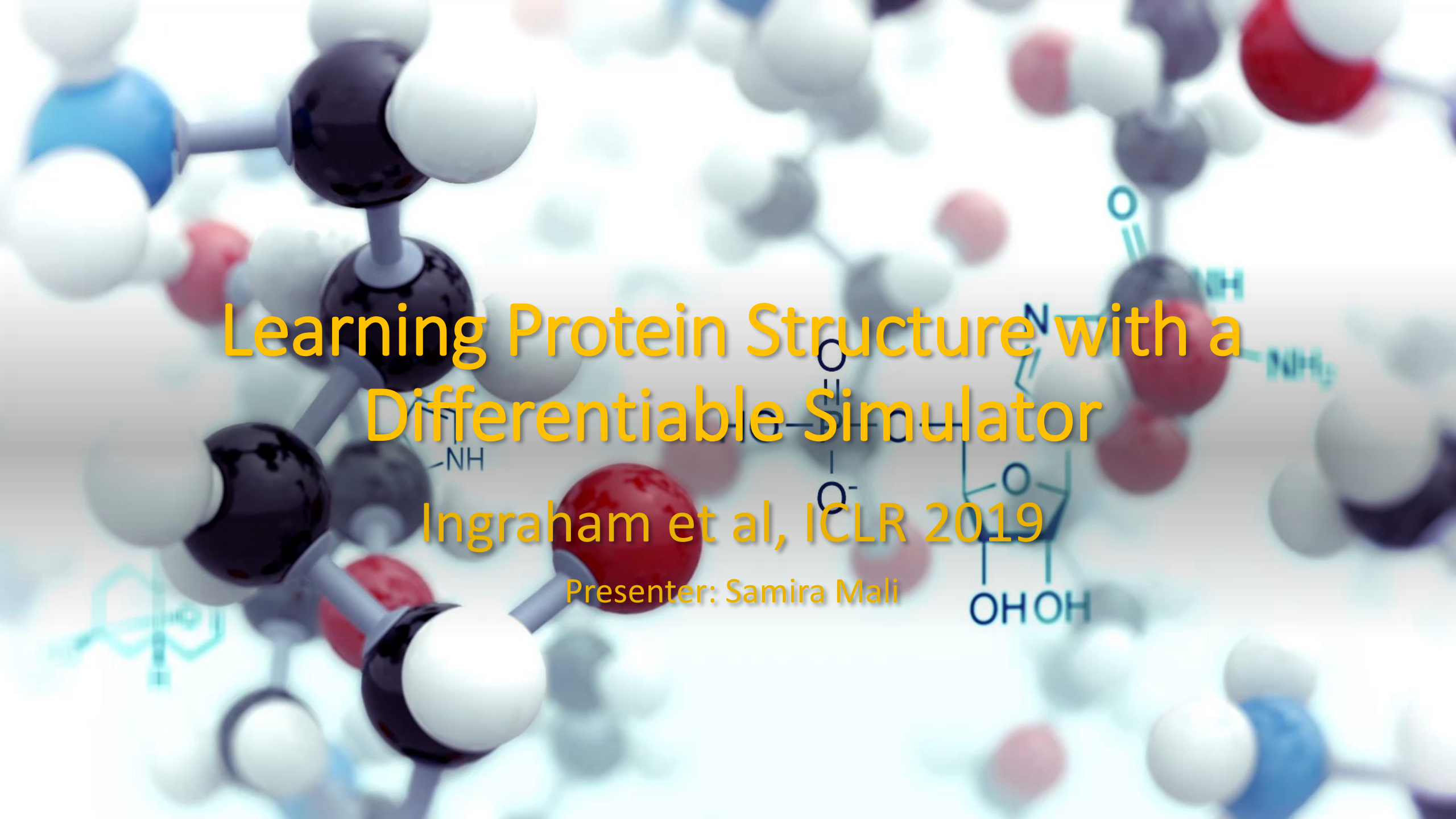


Learning Protein Structure with a Differentiable Simulator

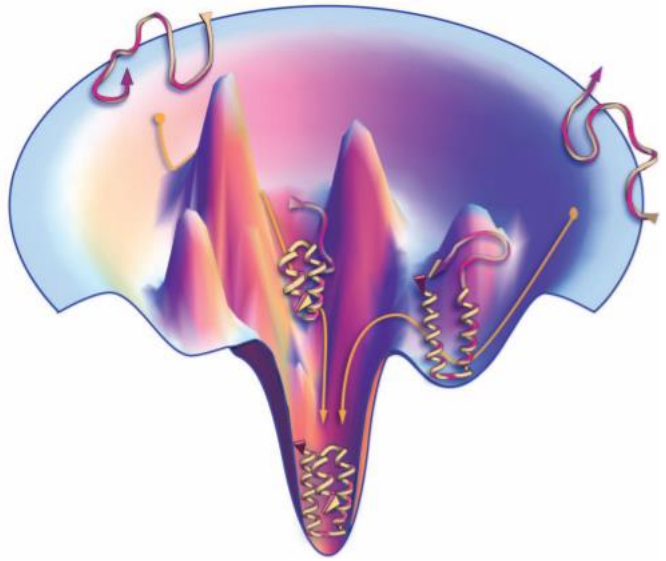
Ingraham et al, ICLR 2019

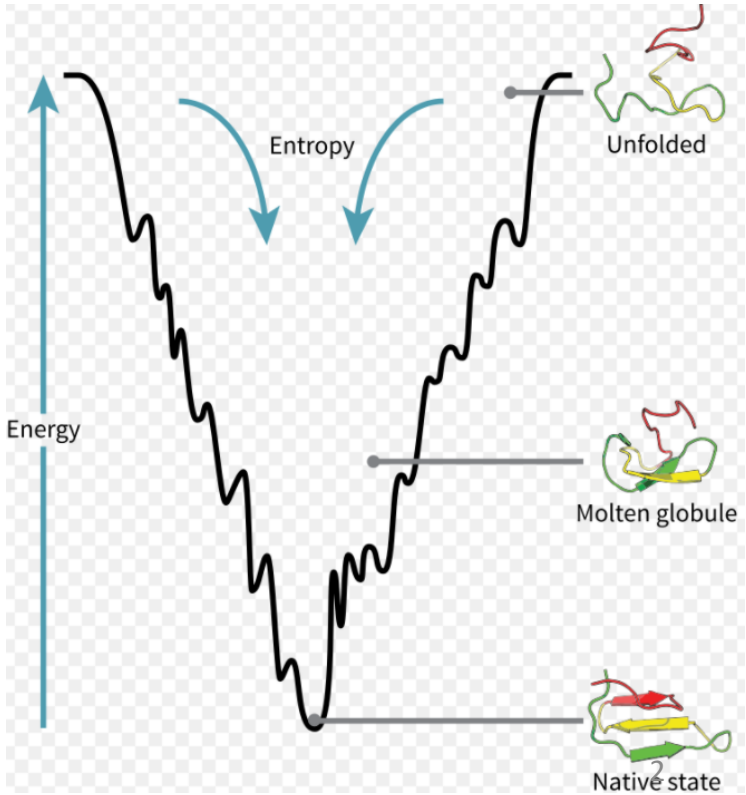
Presenter: Samira Mali

Introduction

- Energy landscape theory of protein folding

Folds that natural protein sequences adopt are those that minimize free energy.





[1] Ken Dill, Robert L Jernigan, and Ivet Bahar. Protein Actions: Principles and Modeling. Garland Science, 2017.

[2] Dill, Ken A., and Justin L. MacCallum. "The protein-folding problem, 50 years on." science 338.6110 (2012): 1042-1046.

Motivation

- End-to-end **differentiable** model:

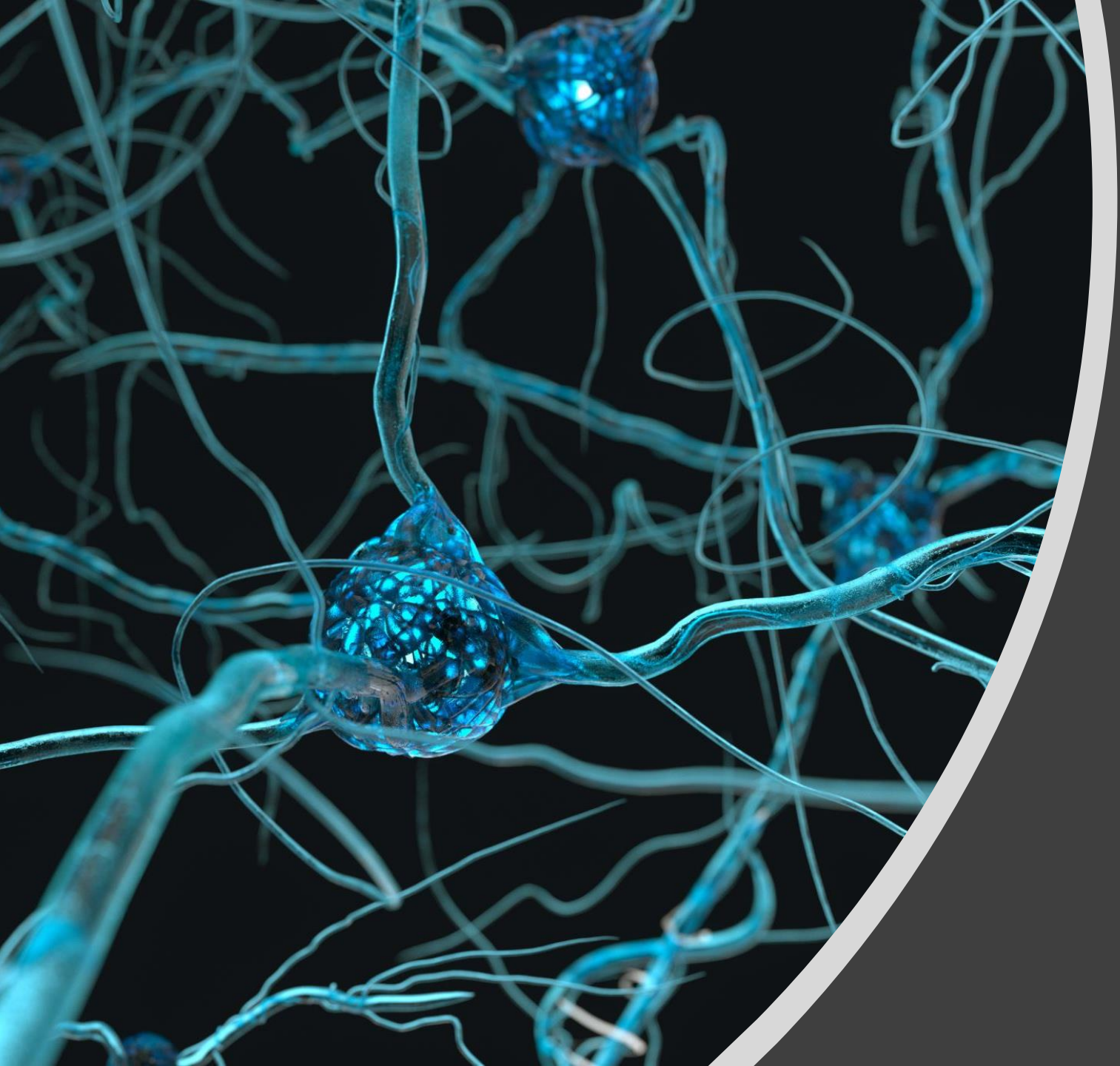
  Predict protein structure given amino acid sequence

Monte carlo simulation	MD simulation Langevin Dynamics

- Boltzmann Energy function $$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{Z} \exp\left(-U_{\boldsymbol{\theta}}[\boldsymbol{x}]\right)$$

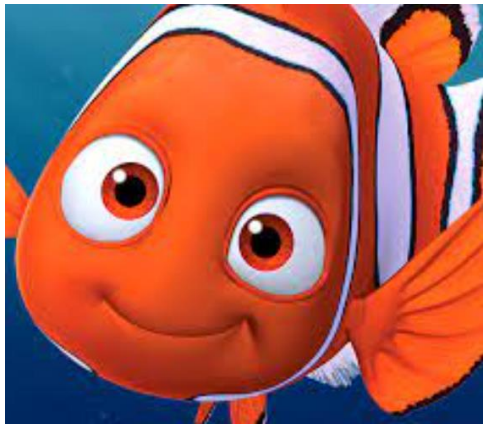
  Sampling from Boltzmann distribution is difficult ---- Generative models

- Bayesian Inference is isomorphic with statistical mechanics



Neural Energy function

NEMO (**N**eural **E**nergy **M**odeling and **O**ptimization)

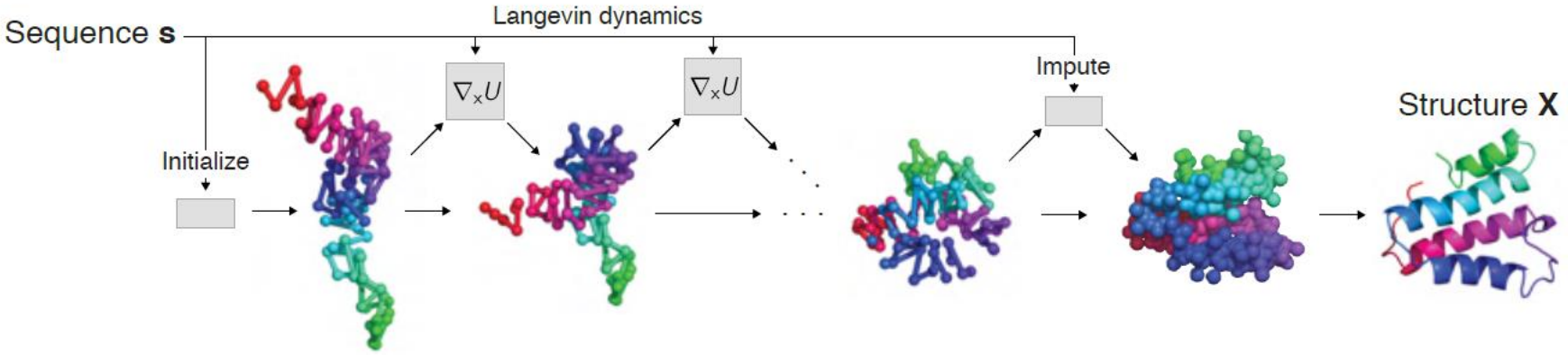


- Disadvantages of previous energy-based models
- This model is based on :

Neural Energy *Simulator* Model

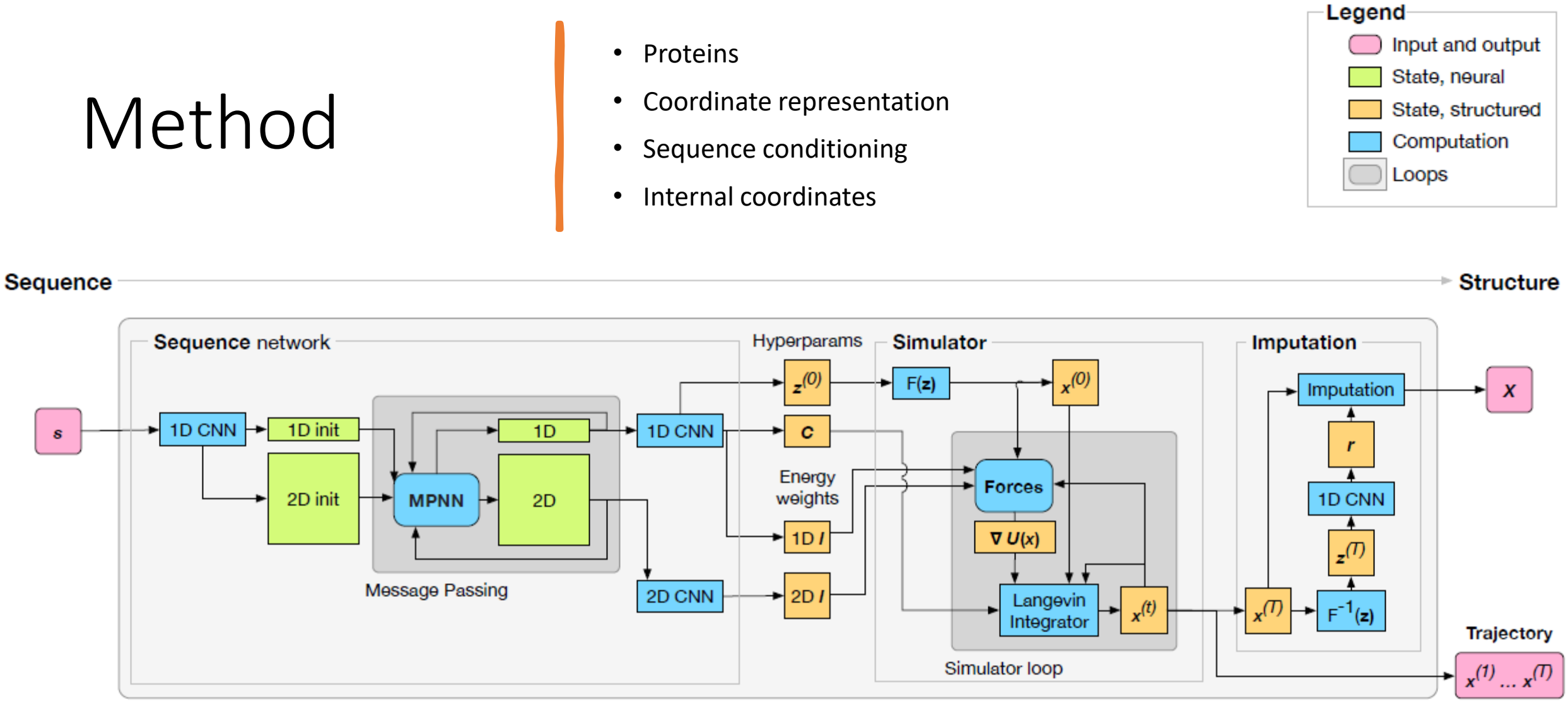
Efficient Sampling Algorithm

- Imputation Network
- When has the simulator converged? Backpropagation through folding



Method

- Proteins
- Coordinate representation
- Sequence conditioning
- Internal coordinates



- How to construct evolutionary profile: PSSM

Method- cont'd

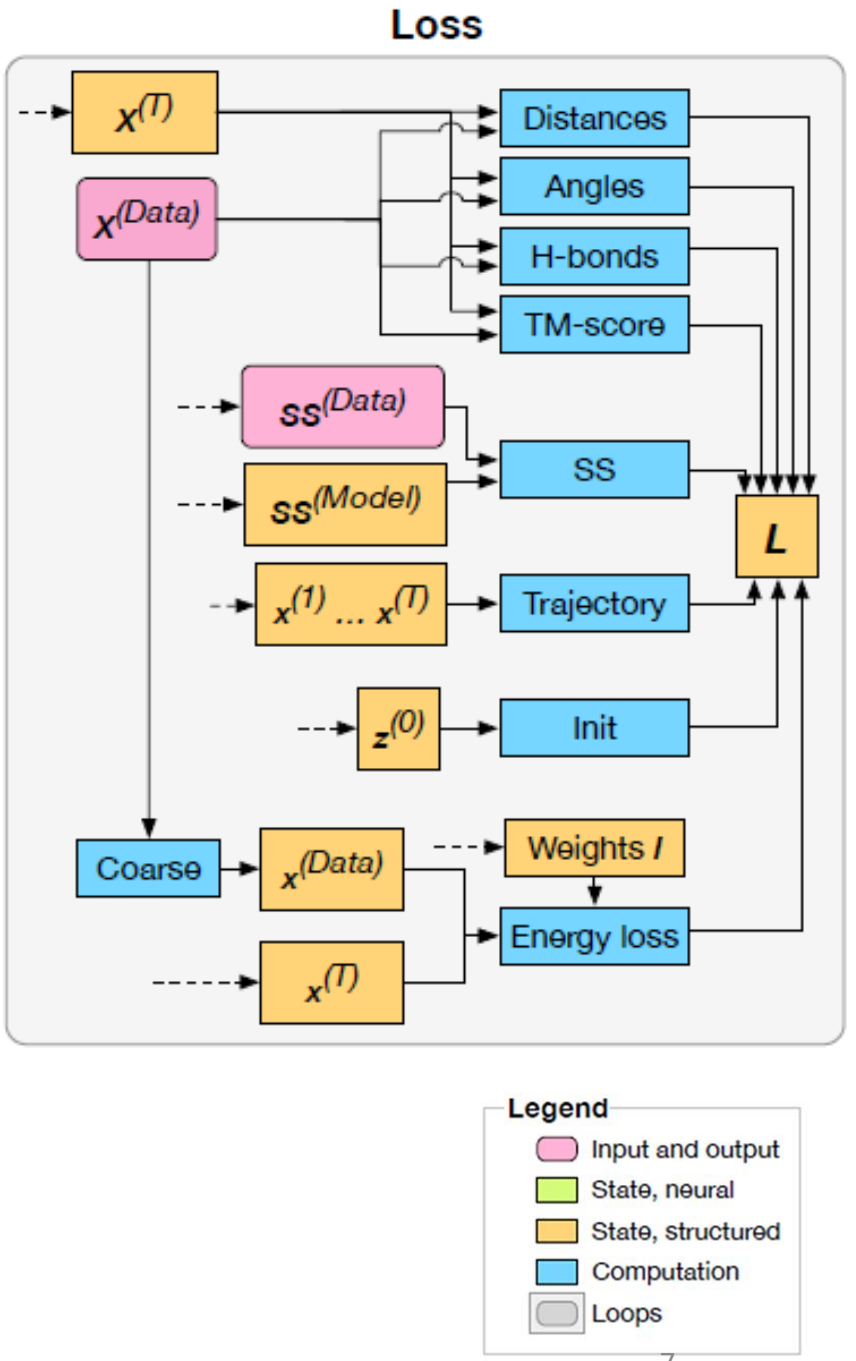
- Training
  Transform Integrator

Algorithm 1: Direct integrator

Input : State $\boldsymbol{z}^{(0)}$, energy $U(\boldsymbol{x})$, step ϵ, time T, scale $\mathbf{C}$
Output : Trajectory $\boldsymbol{x}^{(0)}, \ldots, \boldsymbol{x}^{(T)}$
Initialize $\boldsymbol{x}^{(0)} \leftarrow \mathcal{F}(\boldsymbol{z}^{(0)})$;
while $t < T$ **do**
  Compute forces $\boldsymbol{f}_{\boldsymbol{z}} = -\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}^{T} \nabla_{\boldsymbol{x}} U$;
  Sample $\Delta \boldsymbol{z} \sim \mathcal{N}\left(\frac{1}{2}\epsilon \mathbf{C} \boldsymbol{f}_{\boldsymbol{z}}, \epsilon \mathbf{C}\right)$;
  $\boldsymbol{z}^{(t+\epsilon)} \leftarrow \boldsymbol{z}^{(t)} + \Delta \boldsymbol{z}$;
  $\boldsymbol{x}^{(t+\epsilon)} \leftarrow \mathcal{F}(\boldsymbol{z}^{(t+\epsilon)})$;
  $t \leftarrow t + \epsilon$;
end

Algorithm 2: Transform integrator

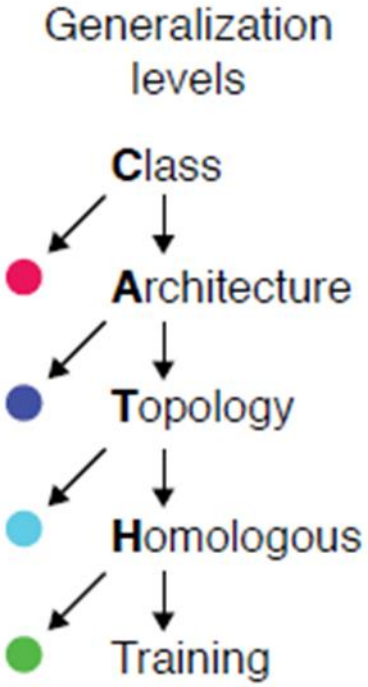
Input : State $\boldsymbol{z}^{(0)}$, energy $U(\boldsymbol{x})$, step ϵ, time T, scale $\mathbf{C}$
Output : Trajectory $\boldsymbol{x}^{(0)}, \ldots, \boldsymbol{x}^{(T)}$
Initialize $\boldsymbol{x}^{(0)} \leftarrow \mathcal{F}(\boldsymbol{z}^{(0)})$;
while $t < T$ **do**
  Compute forces $\boldsymbol{f}_{\boldsymbol{z}} = -\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}^{T} \nabla_{\boldsymbol{x}} U$;
  Sample $\Delta \boldsymbol{z} \sim \mathcal{N}\left(\frac{1}{2}\epsilon \mathbf{C} \boldsymbol{f}_{\boldsymbol{z}}, \epsilon \mathbf{C}\right)$;
  $\tilde{\mathbf{x}} \leftarrow \boldsymbol{x}^{(t)} + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}^{(t)} \Delta \boldsymbol{z}^{(t)}$;
  $\boldsymbol{x}^{(t+\epsilon)} \leftarrow \boldsymbol{x}^{(t)} + \frac{1}{2}\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}^{(t)} + \frac{\partial \tilde{\mathbf{x}}}{\partial \boldsymbol{z}}\right) \Delta \boldsymbol{z}^{(t)}$;
  $t \leftarrow t + \epsilon$;
end



- Loss (Monte carlo estimator/Distance/Angle/Trajectory/Hydrogen bond)
- Stabilizing backpropagation through time

Data

- For a training and validation set, the authors used all protein domains of length L <= 200 from **C**lasses in CATH release 4.1 (2015)
- And then hierarchically purged a randomly selected set of A, T, and H categories.
- CATH hierarchically organizes proteins from the Protein Data Bank (Berman et al., 2000) into domains (individual folds) that are classified at the levels of **C**lass, **A**rchitecture, **T**opology, and **H**omologous superfamily (from general to specific).



Results

Table 1: Test set performance across different levels of generalization

Model	# params	Total	C	A	T	H
NEMO (ours, profile)	21.3m	**0.366**	**0.274**	**0.361**	**0.331**	0.431
NEMO (ours, sequence-only)	19.1m	0.248	0.198	0.245	0.254	0.263
RNN baseline model (profile)						
2x100	5.9m	0.293	0.213	0.230	0.247	0.388
2x300 (avg. of 3)	8.8m	0.335	0.229	0.282	0.278	0.446
2x500	13.7m	0.347	0.222	0.272	0.286	**0.477**
2x700	21.4m	0.309	0.223	0.259	0.261	0.403
Number of structures		10381	1537	1705	3198	3941

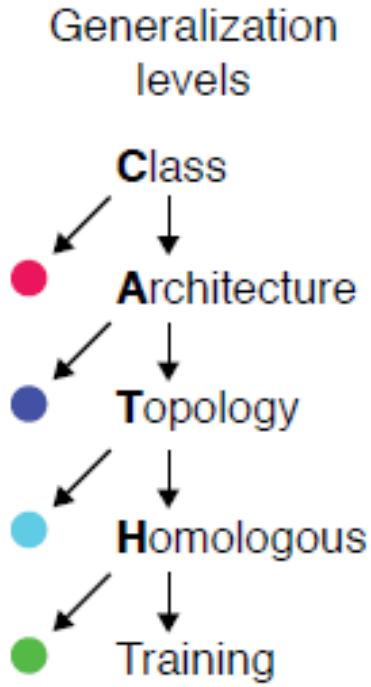
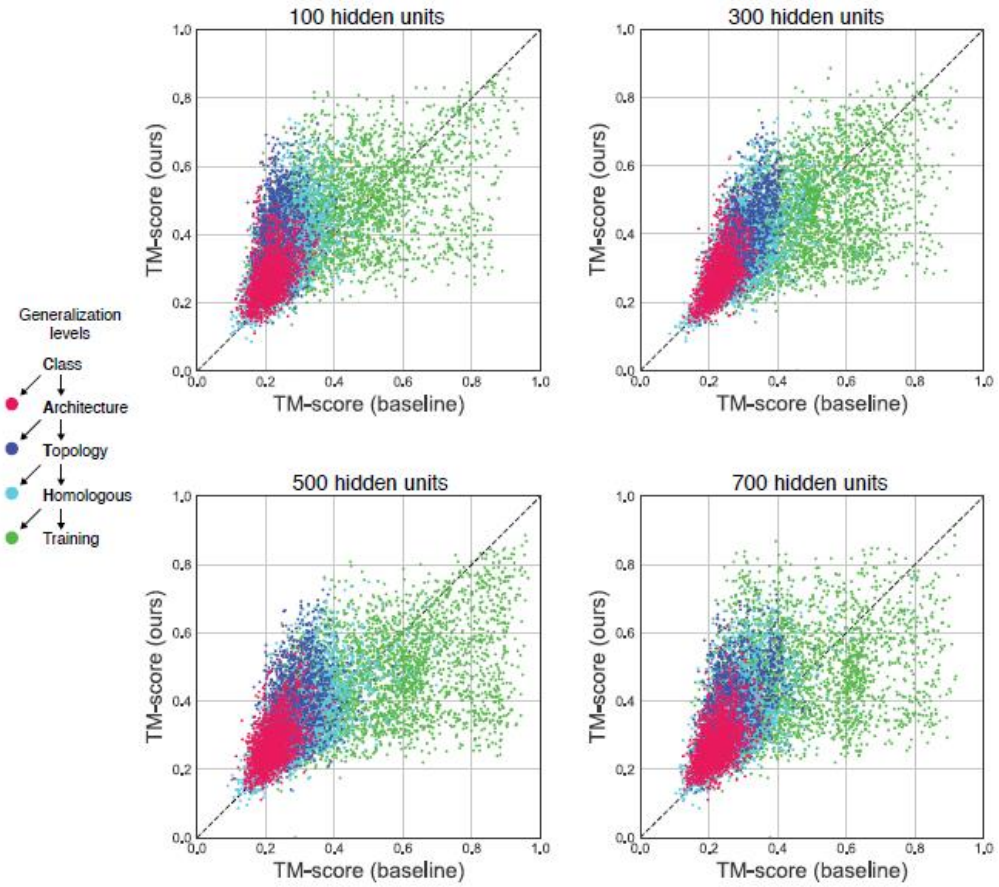
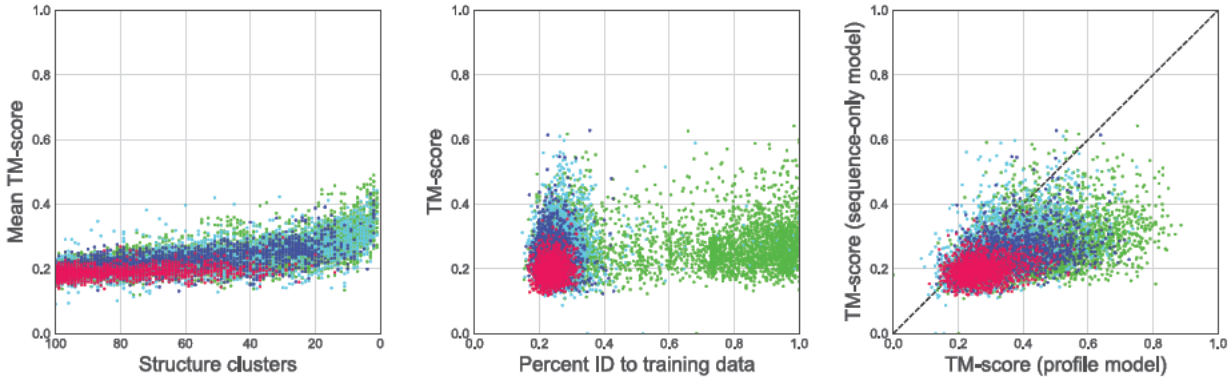


Table 4: **Qualitative timings**. †Results on CATH dataset and 2 M40 GPUs.

Method	Generation time	Training time
RNN baseline†	milliseconds	∼ 1 week
NEMO†	seconds	∼ 2 months
Coevolution-based methods	minutes to hours	Coupled to generation
Physical simulations	days to weeks	N/A

Results- cont'd



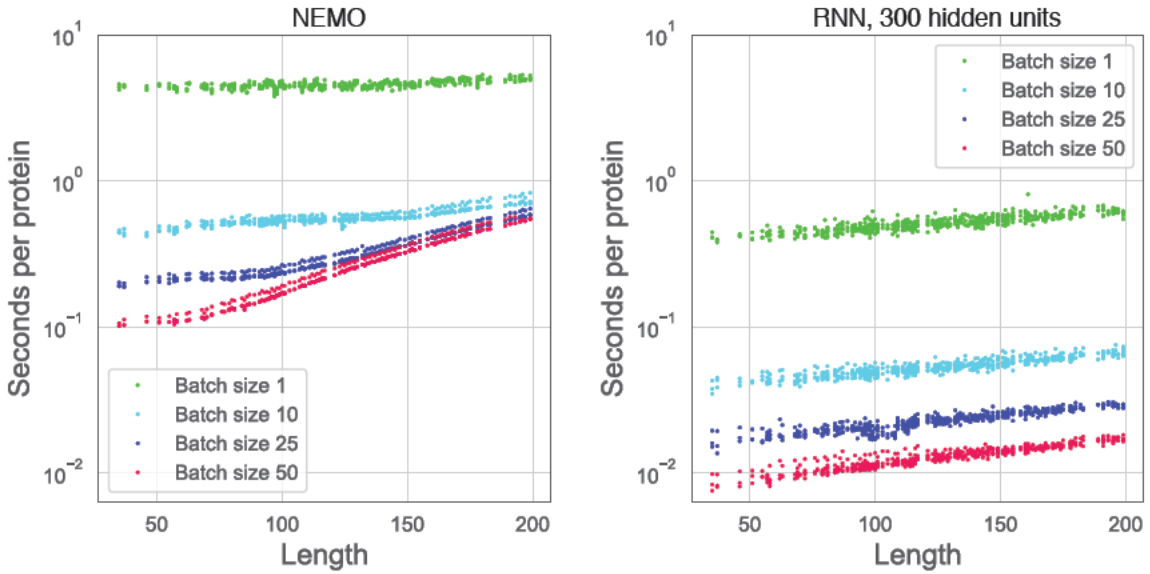


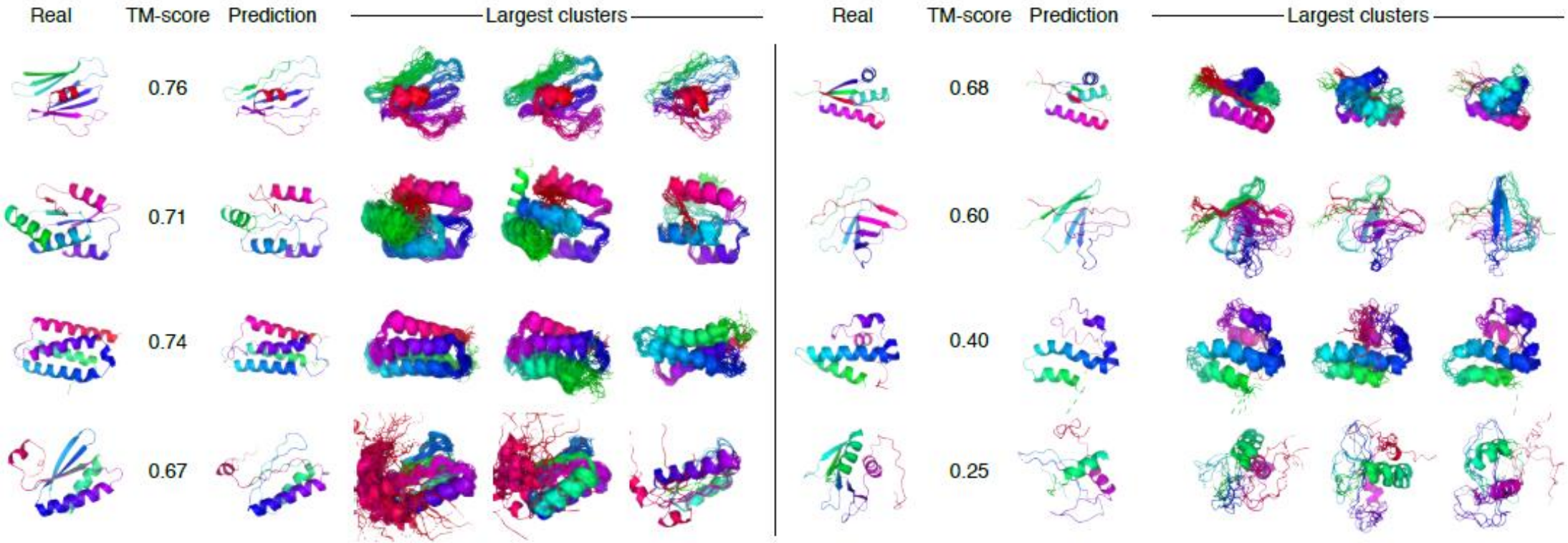
NEMO: Predictive performance of structures generated by the sequence-only/profile model

RNN baseline performance for different hyperparameters

Disadvantages/Future work

- The computational cost of training and sampling is high
- Instability of backpropagating through long simulations




Real
TM-score
Prediction
Largest clusters
0.76
0.71
0.74
0.67
Real
TM-score
Prediction
Largest clusters
0.68
0.60
0.40
0.25