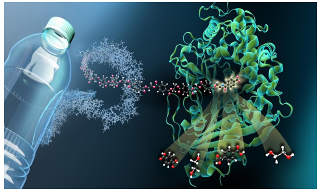
Repurposing Protein Folding Models for All-Atom Generation via Latent Space Compression

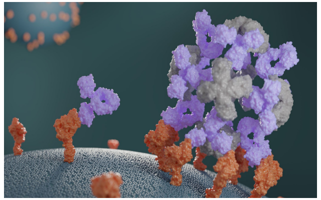
Amy X. Lu
October 1, 2025
In silico #002

Designing novel proteins with desired properties is hard



Plastic degrading enzymes

Vaccine development



Antibody therapeutics

and much more…

HUMAN PROTEINS

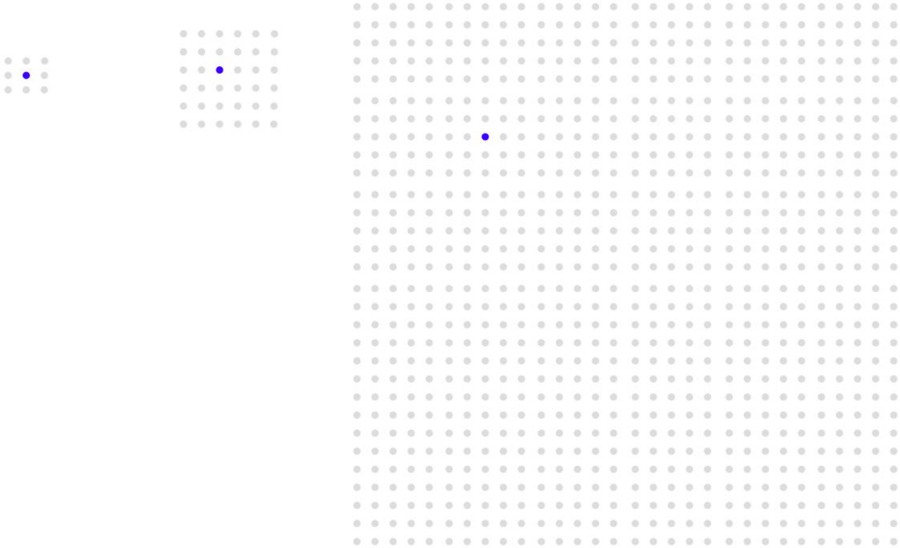
~20,000

UNIQUE PROTEINS ON EARTH

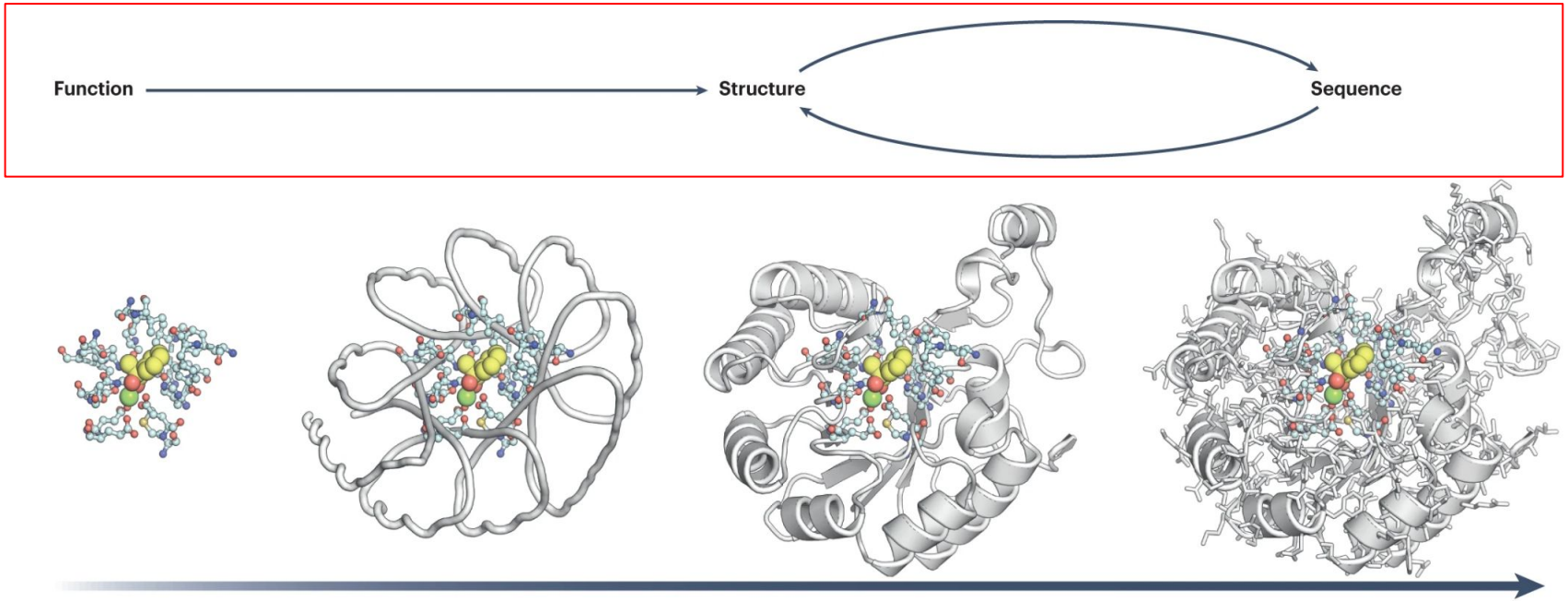
~1,000,000,000,000,000,000

POSSIBLE PROTEINS

~10,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000,000,000,000,
000,000,000,000,000,000,000,000,000

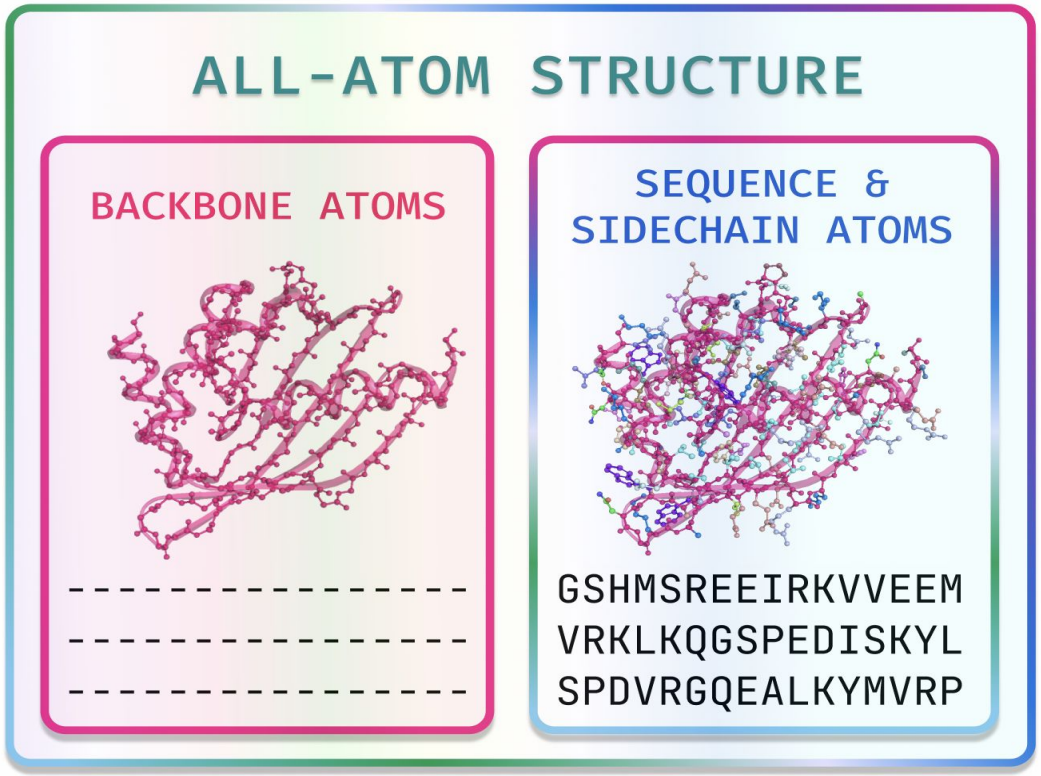


Current all-atom methods iterate between sequence and structure design



Chu, A.E., Lu, T. & Huang, PS. **Sparks of function by de novo protein design**. Nat Biotechnol 42, 203–215 (2024).

All-atom design as a multimodal generation problem



Placing the continuous sidechain atom positions require knowing the discrete sequence.

→ Can we sample from p(sequence, structure)?

Motivation: Can we repurpose priors from pretrained models?

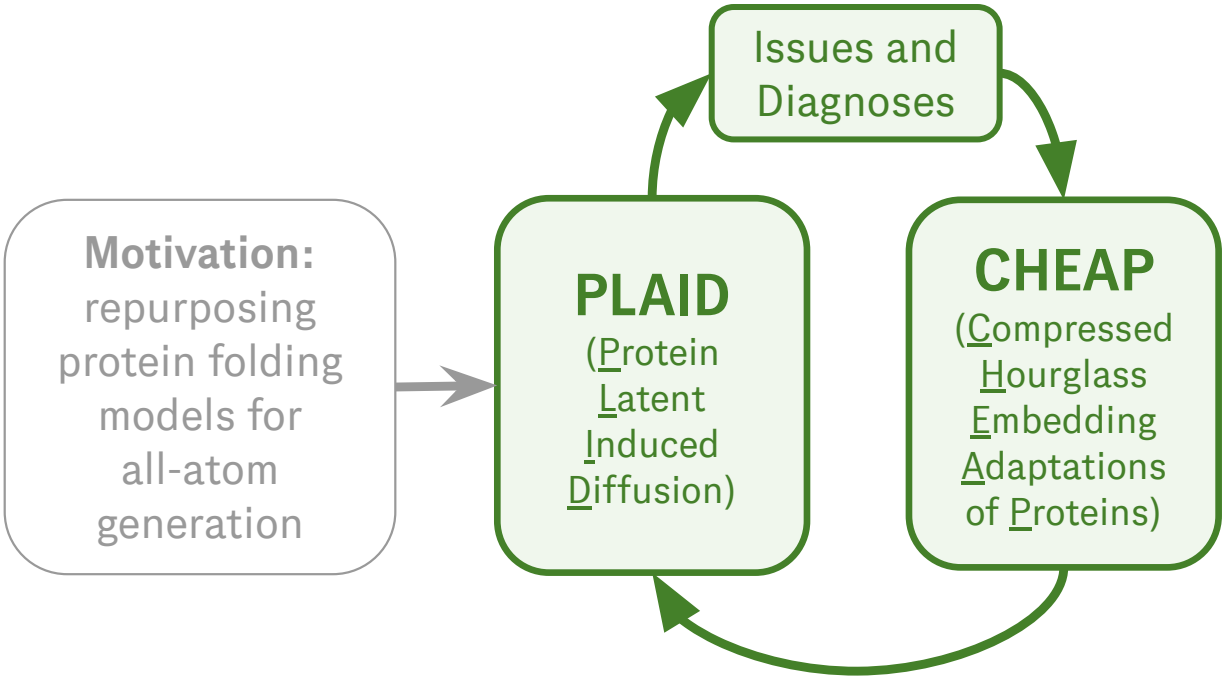
RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

Vision-language models trained on internet-scale datasets capture useful priors for robotic tasks.

How can we apply this to biology?

Can we sample all-atom structure from the joint distribution p(sequence, structure) and use priors from pretrained protein folding models?

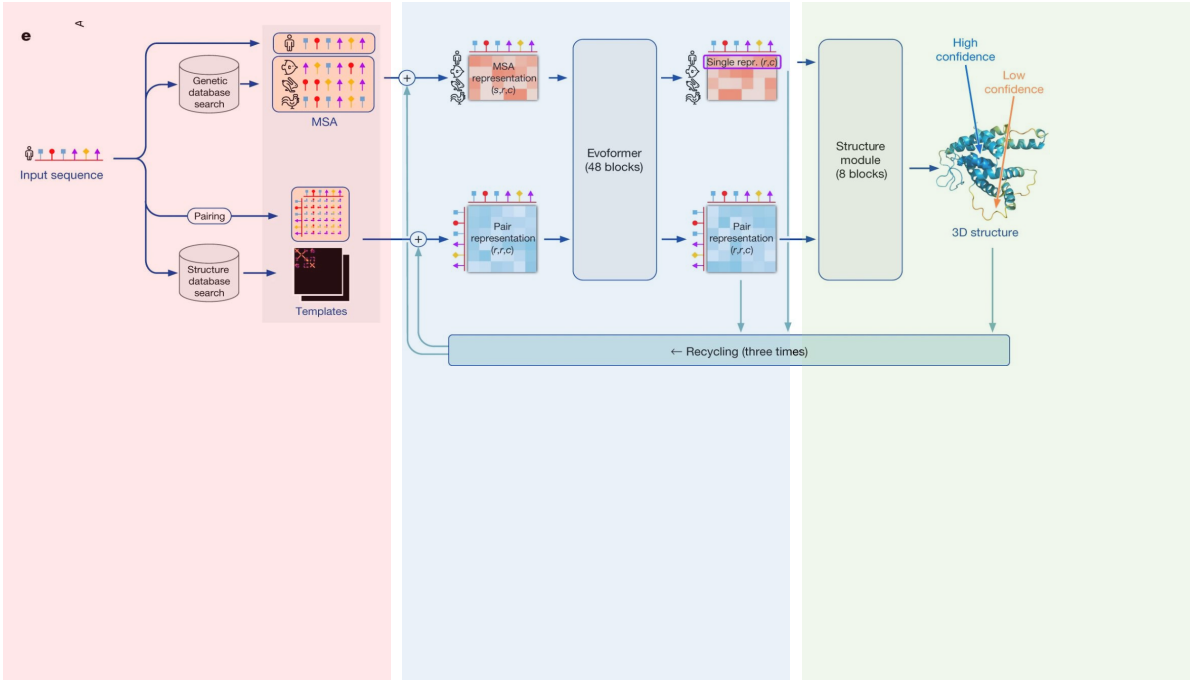
Agenda


Issues and Diagnoses
Motivation: repurposing protein folding models for all-atom generation
PLAID (Protein Latent Induced Diffusion)
CHEAP (Compressed Hourglass Embedding Adaptations of Proteins)

The base components: protein folding model architectures

AlphaFold2:

Uses an explicit retrieval step



harness additional sequence-based priors

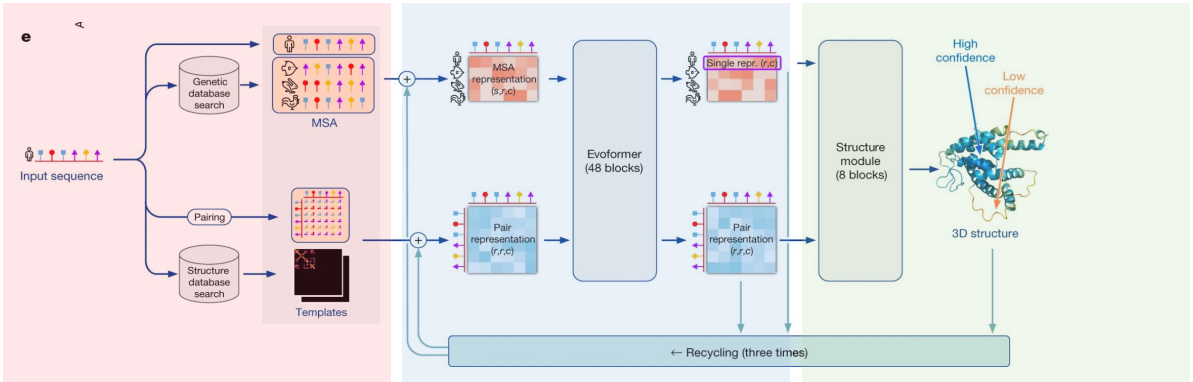
learn structural features from sequence latents

generate structures

The base components: protein folding model architectures

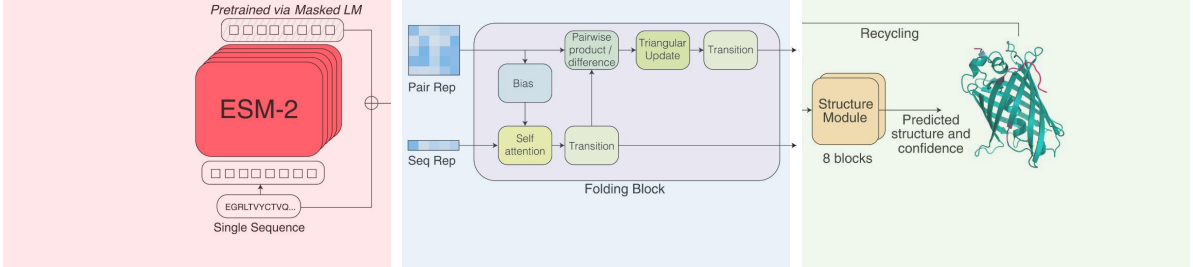
AlphaFold2:

Uses an explicit retrieval step



ESMFold:

Replaces retrieval step with a **language model**



harness additional sequence-based priors

learn structural features from sequence latents

generate structures

main esm / esm / esmfold / v1 / esmfold.py ↑ Top

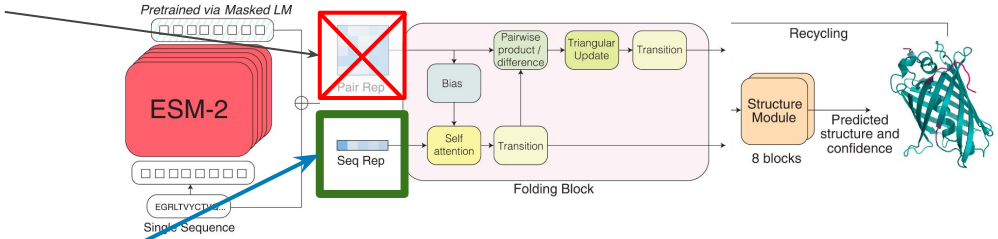
Code Blame 364 lines (305 loc) · 13.6 KB Raw

```
152     def forward(
185         # === ESM ===
186         esmaa = self._af2_idx_to_esm_idx(aa, mask)
187
188         if masking_pattern is not None:
189             esmaa = self._mask_inputs_to_esm(esmaa, masking_pattern)
190
191         esm_s, esm_z = self._compute_language_model_representations(esmaa)
192
193         # Convert esm_s to the precision used by the trunk and
194         # the structure module. These tensors may be a lower precision if, for example,
195         # we're running the language model in fp16 precision.
196         esm_s = esm_s.to(self.esm_s_combine.dtype)
197         esm_s = esm_s.detach()
198
199         # === preprocessing ===
200         esm_s = (self.esm_s_combine.softmax(0).unsqueeze(0) @ esm_s).squeeze(2)
201
202         s_s_0 = self.esm_s_mlp(esm_s)
203         if self.cfg.use_esm_attn_map:
204             esm_z = esm_z.to(self.esm_s_combine.dtype)
205             esm_z = esm_z.detach()
206             s_z_0 = self.esm_z_mlp(esm_z)
207         else:
208             s_z_0 = s_s_0.new_zeros(B, L, L, self.cfg.trunk.pairwise_state_dim)
209
210         s_s_0 += self.embedding(aa)
211
212         structure: dict = self.trunk(
213             s_s_0, s_z_0, aa, residx, mask, no_recycles=num_recycles
214         )
```

Observation: at inference, the pairwise input is initialized as zeros…

Observation: at inference, the pairwise input is initialized as zeros…

→ Sequence representation contains all information about the structure!



harness additional sequence-based priors

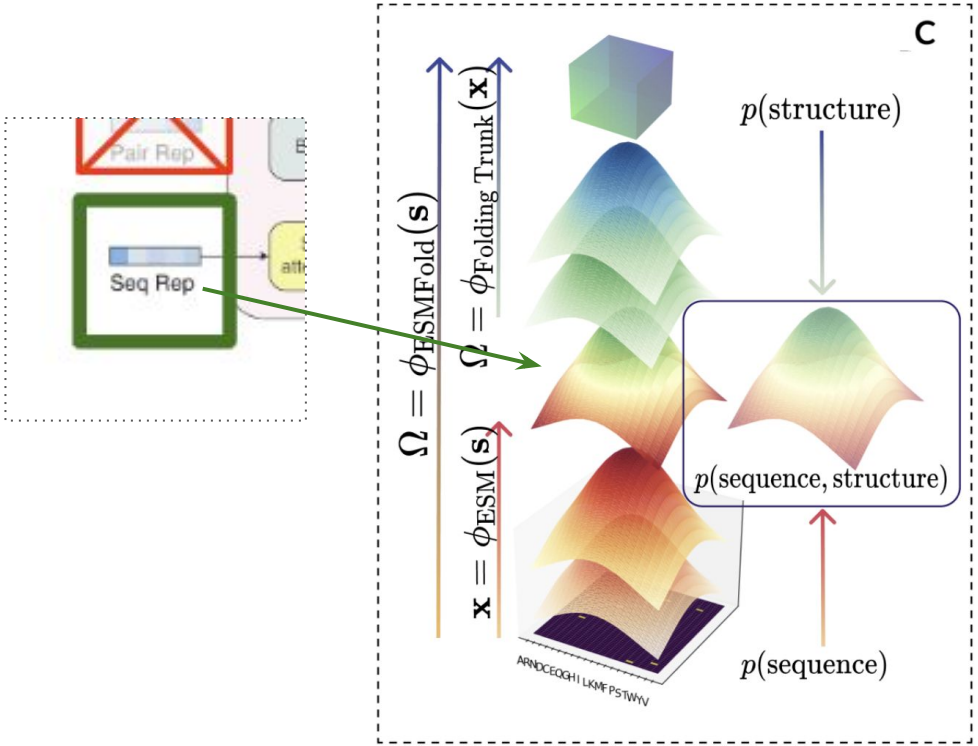
learn structural features from sequence latents

generate structures

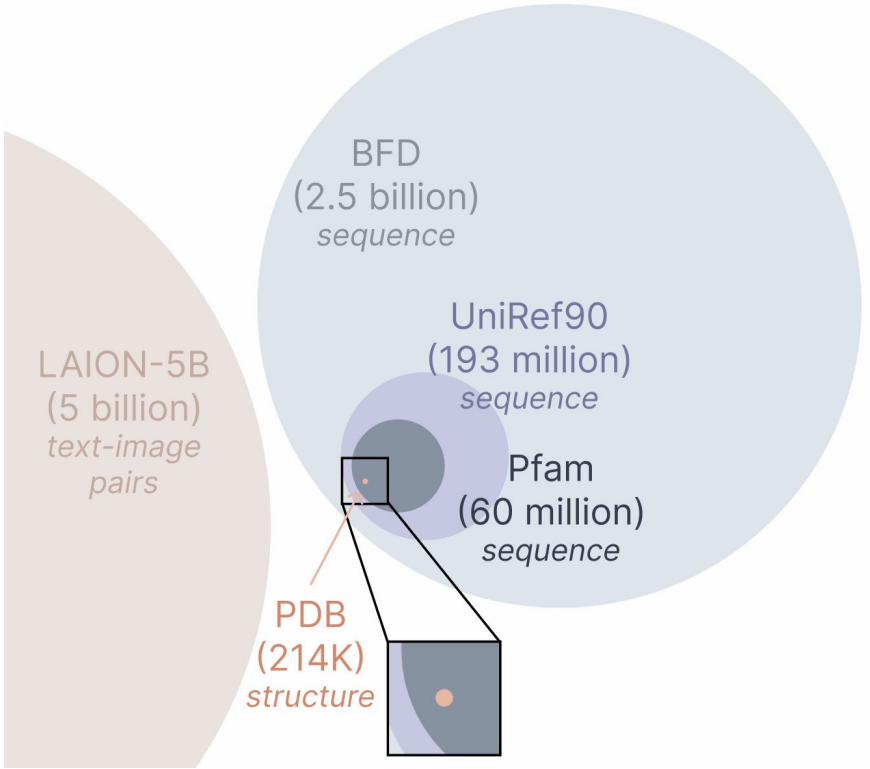
Observation: at inference, the pairwise input is initialized as zeros...

→ Sequence representation contains all information about the structure!

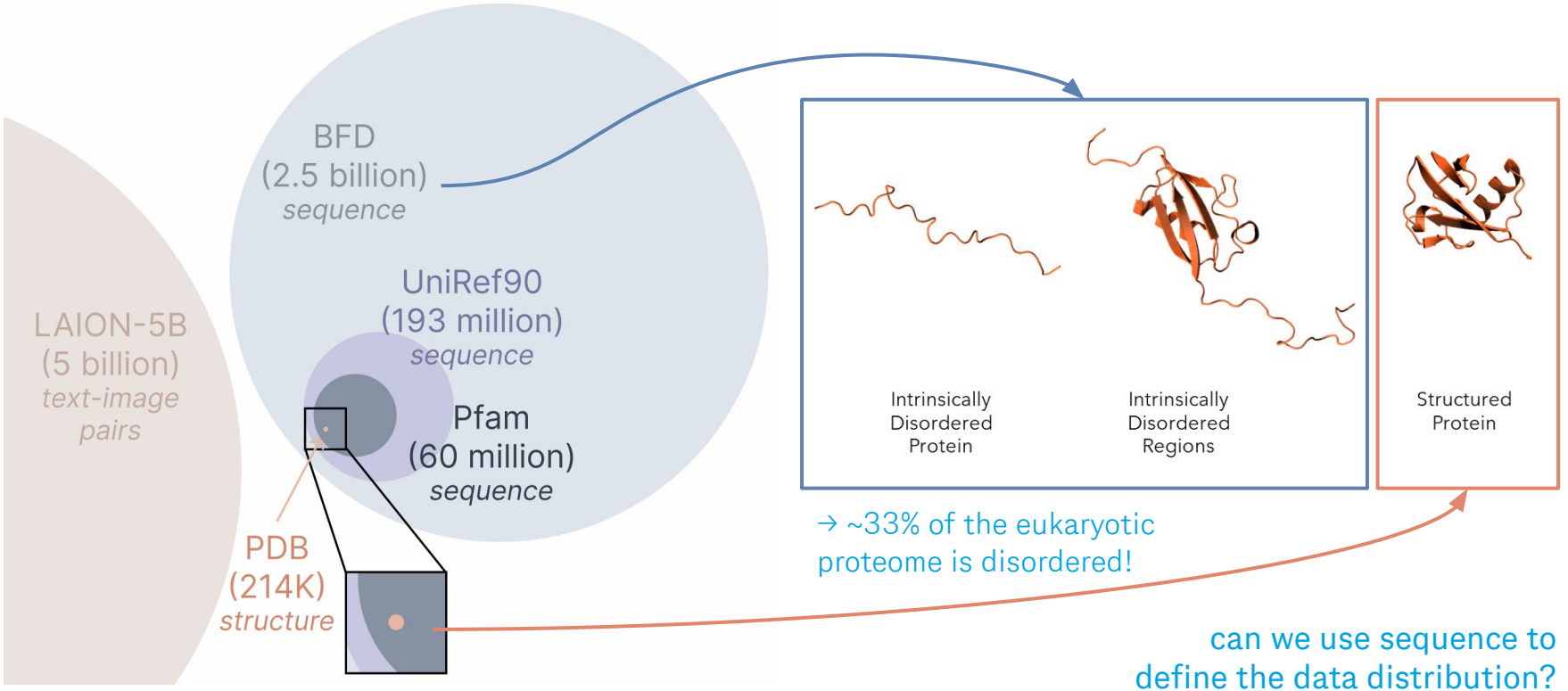
Generating this embedding would only require the sequence during training.



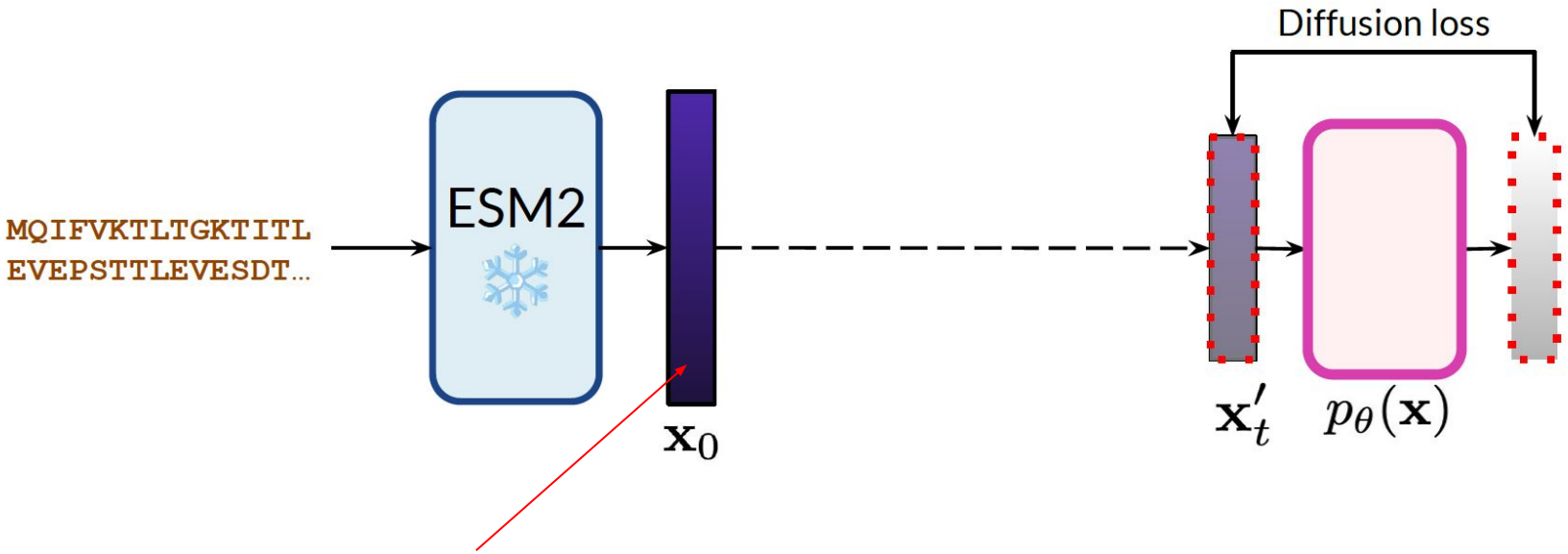
Sequence data is more abundant than structure



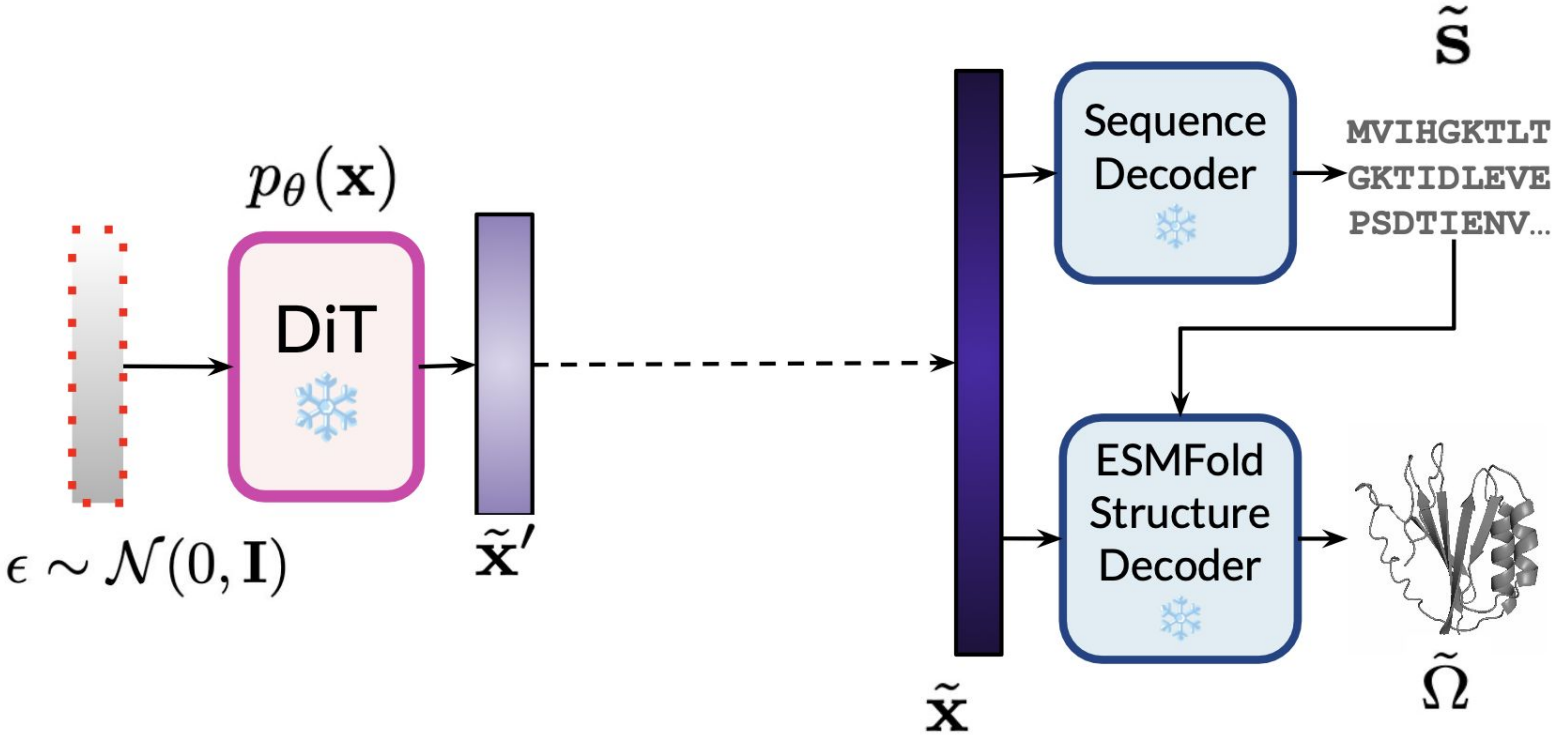
Sequence data has different coverage than structure



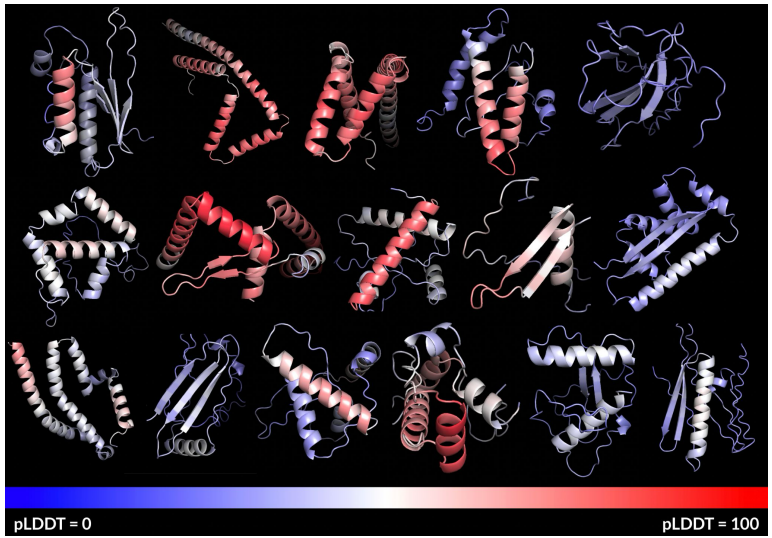
PLAID v0.5: Training a latent diffusion model

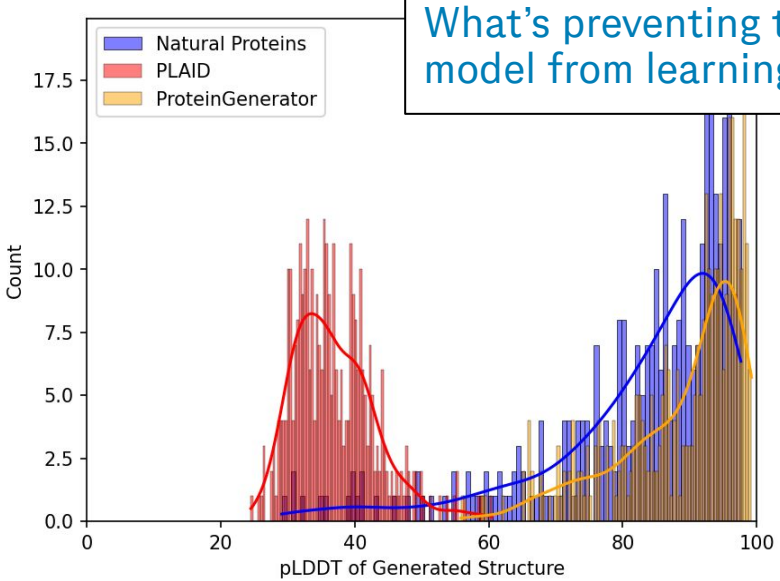


PLAID v0.5: Inference-time all-atom generation



PLAID v0.5: Early attempts

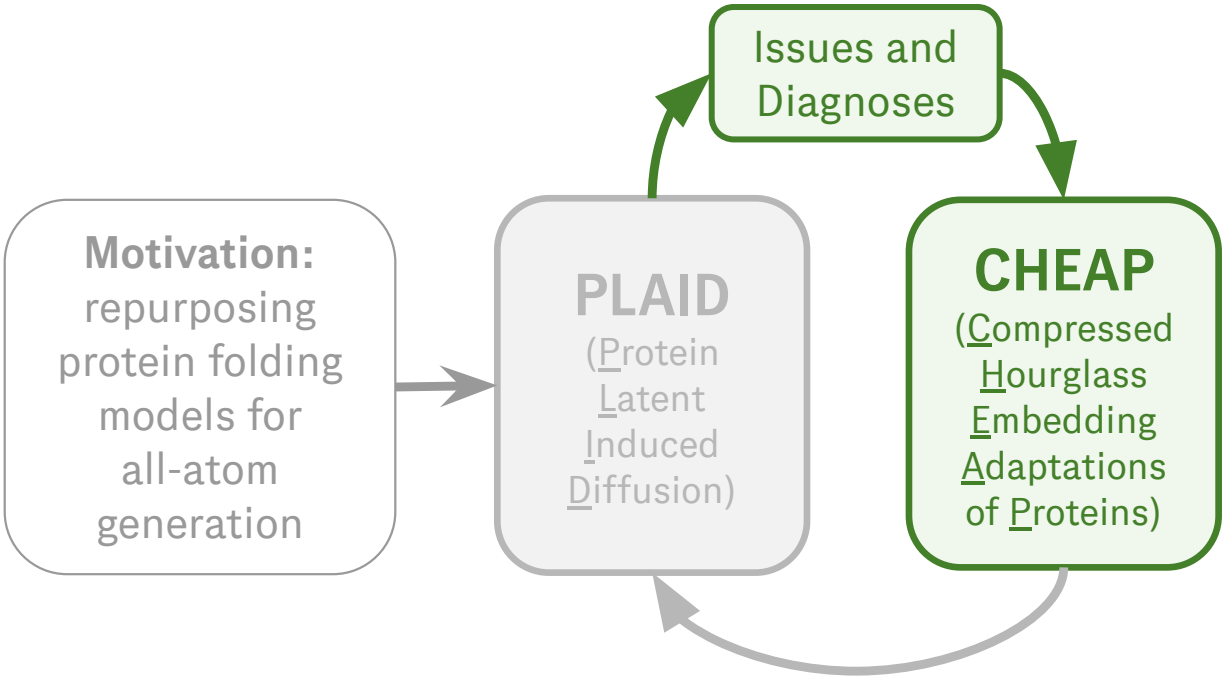




What's preventing the model from learning?

PLAID v0.5: Generating Protein Sequence and Structure Without Structural Training Data
Amy X. Lu, Kevin K. Yang, Pieter Abbeel
ICML 2024 Workshop on Machine Learning for Life and Material Sciences

Agenda


Issues and Diagnoses
Motivation: repurposing protein folding models for all-atom generation
PLAID (Protein Latent Induced Diffusion)
CHEAP (Compressed Hourglass Embedding Adaptations of Proteins)

Issues and hypotheses

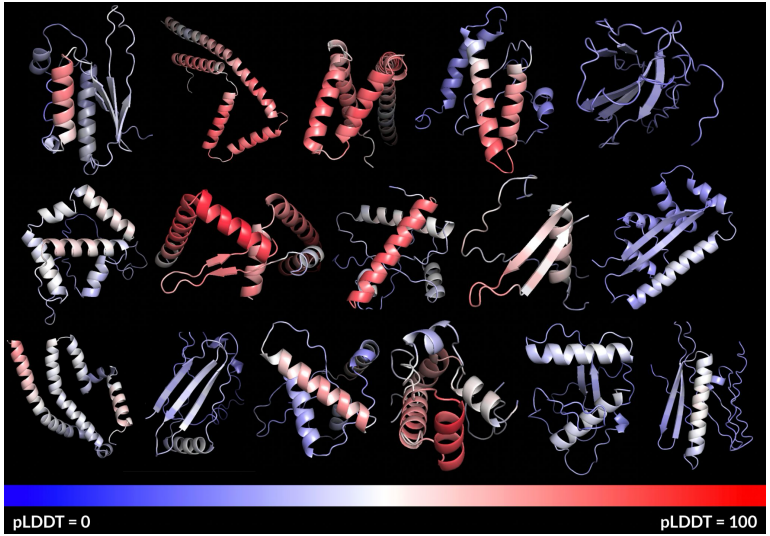
- Latent space requires regularization

In order to avoid arbitrarily high-variance latent spaces, we experiment with two different kinds of regularizations. The first variant, *KL-reg.*, imposes a slight KL-penalty towards a standard normal on the learned latent, similar to a VAE [46, 69], whereas *VQ-reg.* uses a vector quantization layer [96] within the decoder. This model can be interpreted

Rombach et al. High-Resolution Image Synthesis with Latent Diffusion Models, CVPR 2022

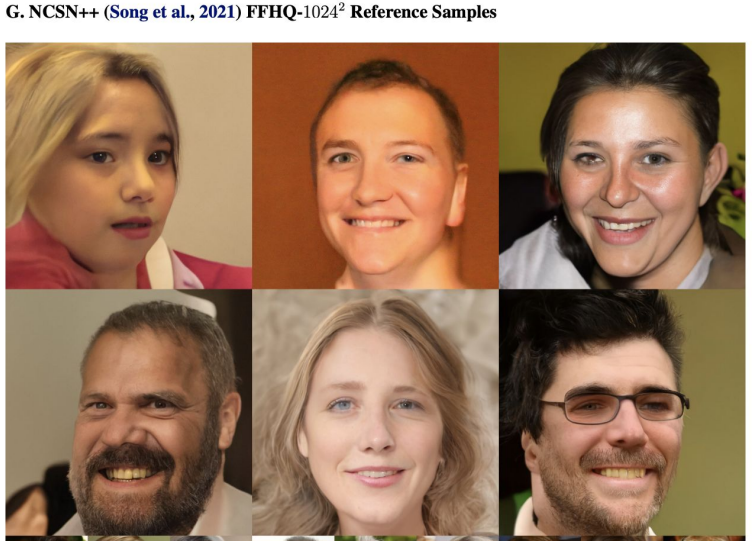
Issues and hypotheses

- Latent space requires regularization
- Overcome $O(L^2)$ memory constraints and increase protein length to 512

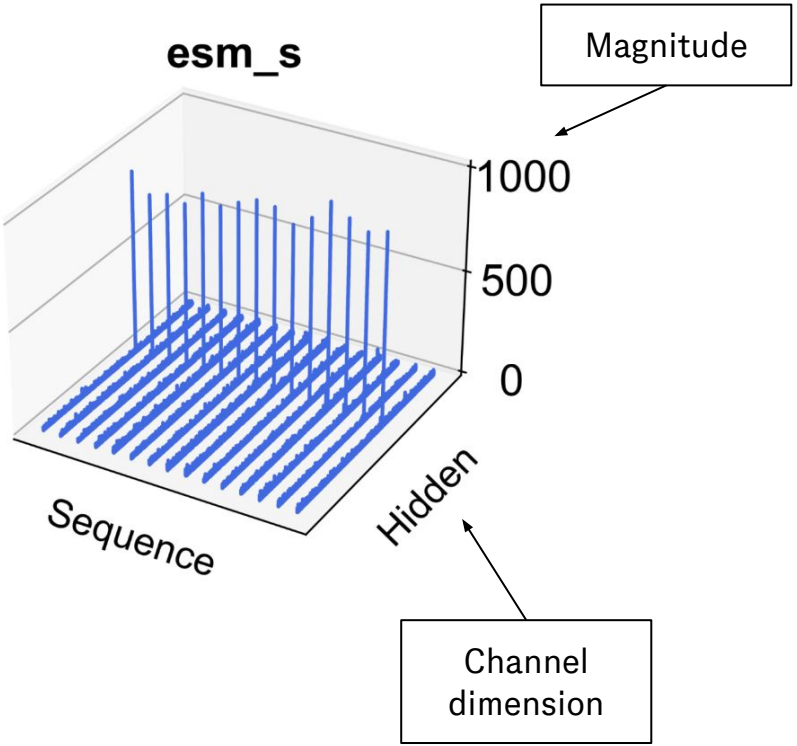


Issues and hypotheses

- Latent space requires regularization
- Overcome $O(L^2)$ memory constraints and increase protein length to 512
- Large latent space corresponds to **high-resolution** image generation
  - Rombach et al. latent space: HxWx4 = 64 x 64 x 4
  - Ours: Lx1024 = 512 x 1024



ESMFold latent space exhibits pathologically large values



Latent space will require regularization for diffusion to work.

~~ESMFold~~ Large transformers latent space exhibits pathologically large values

→ a pervasive issue across LLMs, ViTs, etc.

[Submitted on 27 Feb 2024 (v1), last revised 14 Aug 2024 (this version, v2)]

Massive Activations in Large Language Models

Mingjie Sun, Xinlei Chen, J. Zico Kolter, Zhuang Liu

We observe an empirical phenomenon in Large Language Models (LLMs) -- very few activations exhibit significantly larger values than others (e.g., 100,000 times larger). We call them massive activations. First, we demonstrate the widespread existence of

LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale

Tim Dettmers[λ*] Mike Lewis[†] Younes Belkada[§∓] Luke Zettlemoyer[†λ]

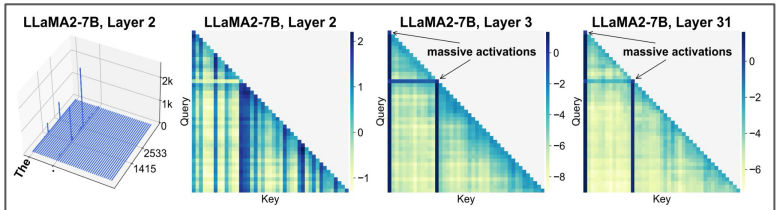
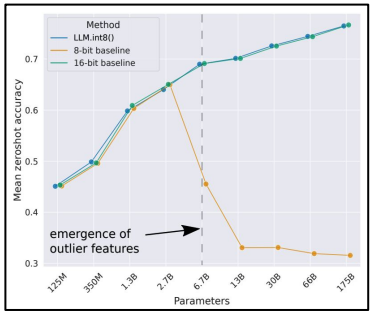
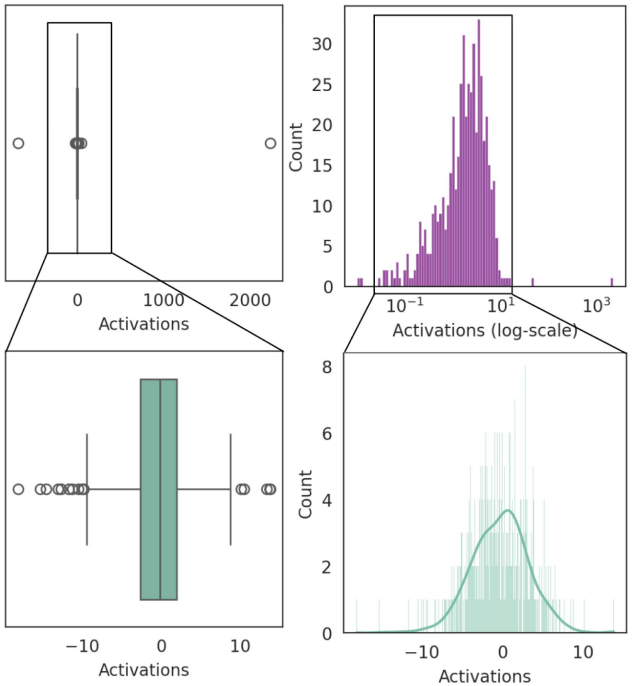


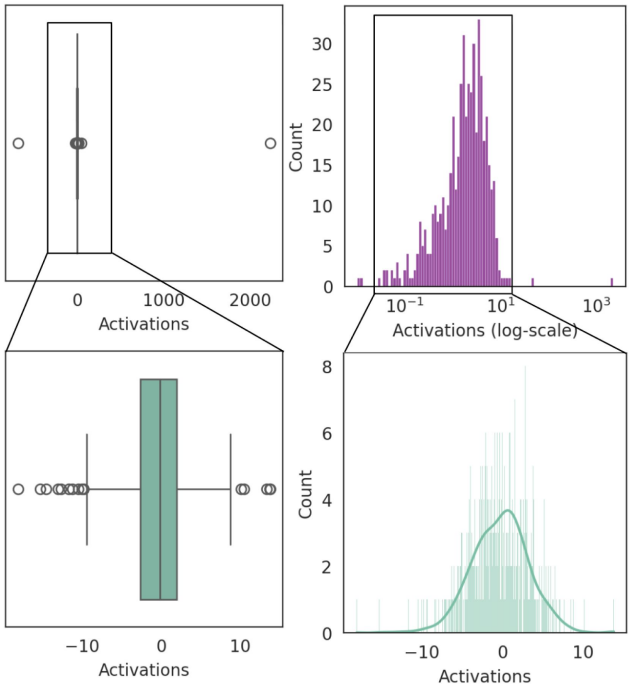
Figure 5: Attention patterns *before* and *after* massive activations appear in LLaMA2-7B. For each layer, we visualize average attention logits (unnormalized scores before softmax) over all heads, for an input sequence.

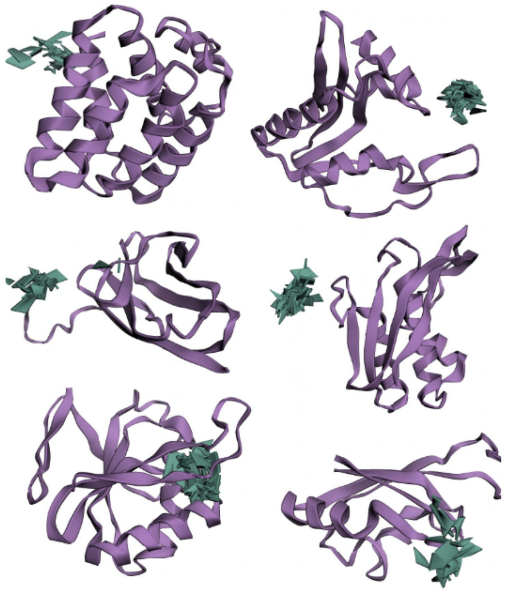


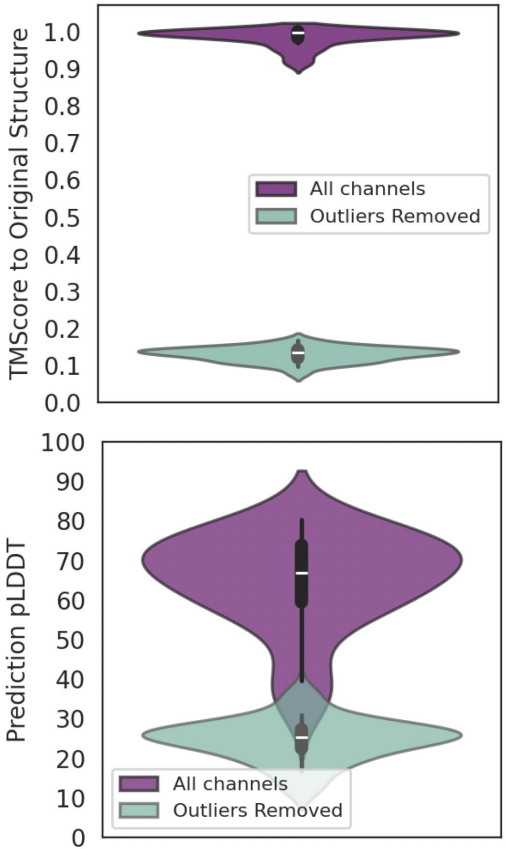
What if we just remove these wacky channels?



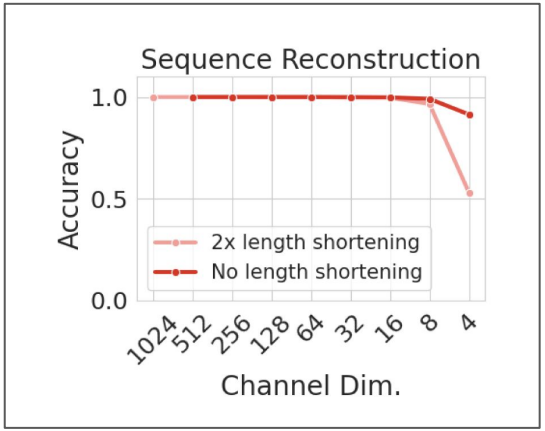
What if we just remove these wacky channels?

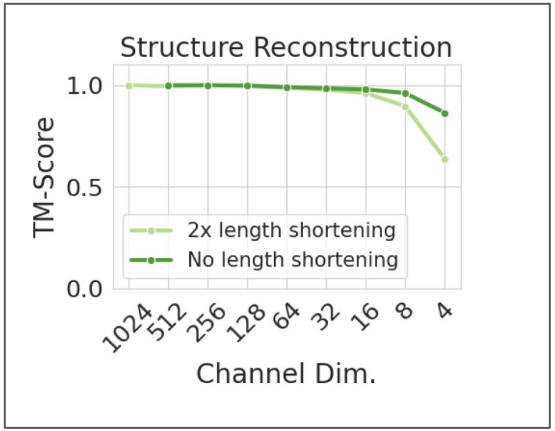






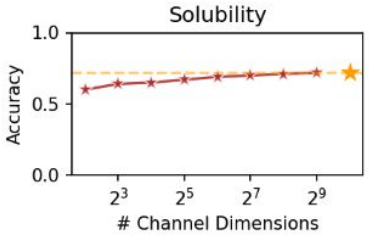
Learning an autoencoder to compress the latent space

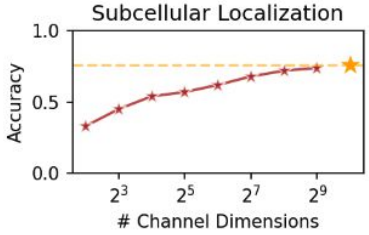




Turns out the latent space is highly compressible!
Sequence information is easier to retain than structure.

What about function information?

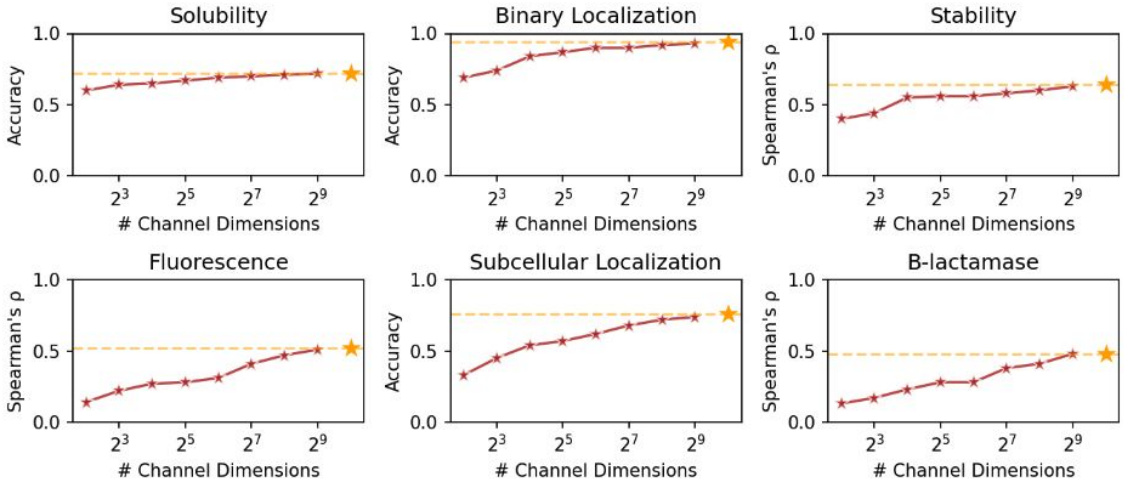




Performance degradation with compression is more gradual…

…for some functions.

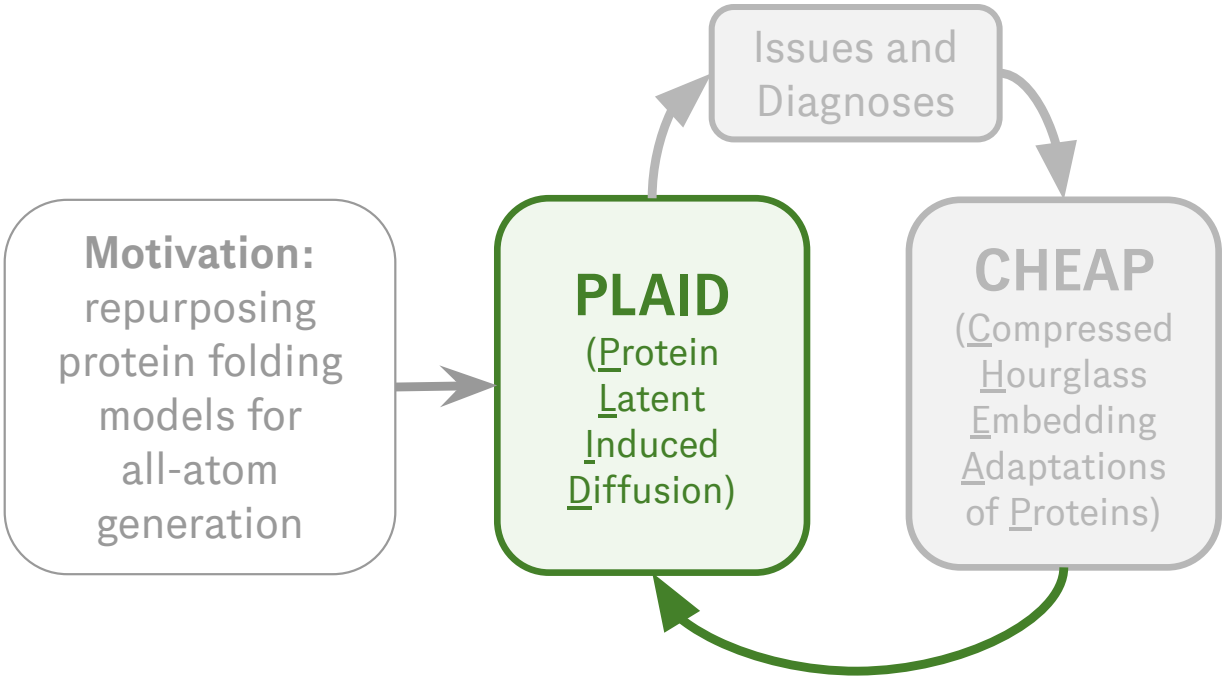
What about function information?



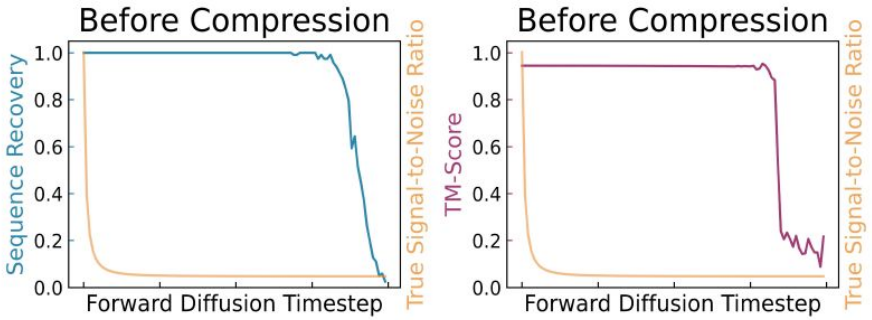
Performance degradation with compression is more gradual…

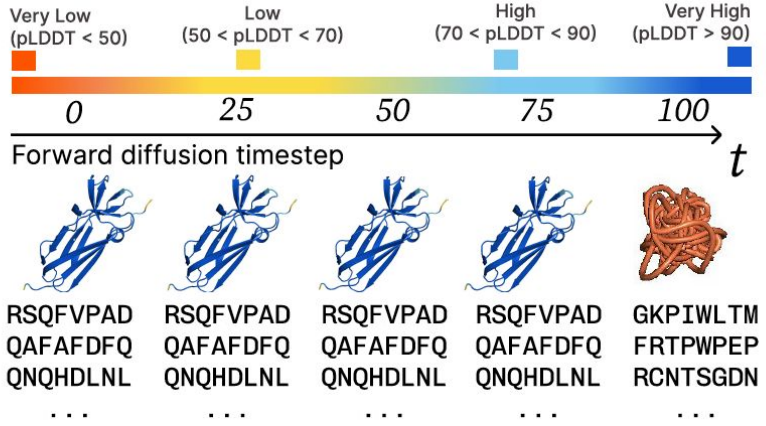
…for some functions.

Agenda

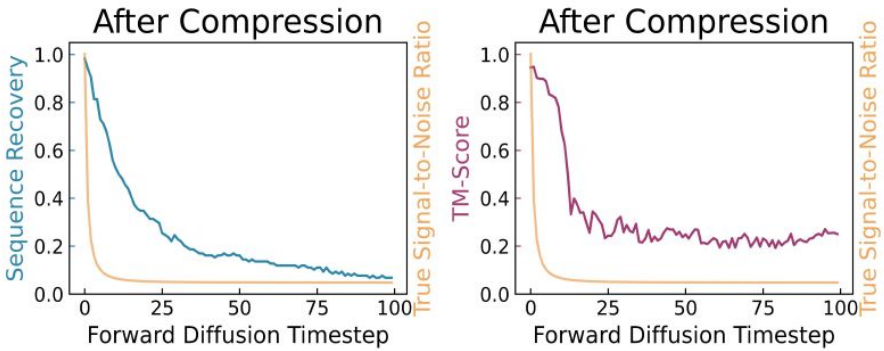

Issues and Diagnoses
Motivation: repurposing protein folding models for all-atom generation
PLAID (Protein Latent Induced Diffusion)
CHEAP (Compressed Hourglass Embedding Adaptations of Proteins)

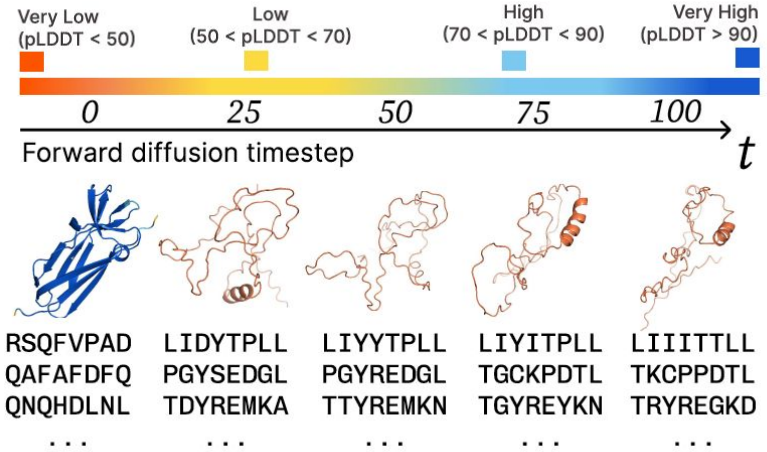
Noising the original latent space does not affect the structure...



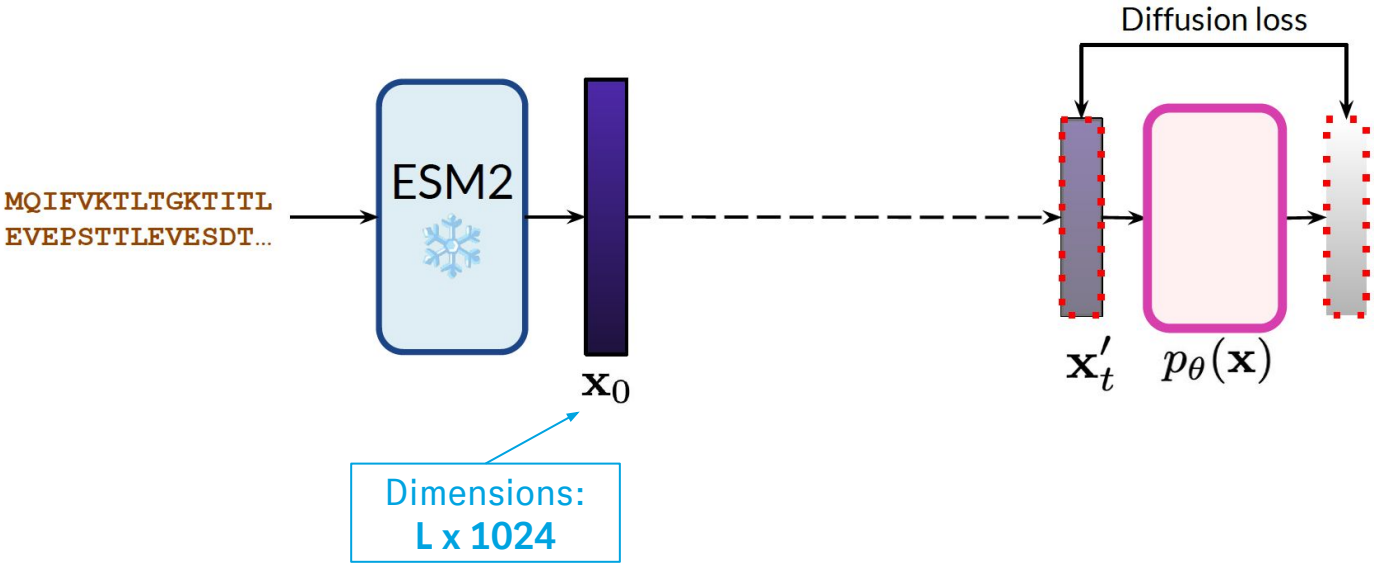


...noising the compressed latent space does map to corrupted structures

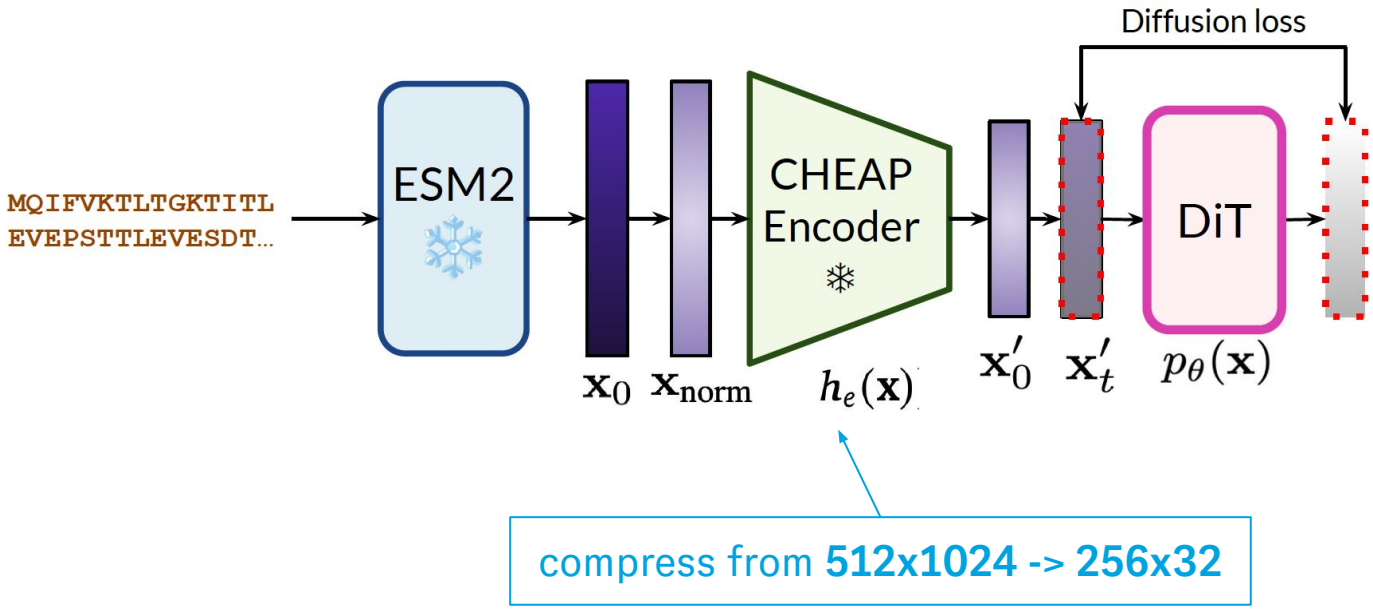




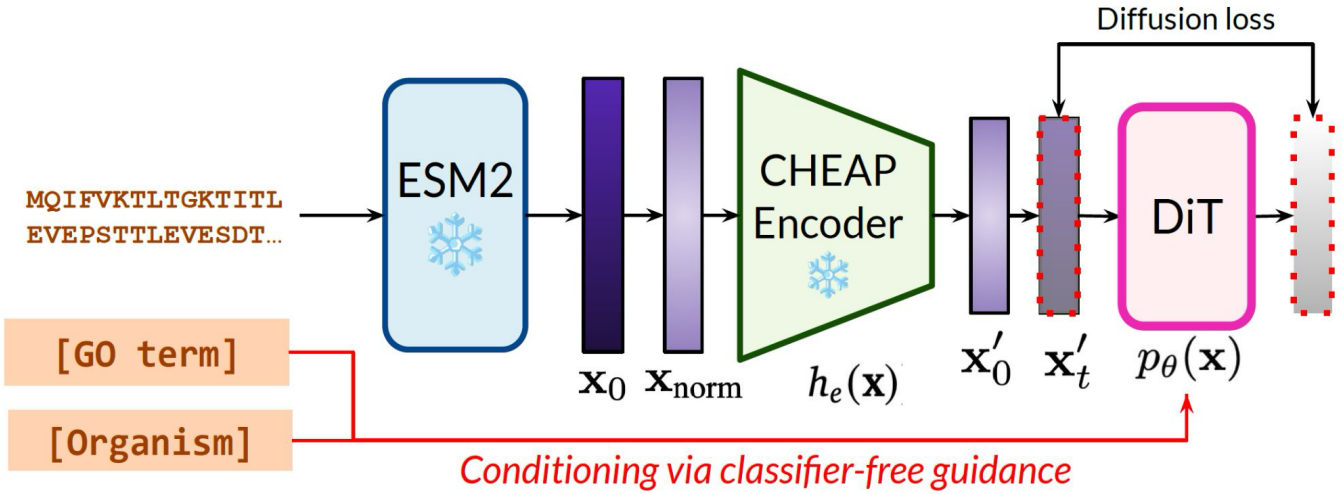
Training the PLAID latent diffusion model…



...but add embedding compression with CHEAP

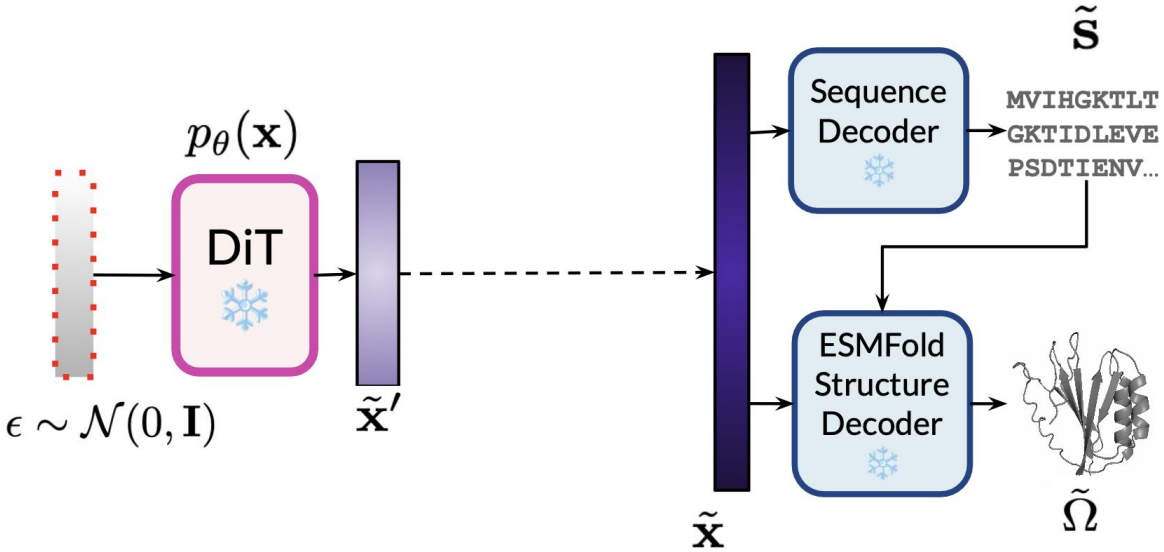


Adding compositional function + taxonomic conditioning

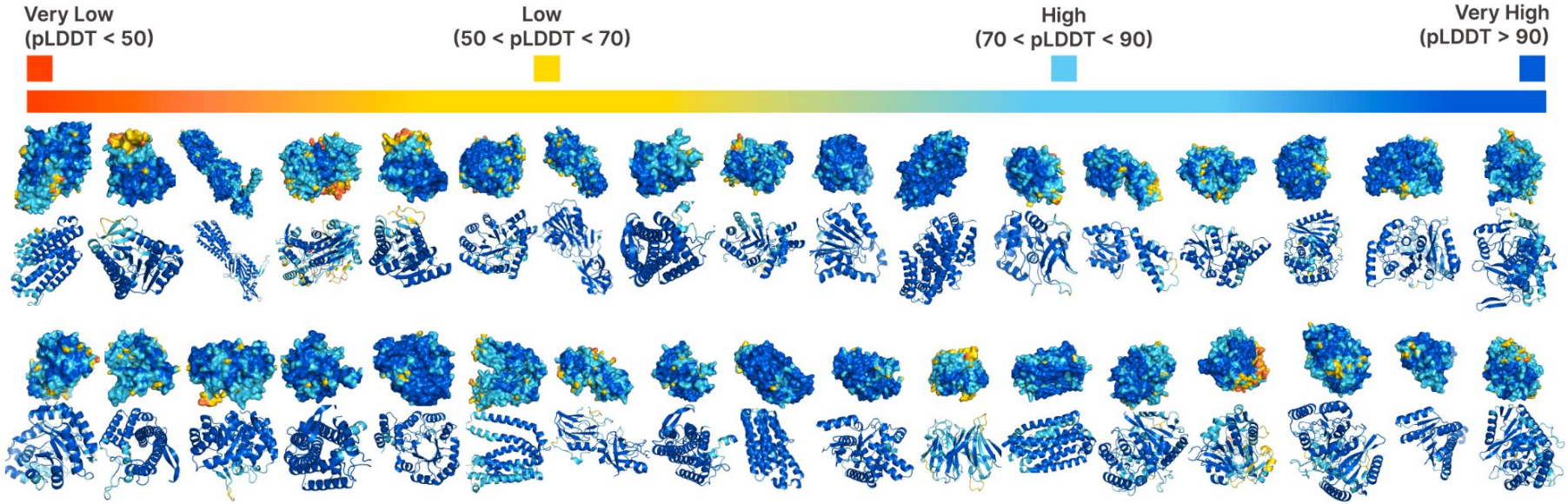


Sequence databases have more sample-annotation pairs!
-> unlocks new axis of controllability.

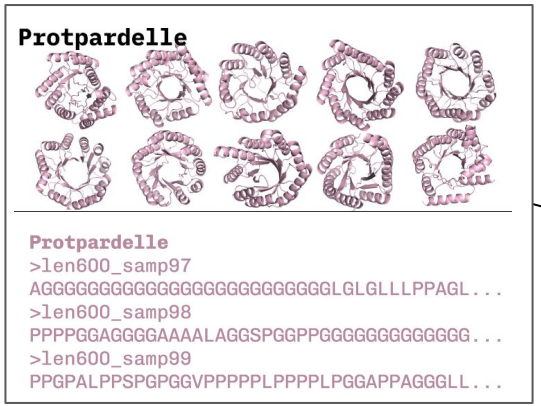
PLAID Inference with CHEAP decoder

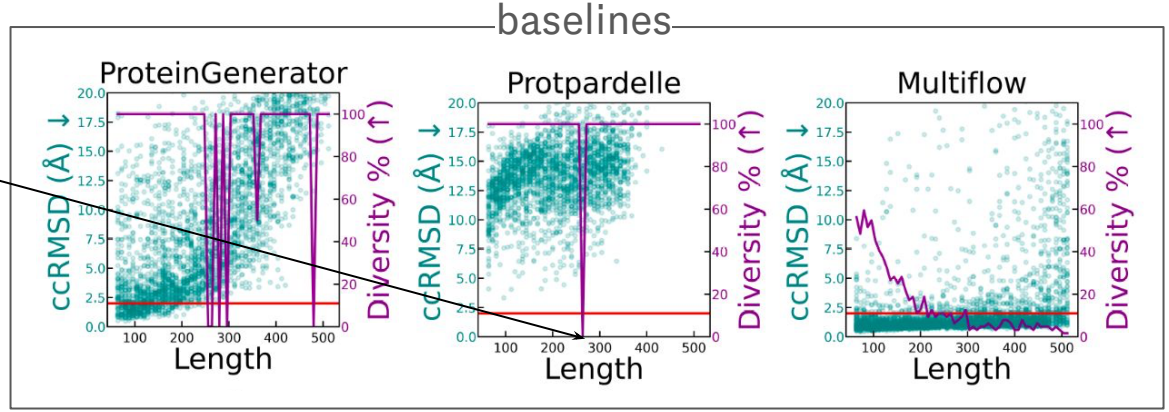


PLAID unconditionally generates diverse all-atom structures



PLAID unconditionally generates diverse, high-quality folds

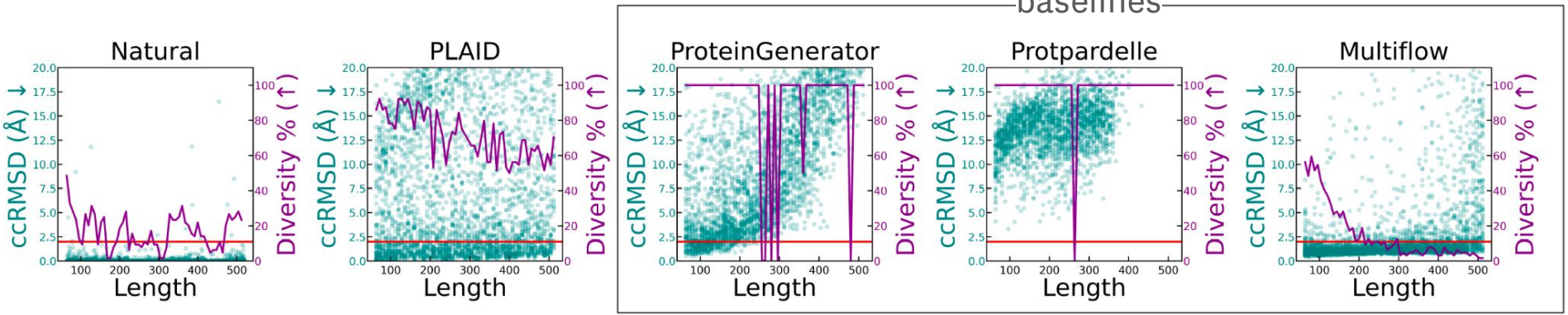




purple: diversity (↑)
(# of foldseek clusters / # of samples)

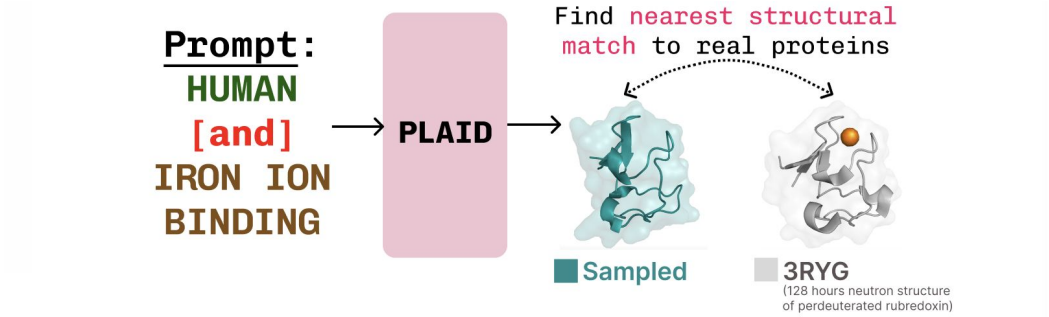
teal: quality (↓)
(ccRMSD between generated structure and predicted structure of generated sequence)

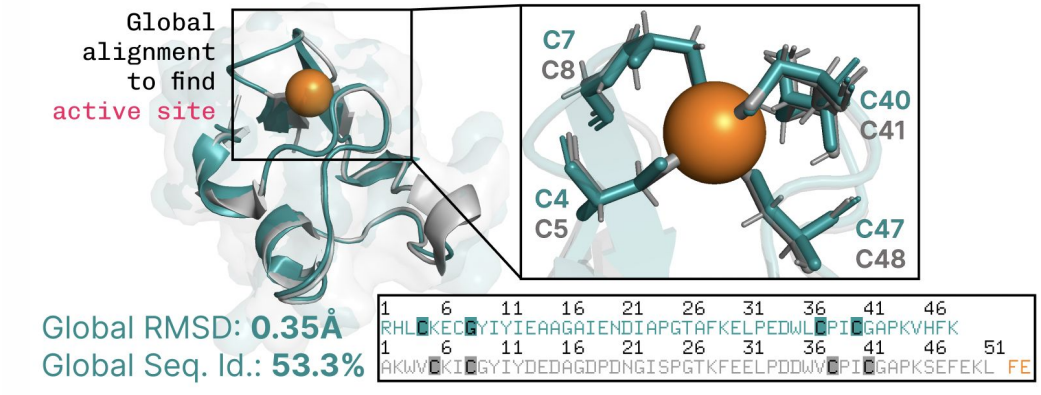
PLAID unconditionally generates diverse, high-quality folds



PLAID better balances diversity and quality, especially at longer sequence lengths.

Function-prompted generations learn active site sidechains

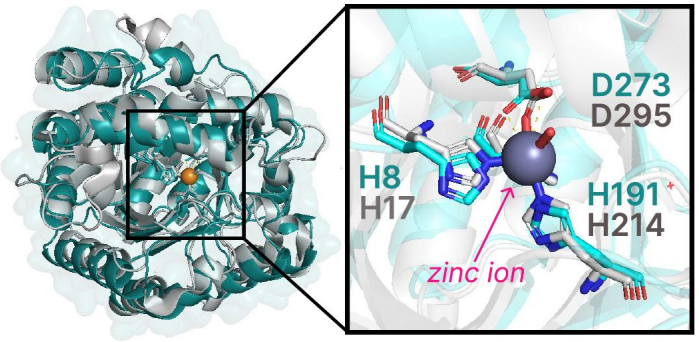




PLAID not only learns that cysteines coordinate the iron ion, but also the sidechain positioning...

Function-prompted generations learn active site sidechains

Prompt:
HUMAN [and] DEAMINASE ACTIVITY

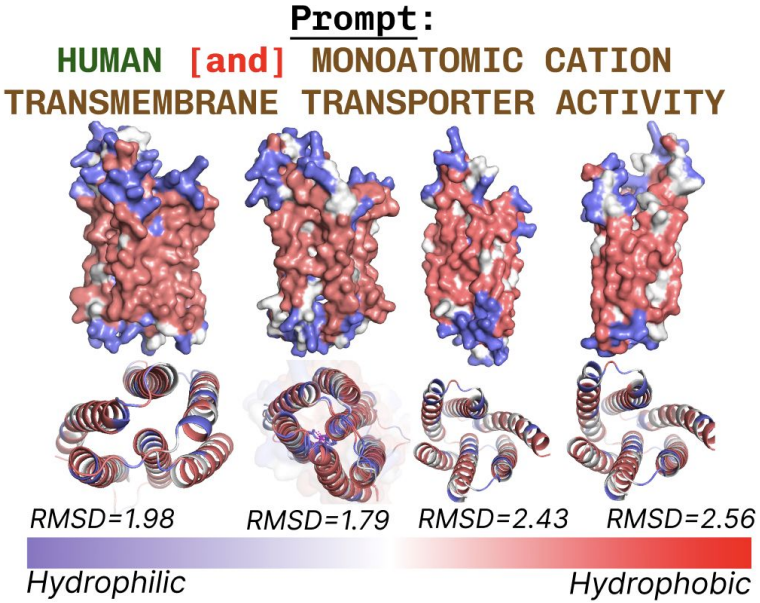


RMSD: **2.25Å**
Seq. Id.: **24.3%**

Sampled
7RTG (Crystal Structure of the Human Adenosine Deaminase 1)

...despite these key residues not being adjacent in the sequence.

Transmembrane proteins exhibit expected hydrophobicity patterns



Hydrophobic residues are found at the core, as expected.

Thanks!

Berkeley
Amy X. Lu
Wilson Yan
Pieter Abbeel

Microsoft Research
Kevin Yang

Prescient Design
Sai Pooja Mahajan
Sarah Robinson
Simon Kelow
Vladimir Gligorijevic
Kyunghyun Cho
Richard Bonneau
Nathan C. Frey

@amyxlu amyxlu.github.io amyxlu@berkeley.edu