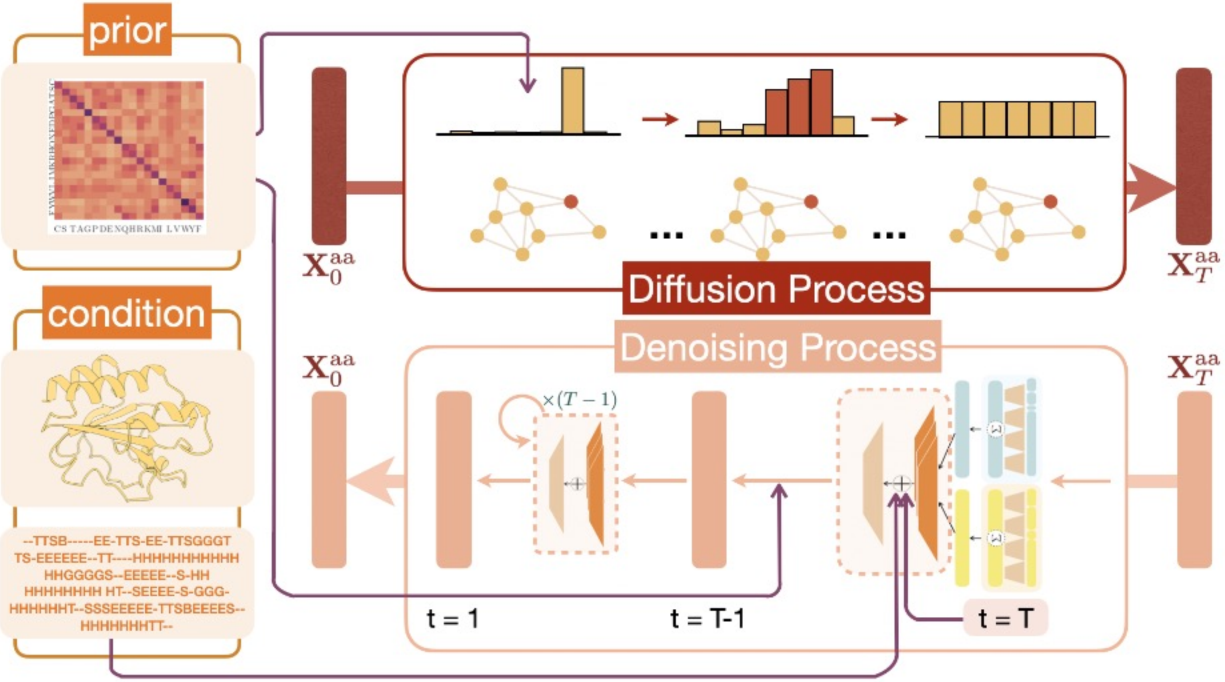

multi-scale microenvironment features
central AA
connected AA
1-hop AA
Sequential pos
$\mathbf{X}_{\text{pos}}$ $\mathbf{X}_{\text{aa}}$ $\mathbf{X}_{\text{b-fac}}$ $\mathbf{X}_{\text{sasa}}$ $\mathbf{X}_{\text{agl}}$ $\mathbf{E}_{\text{frame}}$ $\mathbf{E}_{\text{rbf}}$ $\rho(\mathbf{X};\lambda)$ $\mathbf{E}_{\text{contact}}$ $\mathbf{d}$
$\tilde{\mathbf{X}}_{\text{aa}}$
geometric
biochemical
spatial
sequential
node properties
edge attributes
3D coordinates
kNN graph
edge encoding
node encoding
layer norm
EGC layers
$\phi(\mathbf{R}x+\mathbf{t}) = \mathbf{R}\phi(x)+\mathbf{t}$
Let $\mathbf{a}_{ij}$ contain geometric information!
$\mathbf{f}_j$
$\mathbf{x}_j$
$\mathbf{a}_{ij}$
$\mathbf{f}_i$
$\mathbf{x}_i$

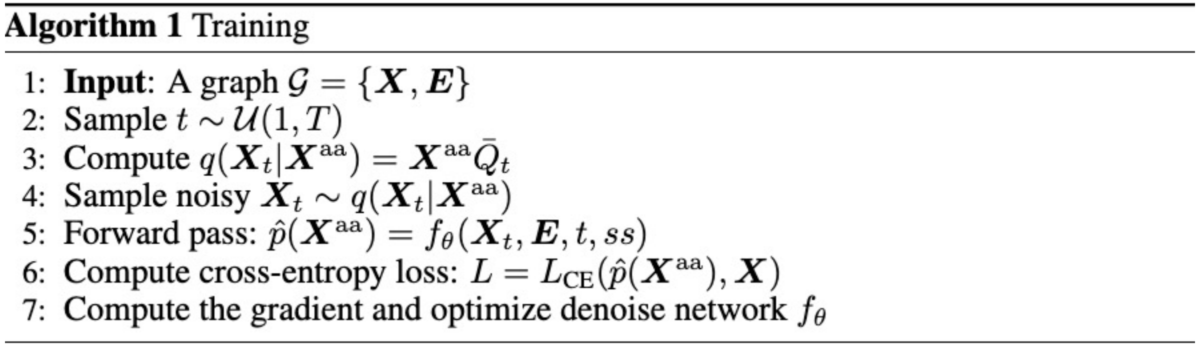


Algorithm 1 Training

```
1: Input: A graph $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{E}\}$
2: Sample $t \sim \mathcal{U}(1, T)$
3: Compute $q(\boldsymbol{X}_t|\boldsymbol{X}^{\text{aa}}) = \boldsymbol{X}^{\text{aa}}\bar{Q}_t$
4: Sample noisy $\boldsymbol{X}_t \sim q(\boldsymbol{X}_t|\boldsymbol{X}^{\text{aa}})$
5: Forward pass: $\hat{p}(\boldsymbol{X}^{\text{aa}}) = f_\theta(\boldsymbol{X}_t, \boldsymbol{E}, t, ss)$
6: Compute cross-entropy loss: $L = L_{\text{CE}}(\hat{p}(\boldsymbol{X}^{\text{aa}}), \boldsymbol{X})$
7: Compute the gradient and optimize denoise network $f_\theta$
```

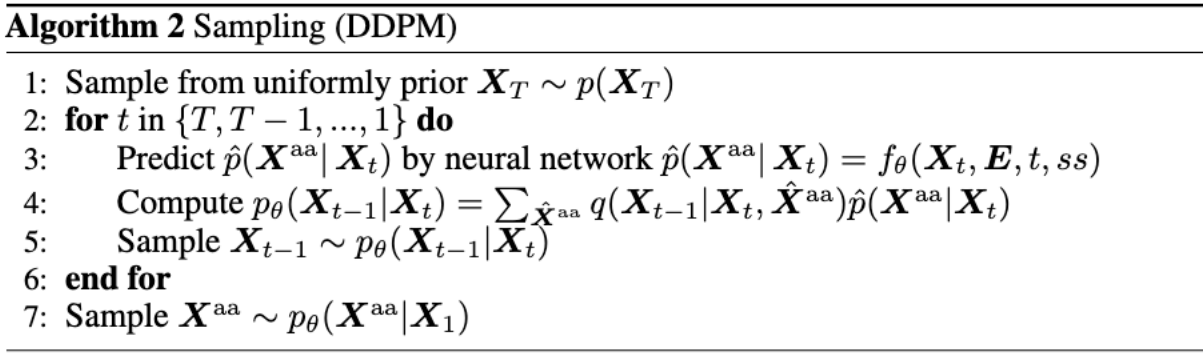
Graph Denoising Diffusion for Inverse Protein Folding





Algorithm 1 Training

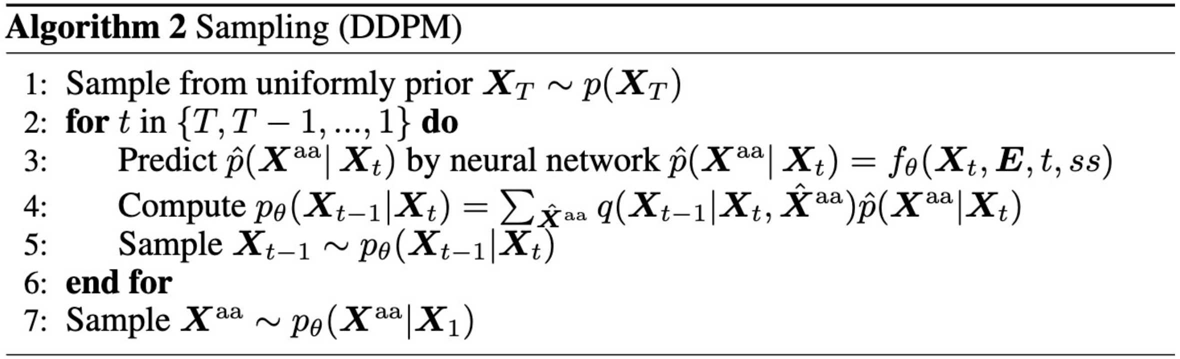
```
Input: A graph $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{E}\}$
Sample $t \sim \mathcal{U}(1, T)$
Compute $q(\boldsymbol{X}_t | \boldsymbol{X}^{\text{aa}}) = \boldsymbol{X}^{\text{aa}} \bar{Q}_t$
Sample noisy $\boldsymbol{X}_t \sim q(\boldsymbol{X}_t | \boldsymbol{X}^{\text{aa}})$
Forward pass: $\hat{p}(\boldsymbol{X}^{\text{aa}}) = f_\theta(\boldsymbol{X}_t, \boldsymbol{E}, t, ss)$
Compute cross-entropy loss: $L = L_{\text{CE}}(\hat{p}(\boldsymbol{X}^{\text{aa}}), \boldsymbol{X})$
Compute the gradient and optimize denoise network $f_\theta$
```



Algorithm 2 Sampling (DDPM)

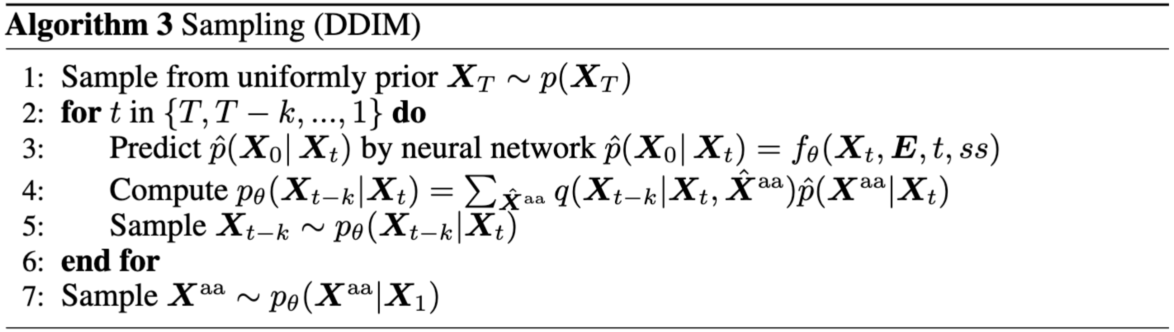
```
Sample from uniformly prior $\boldsymbol{X}_T \sim p(\boldsymbol{X}_T)$
for $t$ in $\{T, T-1, ..., 1\}$ do
    Predict $\hat{p}(\boldsymbol{X}^{\text{aa}} | \boldsymbol{X}_t)$ by neural network $\hat{p}(\boldsymbol{X}^{\text{aa}} | \boldsymbol{X}_t) = f_\theta(\boldsymbol{X}_t, \boldsymbol{E}, t, ss)$
    Compute $p_\theta(\boldsymbol{X}_{t-1} | \boldsymbol{X}_t) = \sum_{\hat{\boldsymbol{X}}^{\text{aa}}} q(\boldsymbol{X}_{t-1} | \boldsymbol{X}_t, \hat{\boldsymbol{X}}^{\text{aa}}) \hat{p}(\boldsymbol{X}^{\text{aa}} | \boldsymbol{X}_t)$
    Sample $\boldsymbol{X}_{t-1} \sim p_\theta(\boldsymbol{X}_{t-1} | \boldsymbol{X}_t)$
end for
Sample $\boldsymbol{X}^{\text{aa}} \sim p_\theta(\boldsymbol{X}^{\text{aa}} | \boldsymbol{X}_1)$
```

Generative Process



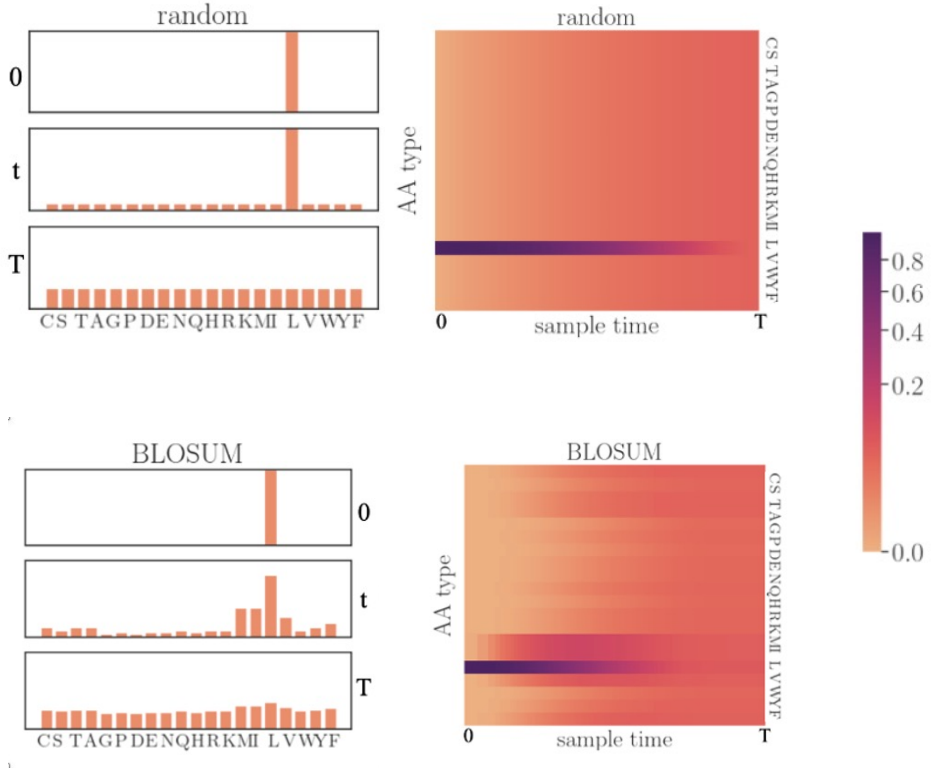
Algorithm 2 Sampling (DDPM)

```
Sample from uniformly prior $\boldsymbol{X}_T \sim p(\boldsymbol{X}_T)$
for $t$ in $\{T, T-1, ..., 1\}$ do
    Predict $\hat{p}(\boldsymbol{X}^{\text{aa}}|\, \boldsymbol{X}_t)$ by neural network $\hat{p}(\boldsymbol{X}^{\text{aa}}|\, \boldsymbol{X}_t) = f_\theta(\boldsymbol{X}_t, \boldsymbol{E}, t, ss)$
    Compute $p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t) = \sum_{\hat{\boldsymbol{X}}^{\text{aa}}} q(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t, \hat{\boldsymbol{X}}^{\text{aa}})\hat{p}(\boldsymbol{X}^{\text{aa}}|\boldsymbol{X}_t)$
    Sample $\boldsymbol{X}_{t-1} \sim p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t)$
end for
Sample $\boldsymbol{X}^{\text{aa}} \sim p_\theta(\boldsymbol{X}^{\text{aa}}|\boldsymbol{X}_1)$
```



Algorithm 3 Sampling (DDIM)

```
Sample from uniformly prior $\boldsymbol{X}_T \sim p(\boldsymbol{X}_T)$
for $t$ in $\{T, T-k, ..., 1\}$ do
    Predict $\hat{p}(\boldsymbol{X}_0|\, \boldsymbol{X}_t)$ by neural network $\hat{p}(\boldsymbol{X}_0|\, \boldsymbol{X}_t) = f_\theta(\boldsymbol{X}_t, \boldsymbol{E}, t, ss)$
    Compute $p_\theta(\boldsymbol{X}_{t-k}|\boldsymbol{X}_t) = \sum_{\hat{\boldsymbol{X}}^{\text{aa}}} q(\boldsymbol{X}_{t-k}|\boldsymbol{X}_t, \hat{\boldsymbol{X}}^{\text{aa}})\hat{p}(\boldsymbol{X}^{\text{aa}}|\boldsymbol{X}_t)$
    Sample $\boldsymbol{X}_{t-k} \sim p_\theta(\boldsymbol{X}_{t-k}|\boldsymbol{X}_t)$
end for
Sample $\boldsymbol{X}^{\text{aa}} \sim p_\theta(\boldsymbol{X}^{\text{aa}}|\boldsymbol{X}_1)$
```



Outline

- Introduction
- Protein Graph Representation & Inverse Folding
- Graph Denoising Diffusion for Inverse Protein Folding (GraDe-IF)
- Experiment & Evaluation

Experiment & Evaluation

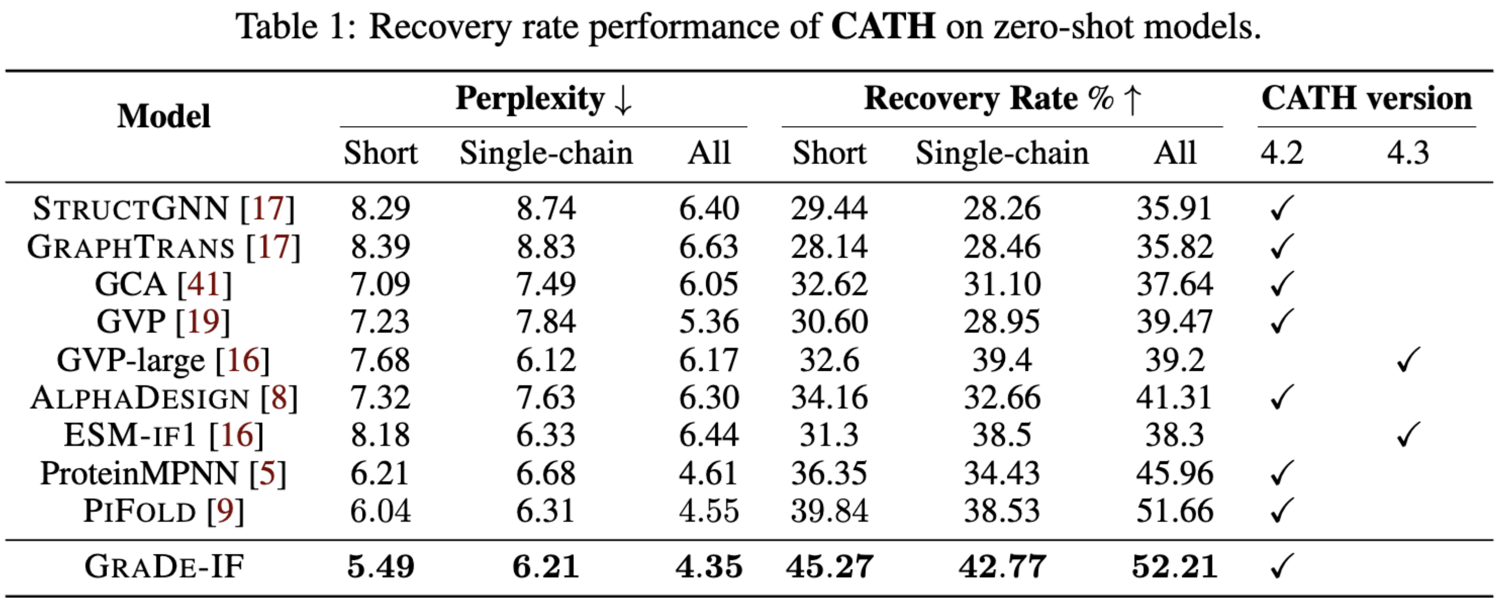


Table 1: Recovery rate performance of **CATH** on zero-shot models.

Model	**Perplexity ↓**			**Recovery Rate % ↑**			**CATH version**	
	Short	Single-chain	All	Short	Single-chain	All	4.2	4.3
STRUCTGNN [17]	8.29	8.74	6.40	29.44	28.26	35.91	✓	
GRAPHTRANS [17]	8.39	8.83	6.63	28.14	28.46	35.82	✓	
GCA [41]	7.09	7.49	6.05	32.62	31.10	37.64	✓	
GVP [19]	7.23	7.84	5.36	30.60	28.95	39.47	✓	
GVP-large [16]	7.68	6.12	6.17	32.6	39.4	39.2		✓
ALPHADESIGN [8]	7.32	7.63	6.30	34.16	32.66	41.31	✓	
ESM-IF1 [16]	8.18	6.33	6.44	31.3	38.5	38.3		✓
ProteinMPNN [5]	6.21	6.68	4.61	36.35	34.43	45.96	✓	
PIFOLD [9]	6.04	6.31	4.55	39.84	38.53	51.66	✓	
GRADE-IF	**5.49**	**6.21**	**4.35**	**45.27**	**42.77**	**52.21**	✓	

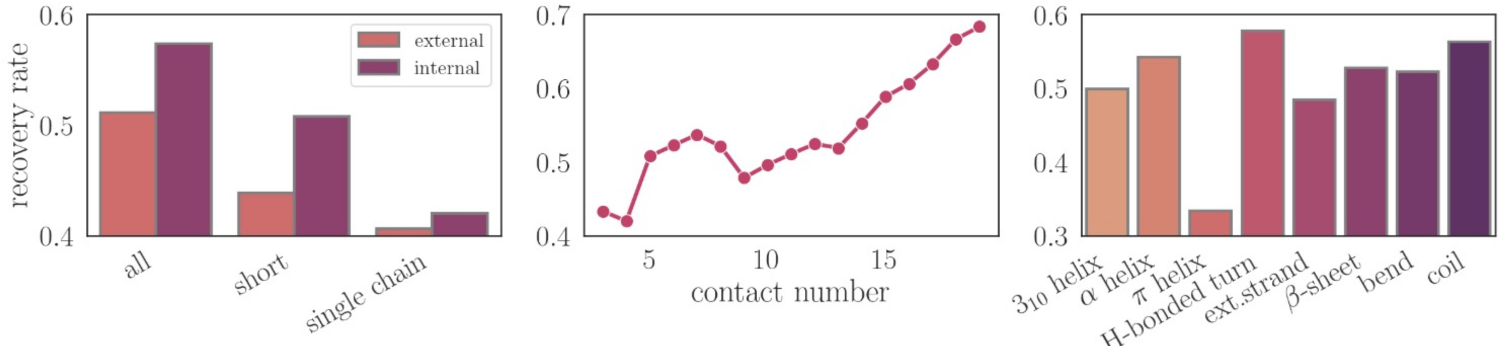
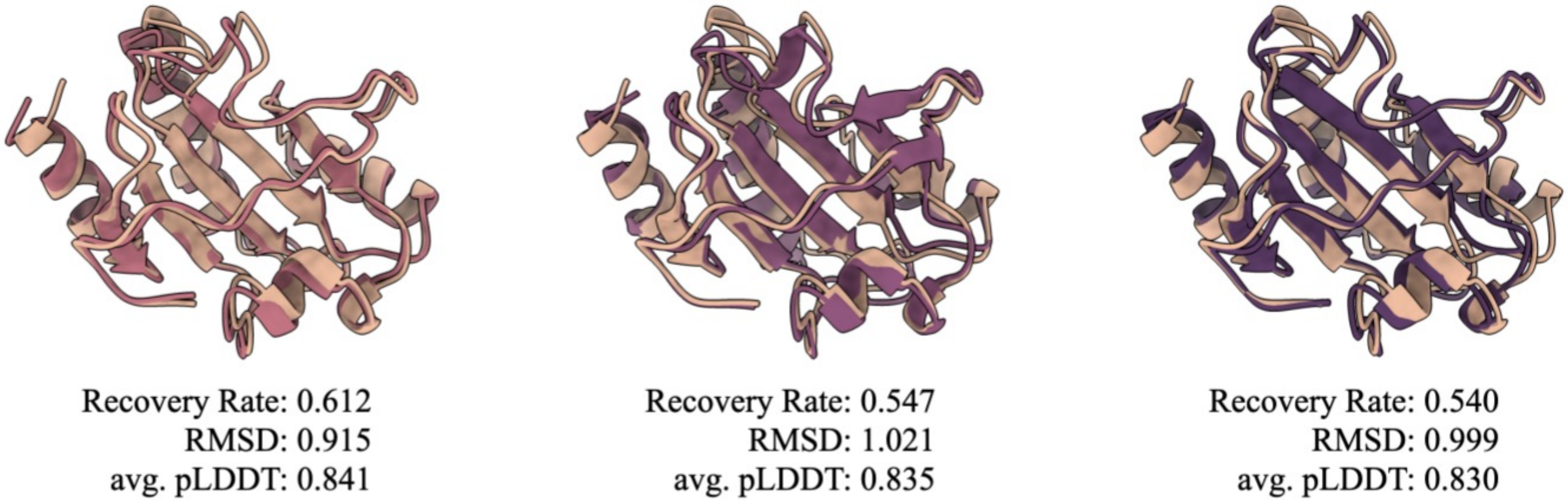


Figure 3: Recovery rate on core and surface residues and different secondary structure

Experiment & Evaluation



Recovery Rate: 0.612
RMSD: 0.915
avg. pLDDT: 0.841

Recovery Rate: 0.547
RMSD: 1.021
avg. pLDDT: 0.835

Recovery Rate: 0.540
RMSD: 0.999
avg. pLDDT: 0.830

Figure 6: Folding prediction of generated protein sequence by GraDe-IF with respect to the native protein (PDB ID: 3FKF, colored in nude).

Experiment & Evaluation

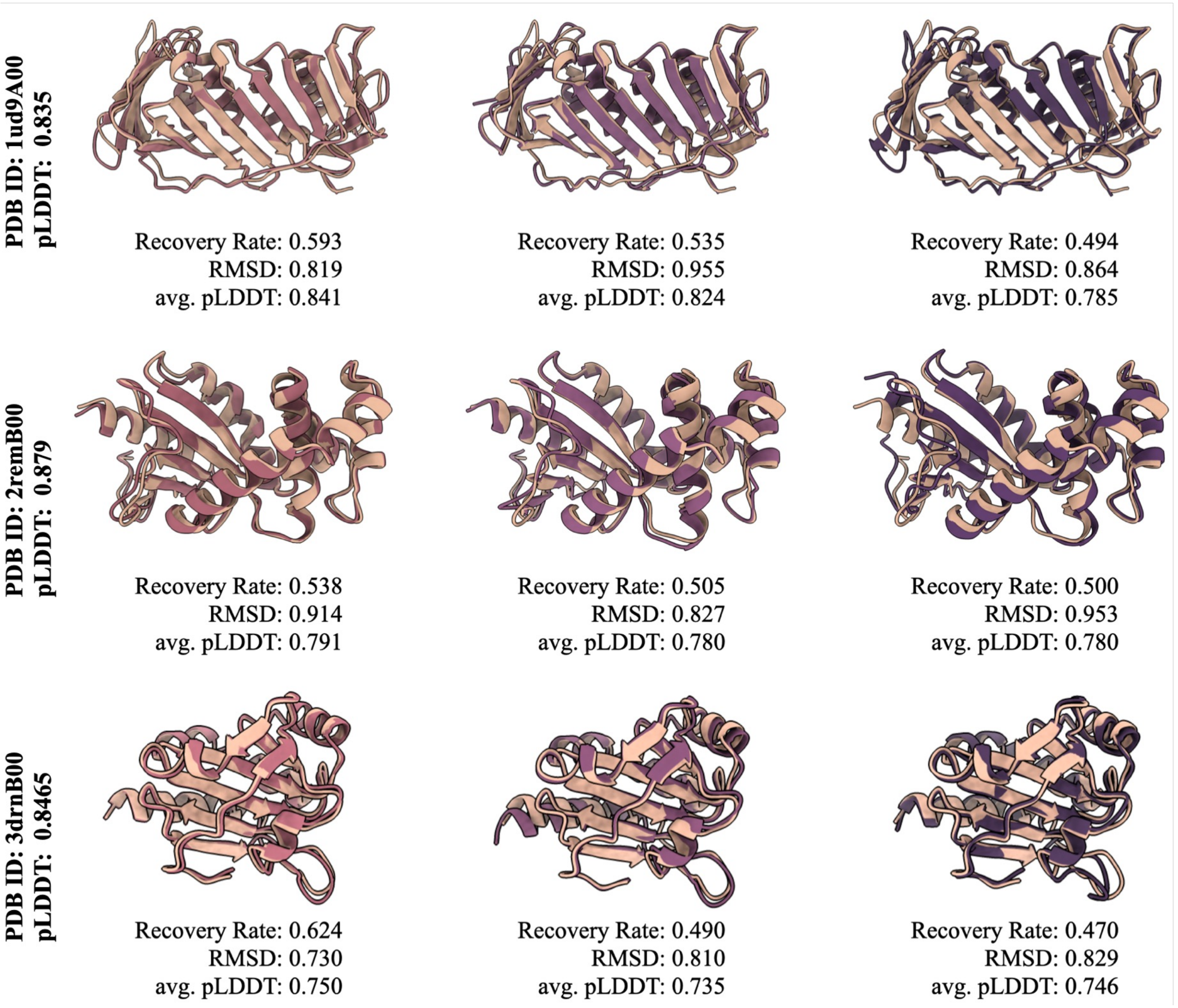


Table 2. Average RMSD and pLDDT across 42 protein structures (folded by AlphaFold2) with model-generated sequences.

Model	avg. pLDDT	avg RMSD
PiFold	0.847 ± 0.160	1.530 ± 0.928
ProteinMPNN	0.881 ± 0.084	1.499 ± 0.816
GraDe-IF	0.892 ± 0.106	1.284 ± 0.747

Table 4. TM score comparison. Each protein backbone generates 30 sequences.

Protein	GraDe-IF	ProteinMPNN	PiFold
1ud9.A	0.97 ± 0.01	0.95 ± 0.04	0.94 ± 0.04
2rem.B	0.96 ± 0.01	0.94 ± 0.01	0.93 ± 0.01
3drn.B	0.96 ± 0.01	0.95 ± 0.01	0.95 ± 0.01
3fkf.A	0.76 ± 0.03	0.67 ± 0.01	0.64 ± 0.02

Table 5. TM score comparison on 42 protein backbones, with each backbone generating 1 sample. A sequence is considered foldable if its TM score > 0.5.

Method	Success	TM score
PiFold	37	0.83 ± 0.21
ProteinMPNN	39	0.82 ± 0.16
GraDe-IF	39	0.87 ± 0.17

Reference

Yi, Kai, et al. "Graph Denoising Diffusion for Inverse Protein Folding." *arXiv preprint arXiv:2306.16819* (2023).

Hsu, Chloe, et al. "Learning inverse folding from millions of predicted structures." . PMLR, 2022.

Austin, Jacob, et al. "Structured denoising diffusion models in discrete state-spaces." *Advances in Neural Information Processing Systems* 34 (2021): 17981-17993.

Dauparas, Justas, et al. "Robust deep learning–based protein sequence design using ProteinMPNN." *Science* 378.6615 (2022): 49-56.

Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." *Nature* 596.7873 (2021): 583-589.