Learning Protein Structure with a Differentiable Simulators (John Ingraham et al., ICLR 2019)

COS598L, Spr26

Yufan Xia

Feb 02, 2026

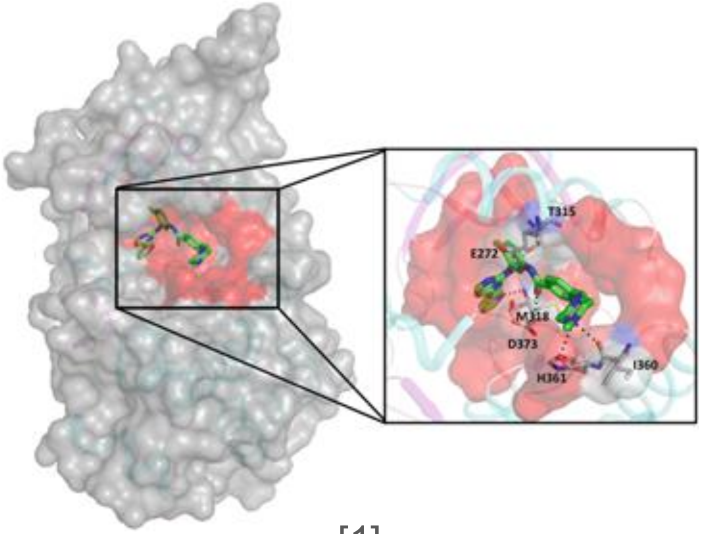
Outline

- Motivation
  - Why care about protein structure?
  - Why computational tool to determine the structure?
- Background
  - Related Method: Physics-based vs. statistical
  - SOTA = AlphaFold1
- Method
  - Problem Statement
  - Technical Primer
  - Dataset + Representation
  - Model Structure
  - Training Tricks
- Results
  - 3D Folding Procedure
- Discussion
  - Challenges
  - Strengths & Weaknesses
  - Comparison to AlphaFold1
- Conclusion & Future Directions

Motivation - Protein structures determine their functions

Proteins functions: Binding, signaling, and regulation etc. *depend on 3D shape*

- ➢ Drug discovery & design
- ➢ Disease mechanism understanding
- ➢ Protein engineering

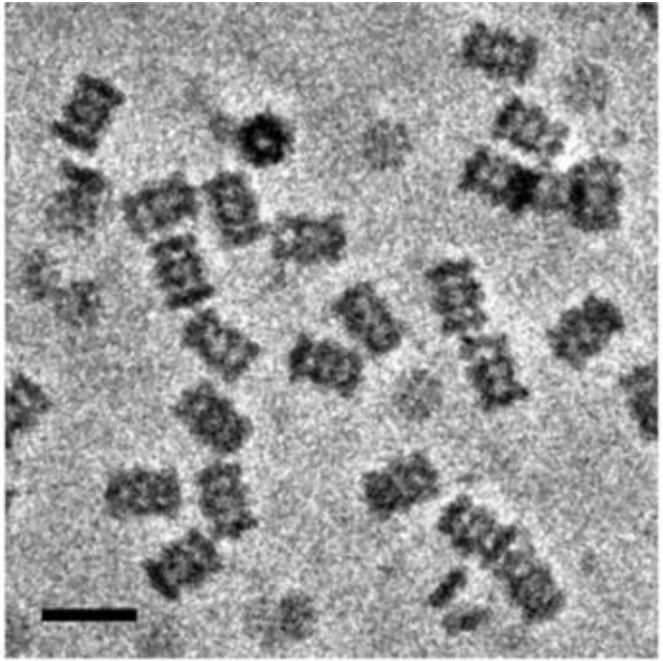


[1]

Motivation - Experimental protein structure determination

We can experimentally “see” the protein structure by:

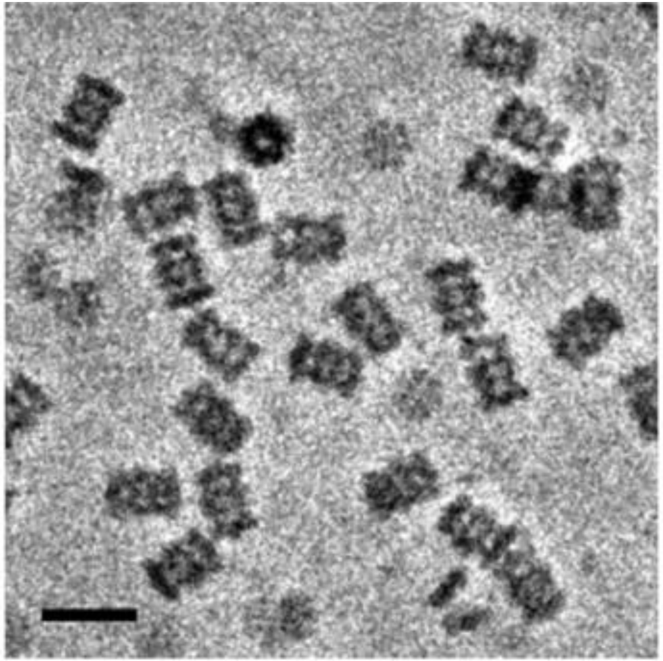
1. NMR spectroscopy
2. X-ray crystallography
3. *Cryo-electron microscopy (cryo-EM)*



cryo-EM images of recombinant Thermoplasma acidophilum 20S proteasomes[2]

Motivation - Experimental protein structure determination

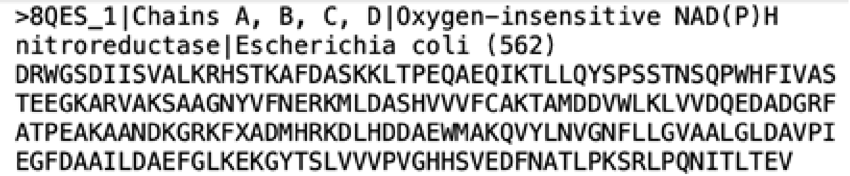
https://www.nature.com/nature-index/news/must-have-multimillion-dollar-microscopy-machine-cryo-em



cryo-EM images of recombinant Thermoplasma acidophilum 20S proteasomes[2]

Motivation - Why computational prediction?

Amino acid sequence



```
>8QES_1|Chains A, B, C, D|Oxygen-insensitive NAD(P)H
nitroreductase|Escherichia coli (562)
DRWGSDIISVALKRHSTKAFDASKKLTPEQAEQIKTLLQYSPSSTNSQPWHFIVAS
TEEGKARVAKSAAGNYVFNERKMLDASHVVVFCAKTAMDDVWLKLVVDQEDADGRF
ATPEAKAANDKGRKFXADMHRKDLHDDAEWMAKQVYLNVGNFLLGVAALGLDAVPI
EGFDAAILDAEFGLKEKGYTSLVVVPVGHHSVEDFNATLPKSRLPQNITLTEV
```

PDB Bank - 8QES

https://www.rcsb.org/structure/8QES

3D structure

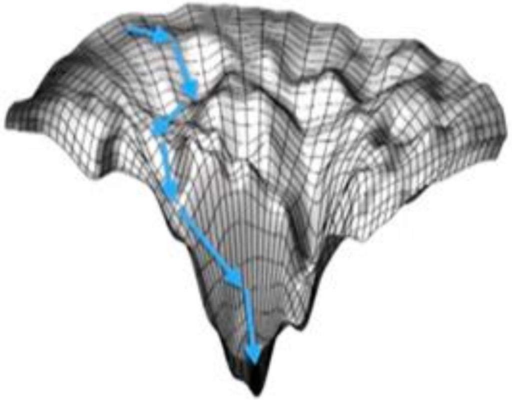
We can experimentally “see” the protein structure,

But, requires big machines + experts + money

So computational tool: *cheap + fast*

Background - Related Method - Physics

- ❖ Assumption:
  - ➢ A protein folds into the structure that minimizes its free energy.
- ❖ Method:
  - ➢ Define an energy function, considering all physical, chemical properties
  - ➢ Search for the lowest-energy conformation
- ❖ Slow but works without large databases



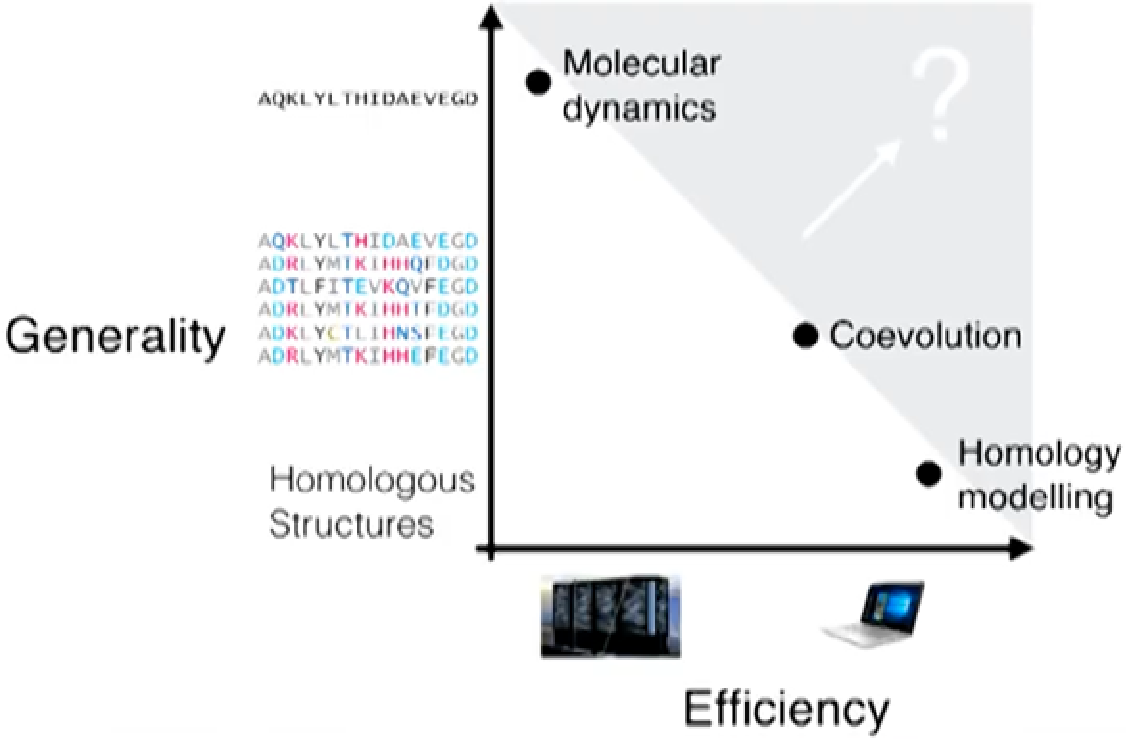
Energy landscape

https://www.youtube.com/watch?v=R20_s8XPw8U&t=145s

Background - Related Method - Statistics

- Assumption:
  - If two residues interact in 3D, they tend to mutate together across evolution.
- Method - data-driven:
  - Start with thousands of homologous protein sequences
  - Align residue by residue, find covariation / co-evolution:
    - If position i mutates, position j also mutates
    - And it happens across many sequences
    - Then positions i and j are likely close in 3D space
- Fast but limited generalization

Background - ML-based model: efficient + generalizable



https://www.youtube.com/watch?v=R20_s8XPw8U&t=145s

Background - Computational Method - SOTA

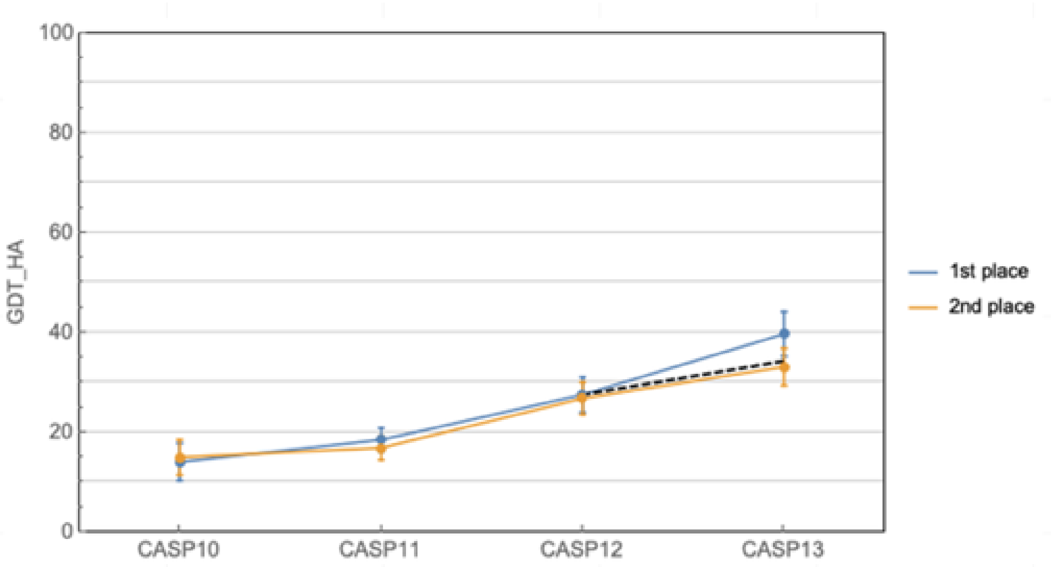
Timeline:

Alphafold1 - **NEMO** - Alphafold2

Alphafold1 won CASP13

Strength:

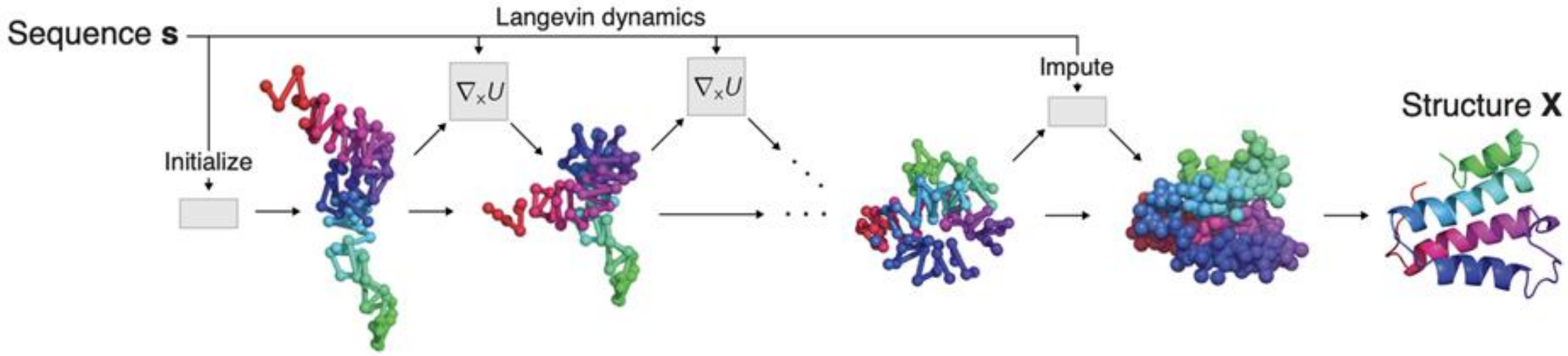
1. High generalizability
2. Short inference time



https://moalquraishi.wordpress.com/2018/12/09/alphafold-casp13-what-just-happened/

Method

Method - Problem Statement



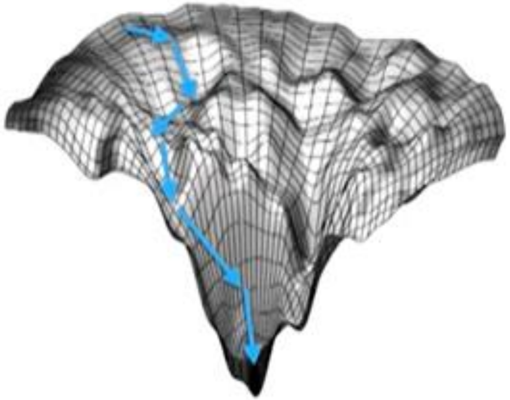
Given an amino acid sequence, generate realistic 3D protein structures by simulating a learned folding process

- ➢ Input: protein sequence
- ➢ Output: one or more plausible 3D structures
- ➢ Constraint: sampling must be fast (fixed number of steps)

Method - Technical Primer

- Boltzmann distribution: A probability distribution over structures defined implicitly by the learned energy function

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

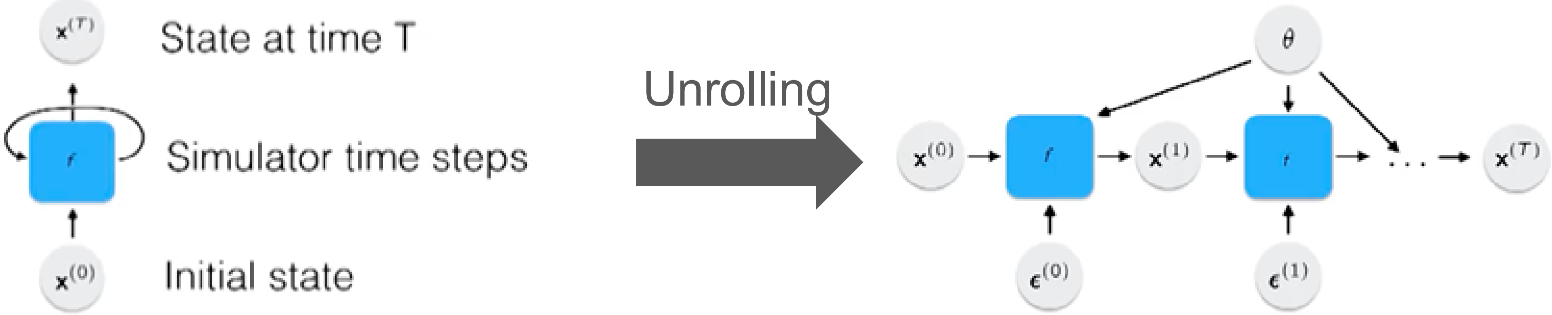


Energy landscape

https://www.youtube.com/watch?v=R20_s8XPw8U&t=145s

Method - Technical Primer

- Monte Carlo algorithm: Samples structures via random proposals with accept or reject steps based on energy
- **(Unrolled) langevin dynamics: Samples structures via gradient-based updates plus Gaussian noise (no accept or reject)**



https://www.youtube.com/watch?v=R20_s8XPw8U&t=145s

Method - Technical Primer

- Template modelling (TM) score (0,1]: A measure of similarity between two protein structures
  - 1 = perfect match
  - > 0.5 roughly same fold

$$\sum_i \frac{1}{1+\left(\frac{D_i}{D_0}\right)^2}$$

$$D_i = ||\boldsymbol{x}^{(\text{Model})} - \bar{\boldsymbol{x}}^{(\text{Data})}||$$

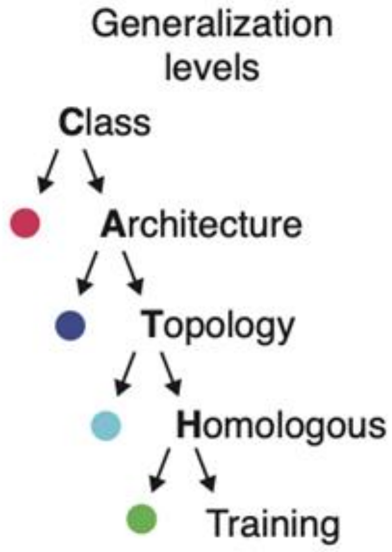
Method - Dataset

From the **CATH** hierarchical classification of protein folds [3]

Which hierarchically organizes proteins from the Protein Data Bank [4] into domains that are classified at the levels of **C**lass, **A**rchitecture, **T**opology, and **H**omologous superfamily

Difficult level: C > A > T > H

- ❑ Training: 35k folds
- ❑ Validation: 21k folds
- ❑ Test: 10k folds



Method - Representation of Sequence

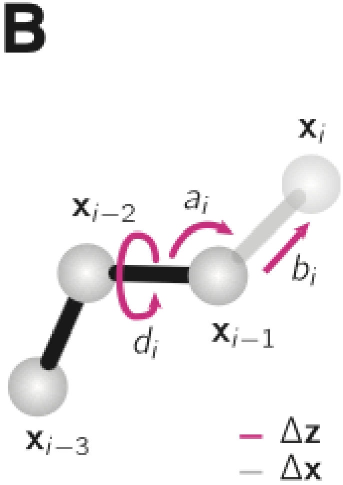
NEMO considers two modes for conditioning our model on sequence information:

1. **1-seq,** L x 20 matrix containing a one-hot encoding of the amino acid sequence
2. **Profile**, L x 40 matrix encoding both the amino acid sequence and a profile of evolutionarily related sequences (Frequencies of amino acids at that position in the MSA, computed from aligned homologous sequences)

   E.g. [0,0,1,0,...,0 | 0.05,0.02,0.70,0.01,...]

*L: protein sequence length

Method - Representation of Protein

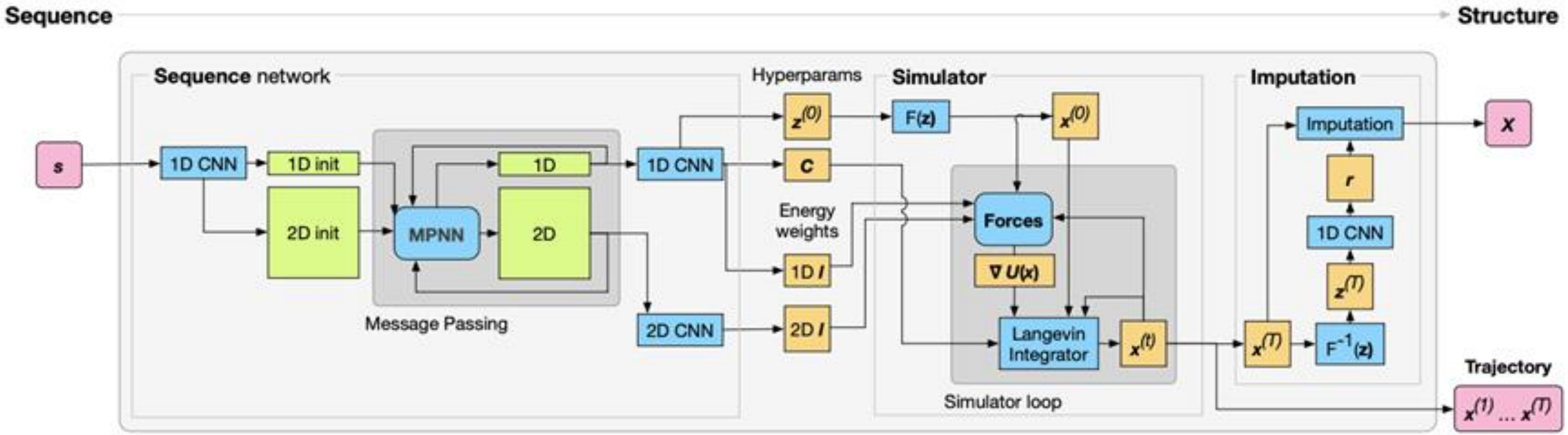


Global coordinates x: Cartesian coordinates

Internal coordinates z: $z_i = \{\tilde{b}_i, \tilde{a}_i, d_i\}$

- ❑ bond length $b_i \in (0, \infty)$
- ❑ bond angle $a_i \in [0, \pi)$
- ❑ dihedral angle $d_i \in [0, 2\pi)$
  - ❑ Plane A formed by x_i, x_{i-1}, x_{i_2}
  - ❑ Plane B formed by x_{i-1}, x_{i_2}, x_{i-3}

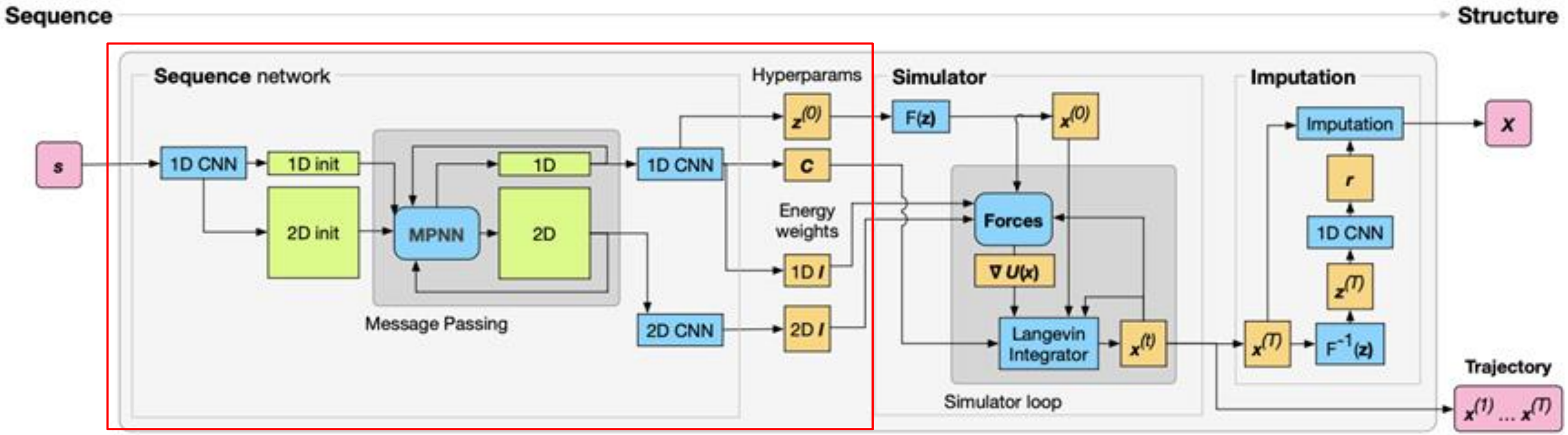
Method - NEMO Model Structure



NEMO is an end-to-end differentiable model of protein structure conditioned on sequence information:

1. **the sequence network** defines what the energy landscape looks like
2. **the simulator network** performs folding on that landscape
3. **the imputation network** converts the final coarse fold into full atomic structure

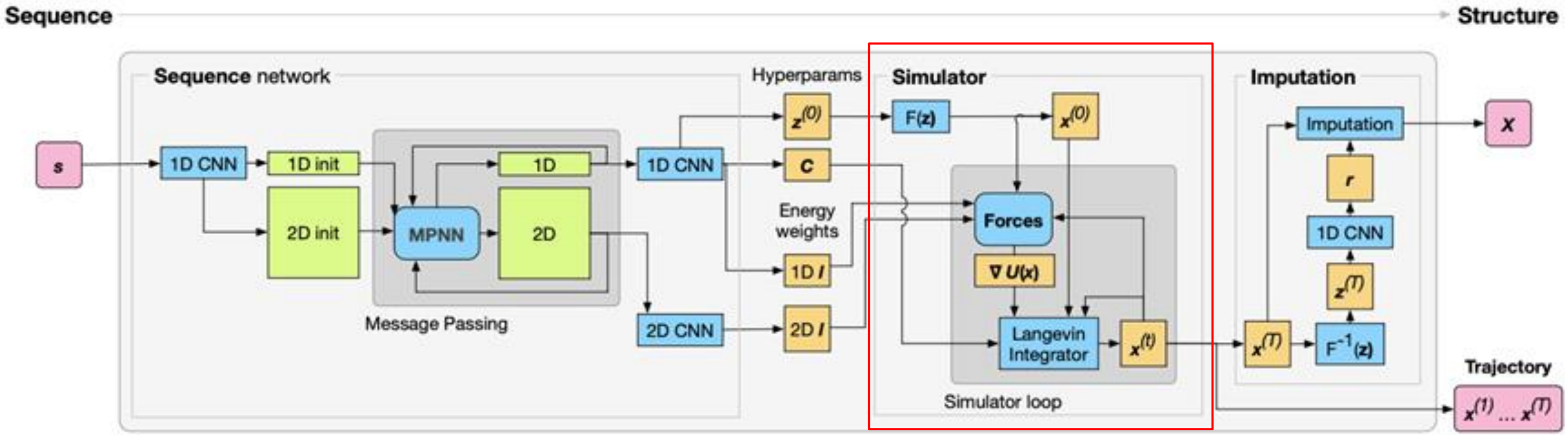
Method - Sequence Network



The sequence network prepare inputs needed for simulation

- Inputs: protein sequence (+ evolutionary profile)
- Outputs to the simulator:
  - Initial coarse structure z: a rough starting backbone
  - Energy weights that parameterize the energy function
  - Simulator hyperparameters that control how folding proceeds

Method - Simulator Network



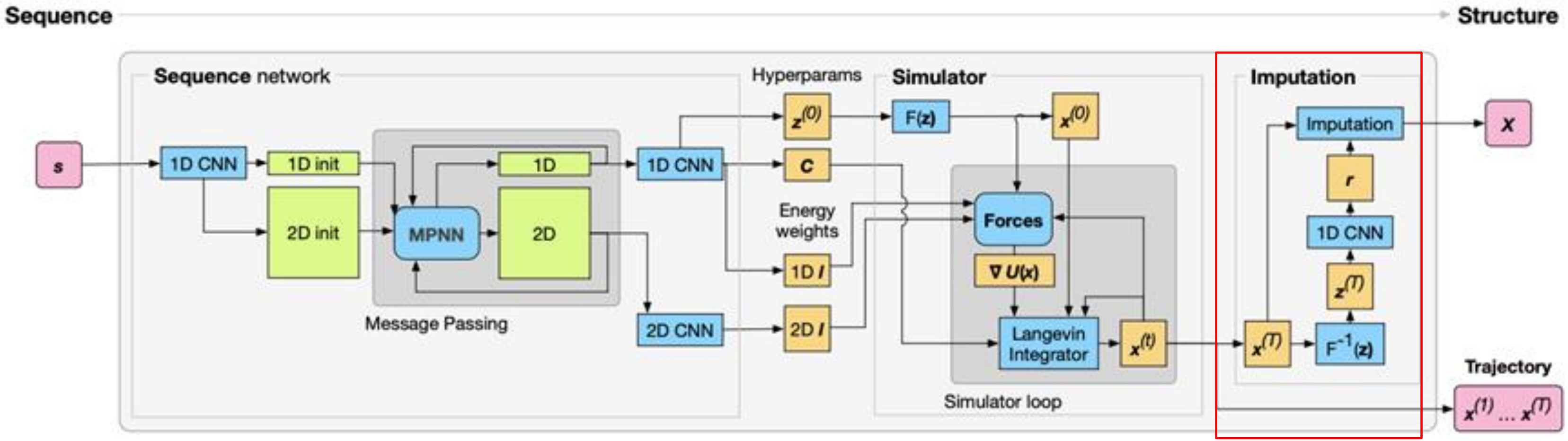
The simulator is the engine that actually folds the protein.

- ❖ Inputs: Initial structure from the sequence network + Energy function U(x|s)
- ❖ Operation, at each step (repeated for a fixed number of steps ≈250):
  - ➢ Compute ∇U (forces from the energy)
  - ➢ Add noise (Langevin dynamics)
  - ➢ Update the structure slightly

$$\boldsymbol{x}^{(t+\epsilon)} \leftarrow \boldsymbol{x}^{(t)} - \boxed{\frac{\epsilon}{2}\nabla_{\boldsymbol{x}} U^{(t)}} + \boxed{\sqrt{\epsilon}\mathbf{p},} \qquad \mathbf{p} \sim \mathcal{N}(0, I).$$

- ❖ Output: a coarse-grained protein structure

Method - Imputation Network



The imputation network refines the protein structure representation

- ❖ Takes the final coarse structure x^(T)
- ❖ Uses local reference frames
- ❖ Output the atomic model X

Method - Loss

$$\mathcal{L} = \mathcal{L}_{\text{ER}} + \mathcal{L}_{\text{ML}}$$

Loss = Likelihood loss + Empirical risk Loss

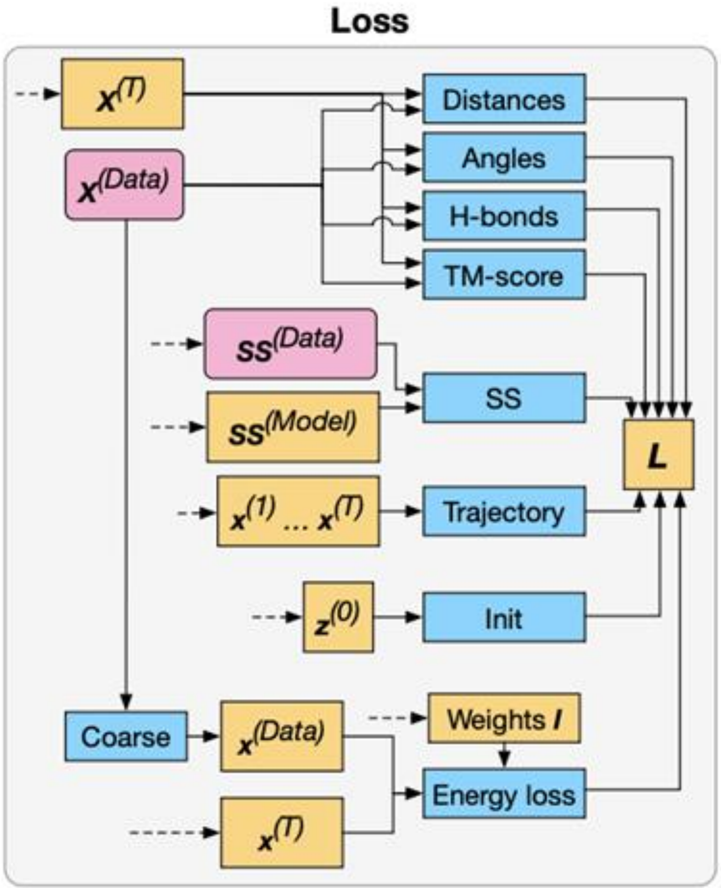
$$\mathcal{L}_{\text{ML}} = U_\theta(\perp(\boldsymbol{x}^{(\text{D})}); \boldsymbol{s}) - U_\theta(\perp(\boldsymbol{x}^{(\text{M})}); \boldsymbol{s})$$

❑ The likelihood loss minimizes the average energy of samples from the data relative to samples from the model

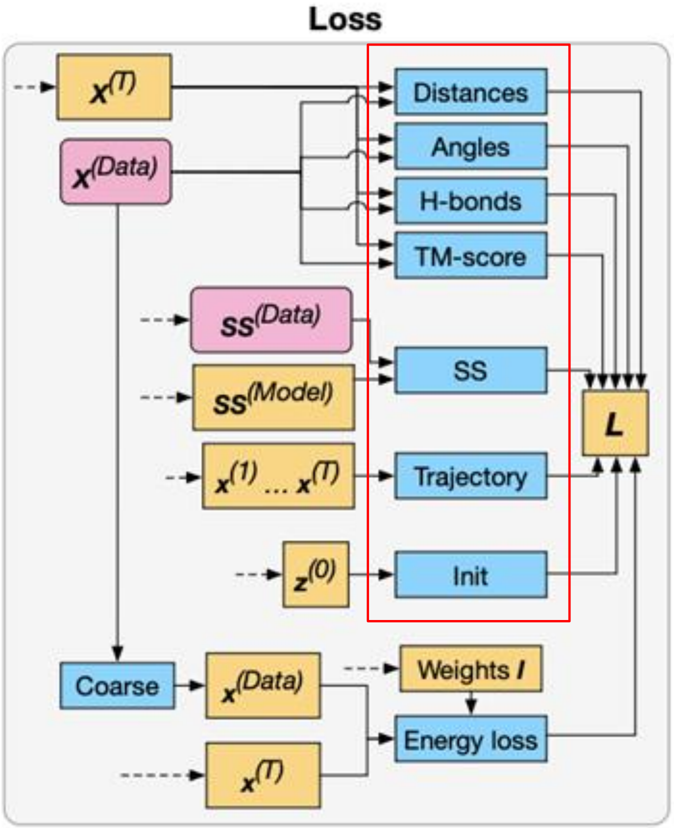
$$\mathcal{L}_{\text{ER}} = \mathcal{L}_{\text{Distances}} + \mathcal{L}_{\text{Angles}} + \mathcal{L}_{\text{H-bonds}}$$

The empiric $+ \mathcal{L}_{\text{TM-score}} + \mathcal{L}_{\text{Init}} + \mathcal{L}_{\text{Trajectory}}$

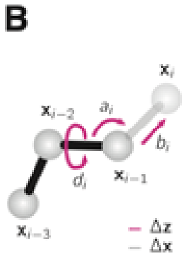
…



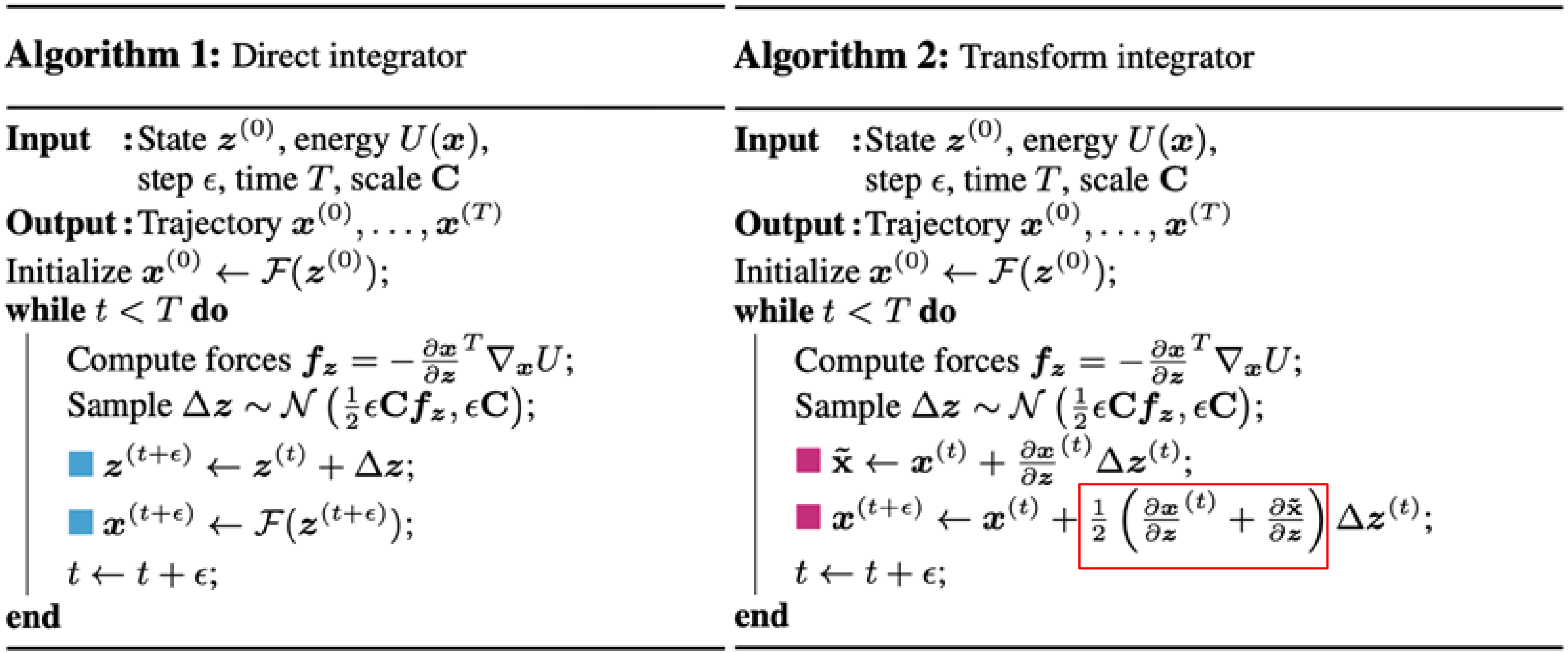
Method - Empirical Loss



- ❖ Distance-based loss: inter-residue distances
- ❖ Angular loss: a, d angles in internal coordinates
- ❖ Hydrogen bond loss: cross-entropy on binary hydrogen-bond labels
- ❖ Template modelling score
- ❖ Secondary structure prediction loss: crc entropy of 8-class predictions of seconc structure
- ❖ Trajectory loss: same distance-based features but applied to intermediate simulated structures
- ❖ Init? Distance-based loss on initial internal coordinates



Highlight: Faster Training via Transformer Integrator



Algorithm 1: Direct integrator

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

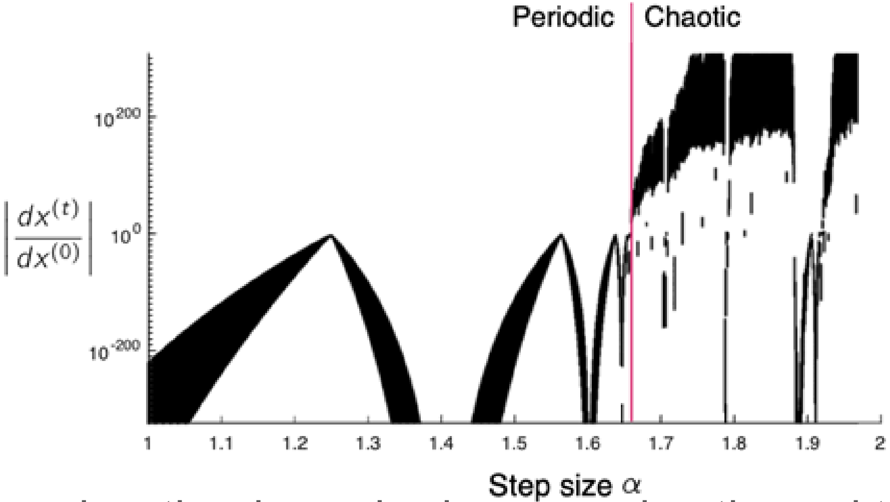
Algorithm 2: Transform integrator

```
Input  : State $z^{(0)}$, energy $U(x)$,
         step $\epsilon$, time $T$, scale $\mathbf{C}$
Output : Trajectory $x^{(0)}, \dots, x^{(T)}$
Initialize $x^{(0)} \leftarrow \mathcal{F}(z^{(0)})$;
while $t < T$ do
    Compute forces $f_z = -\frac{\partial x}{\partial z}^T \nabla_x U$;
    Sample $\Delta z \sim \mathcal{N}\left(\frac{1}{2}\epsilon \mathbf{C} f_z, \epsilon \mathbf{C}\right)$;
    🟪 $\tilde{\mathrm{x}} \leftarrow x^{(t)} + \frac{\partial x}{\partial z}^{(t)} \Delta z^{(t)}$;
    🟪 $x^{(t+\epsilon)} \leftarrow x^{(t)} + \boxed{\frac{1}{2}\left(\frac{\partial x}{\partial z}^{(t)} + \frac{\partial \tilde{\mathrm{x}}}{\partial z}\right)} \Delta z^{(t)}$;
    $t \leftarrow t + \epsilon$;
end
```

- ➢ Push the internal-coordinate update through the Jacobian
- ➢ Avoiding costly geometry reconstruction at every step

Highlight- More Stable Simulation via Regularization

Gradient



After some critical step size, the dynamics become chaotic, and the gradients regularly diverge to huge numbers, this is solved by:

1. Lyapunov regularization: to constrain the simulator update step so that small changes in the input state cannot cause large changes in the next state
2. Damped backpropagation through time: Gradients from earlier time steps are multiplied by a decay factor γ, $g_t \leftarrow \gamma^t g_t$ so that later simulation steps matter more

Result

Evaluation

- Test Dataset: 10k protein structures
- Evaluation steps:
    1. Given protein sequence, sample 100 models from NEMO
    2. Cluster by structural similarity
    3. Select a representative structure
- Metrics: TM-Score [0,1]
- TM>0.5 = good approximation
- Baseline model: RNN (backbone: a two-layer bidirectional LSTM) that directly predicts the structure without folding procedure

Faster Inference than Classical Sampling-based Methods

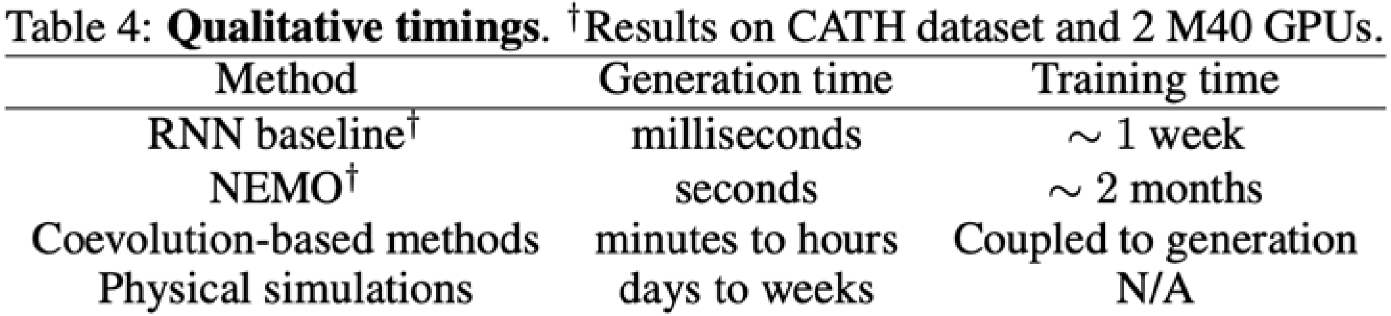
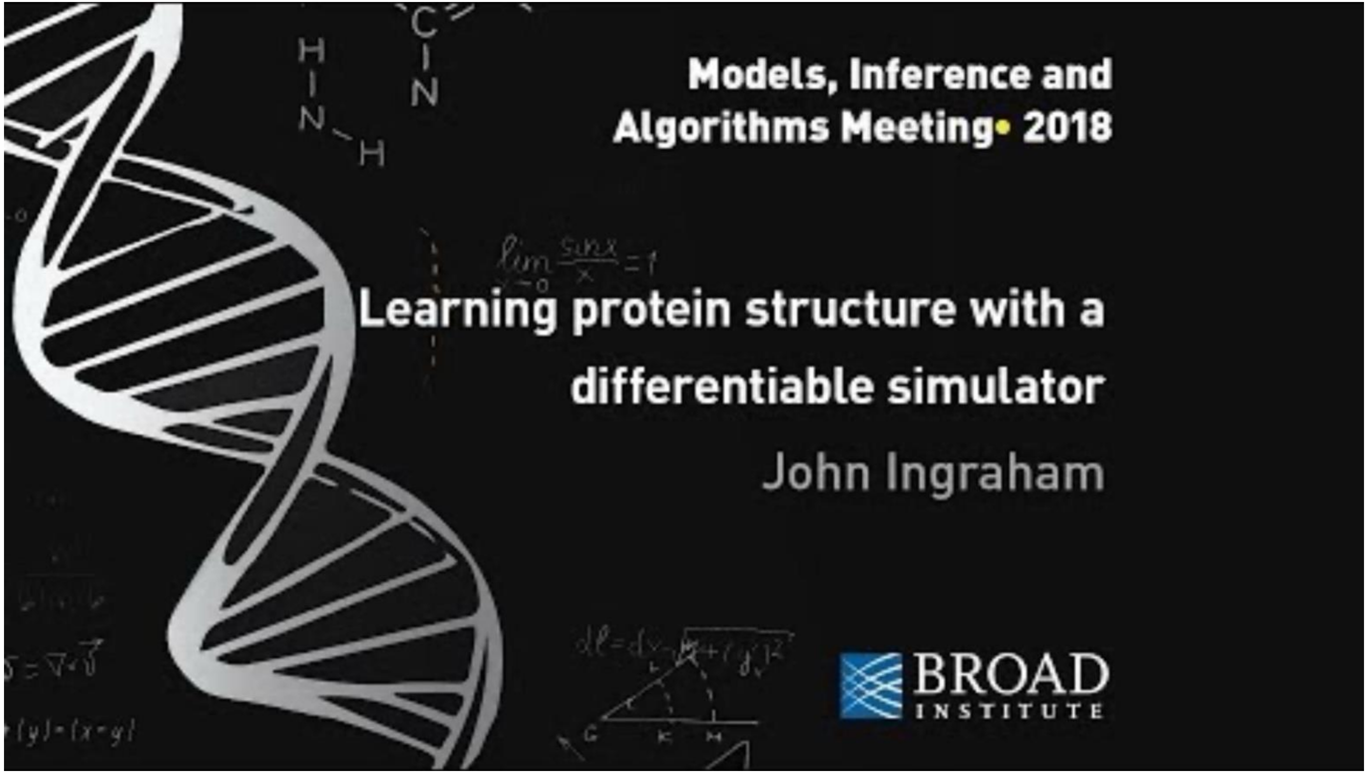


Table 4: **Qualitative timings**. †Results on CATH dataset and 2 M40 GPUs.

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

- ➢ Model sampling (inference) times can be considerably faster than coevolution-based methods and physical simulations in protein structure prediction, but slower than RNN baselines

Realistic 3D Structure Prediction for Small Proteins



Models, Inference and Algorithms Meeting • 2018

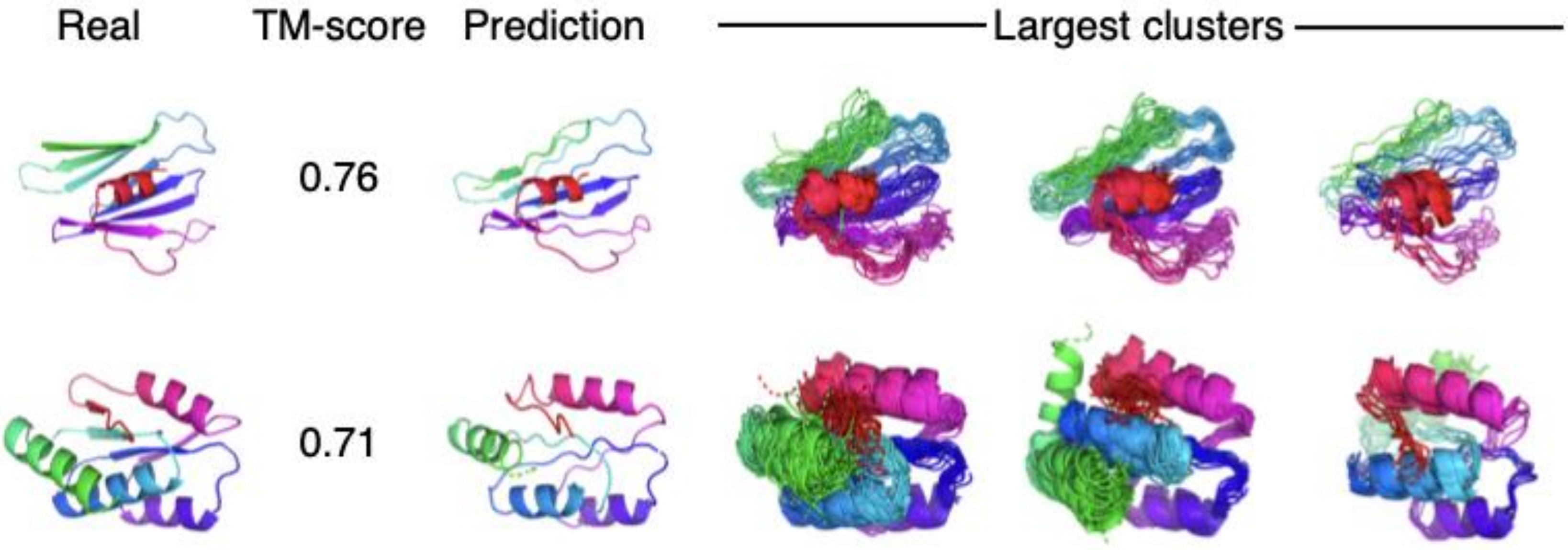
Learning protein structure with a differentiable simulator

John Ingraham

BROAD INSTITUTE

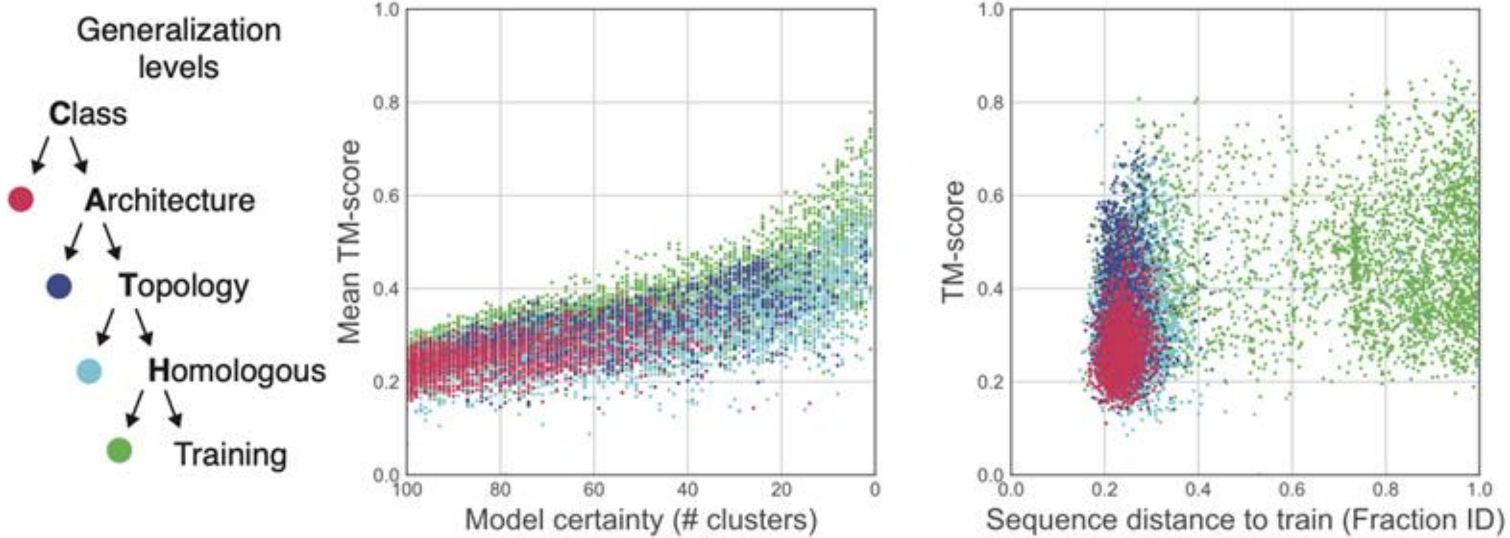
➢ NEMO produces realistic and accurate approximations of the protein structure

Captures Structural Uncertainty via Stochastic Simulation



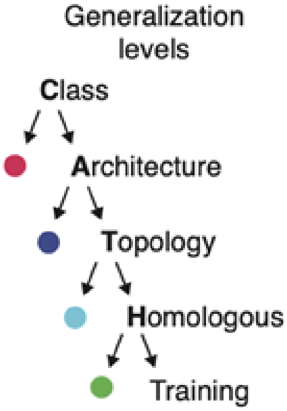
- ➢ The uncertainty along the chain sometimes indicates loosely packed regions of the protein

More Confident Predictions have Higher Accuracy



- ➢ If NEMO is confident - the number of distinct structural clusters is low ~1-3, it is also accurate with predictions, having average TM >0.5
- ➢ NEMO occasionally achieves accurate predictions to unseen data

Better Generalization to Distant Folds than Baselines



Generalization levels

Class

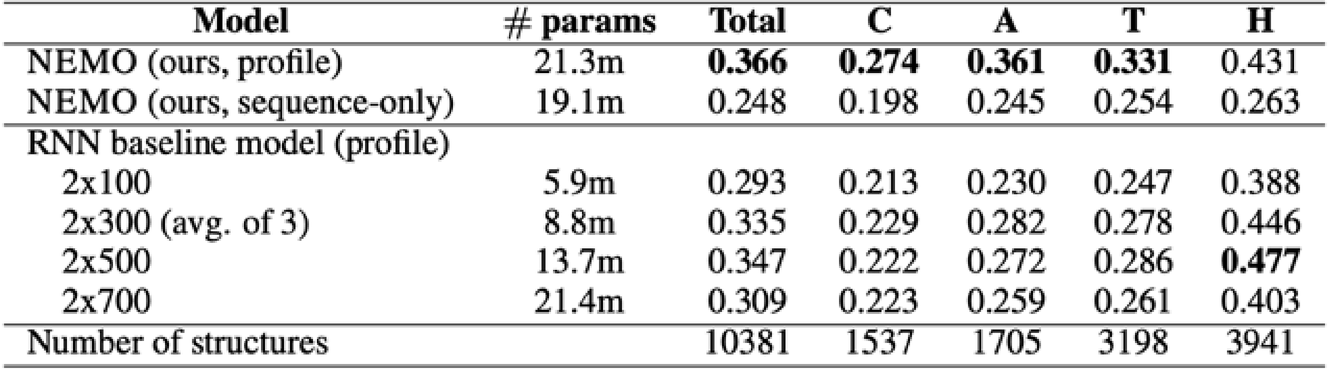
● Architecture

● Topology

● Homologous

● Training

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

- ➢ NEMO (sequence + profile) outperforms baseline model in harder tasks
- ➢ The availability of evolutionary information can significantly improve prediction quality

Discussion

Challenges

Why protein structure prediction task is challenging?

What challenges do ML-based protein prediction model face?

General Challenges in Structure Prediction

- ❖ Trade-off between physical fidelity and computational tractability
  - ➢ High-resolution physics is accurate but slow; coarse or learned models scale better but lose mechanistic grounding.
- ❖ Evaluation and ground-truth ambiguity
  - ➢ Experimental structures may differ by condition; RMSD/TM-score incompletely capture biological relevance.
- ❖ Modeling conformational heterogeneity and dynamics
  - ➢ Proteins exist as ensembles; most predictors output a single static structure.
- ❖ Scalability to large proteins and complexes
  - ➢ Memory and computation scale poorly with sequence length.
- ❖ Generalization to novel folds and rare proteins
  - ➢ Data-driven models struggle outside the training distribution.

Specific Challenges for ML-based Models

- ❖ Intractable sampling and slow convergence of energy-based models
    - ➢ Classical Boltzmann or energy-based models are expressive but impractical due to slow Monte Carlo sampling.
- ❖ Trade-off between global fold exploration and local structural refinement
    - ➢ Cartesian dynamics favor local refinement; internal coordinates favor global topology—neither alone is sufficient.
- ❖ High computational cost of internal-coordinate simulations
    - ➢ Internal coordinates enable global motions but require expensive, sequential Cartesian reconstruction.
- ❖ Dependence on evolutionary information
    - ➢ Without coevolutionary information or templates, often fail to reach correct global minima.
- ❖ Chaotic dynamics and exploding gradients in long unrolled simulations
    - ➢ Backpropagation through long Langevin roll-outs is unstable; gradients can explode catastrophically.
- ❖ Limited scalability compared to direct predictors
    - ➢ Training and inference are slower than angle- or distance-predicting RNNs; limits dataset scale.

Advantages and Disadvantages of NEMO

Advantages

Disadvantages

Advantages and Disadvantages of NEMO

Advantages:

1. Combines physical inductive bias with directed inference
2. Much faster sampling than conventional physics-based methods
3. Produces ensembles, not a single structure
4. Improves generalization to unseen folds

Disadvantages:

1. Higher training and sampling cost than angle-predicting RNNs
2. Hard to scale to very large datasets
3. Training instability (e.g. gradient damping) may limit expressiveness

Comparison to Alphafold1

Similarity

Difference

Comparison to Alphafold1

Similarity:

1. Sequence-conditioned modeling
2. Gradient-based structure generation

Difference:

NEMO	AlphaFold1
Learns energy + simulator	Learns distance distributions
Produces structure ensembles	Produces single structure
Stochastic sampling	Optimization

AlphaFold1 learns what structure should look like, NEMO learns the folding trajectory and preserve uncertainty.

Open Questions

Only one baseline?

Conclusions

Protein structure prediction as a learned, differentiable folding process

- Method:
    - Sequence-conditioned neural energy function
    - Unrolled Langevin simulator trained end-to-end by backpropagation
- Results:
    - Predicts realistic 3D structures for proteins with hundreds of atoms
    - Faster inference than classical sampling-based methods
    - Captures structural uncertainty via stochastic simulation
    - Better generalization to distant folds than strong baselines

Future directions

- **More efficient simulators**: Sub-quadratic N-body dynamics to reduce training and sampling cost
- **Better training stability**: Principled methods for stabilizing long unrolled simulations beyond gradient damping
- **More expressive energy functions**: Reduce constraints imposed by Lipschitz regularization while maintaining stability
- **Scaling to larger datasets**: Longer proteins and greater structural diversity

Reference

[1] Yazhini, A., Chakraborti, S., Srinivasan, N. (2021). Protein Structure, Dynamics and Assembly: Implications for Drug Discovery. In: Singh, S.K. (eds) Innovations and Implementations of Computer Aided Drug Discovery Strategies in Rational Drug Design. Springer, Singapore. https://doi.org/10.1007/978-981-15-8936-2_5

[2] Glaeser, R. M. (2016). How good can cryo-EM become?. Nature methods, 13(1), 28-32.

[3] Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., & Thornton, J. M. (1997). CATH–a hierarchic classification of protein domain structures. Structure, 5(8), 1093-1109.

[4] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., ... & Bourne, P. E. (2000). The protein data bank. Nucleic acids research, 28(1), 235-242.